

Permalink URI:

vscode://joelabenhaim.aicode-ja/chat?sessionId=d6598f1ebf5d43ba8ac1975ad77911d9-Finaliser-liveSession

<USER>

Je te montre la spécification qu'on vient de faire, ensuite je vais faire la feedback loop avec toi: finaliser compilation, lint, tests verts sans tricher. Lis la et dis moi quand tu es prêt. Ne vérifie pas d'avance, lis juste et on démarre.

<SPECIF\_FINALISER\_LIVESESSION\_POUR\_LES\_SESSIONS\_PROMPT\_NORMALES\_20260712\_204731> ##

Spécification — Finaliser `liveSession` pour les sessions Prompt normales ## Contexte La tranche 1 correspond au sous-système fonctionnel introduit au commit `f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33`. La présente réinspection a été refaite depuis zéro sur le dépôt propre courant au commit `c3d66083b46b178e24d4ce497cc3cd5d914a4170` (`c3d66083`). Les commits `b42c734c4178d0725ce98933c085a763288eb697`, `613acafc9ac510e432986f56456d63f185d77347`, `b16c5bdde3b4a27cec2ee3f44c9a471b44465ac6` et `0b9f52cc402ce6d0ae19372fa62a432851c248b0` ne sont pas les ancêtres directs du HEAD courant ; leur base commune pertinente avec le HEAD est `ed3bd94687cc34f6603730689bfc775818881bae`. Le diff entre `f65adc8e5172cc1e4f9781bf54ffa6dd089d3d33` et `ed3bd94687cc34f6603730689bfc775818881bae` est vide. Les écarts entre `b42c734c` et le HEAD concernent l'outillage de spécification ; les écarts supplémentaires depuis `ed3bd946` concernent ce même outillage et une documentation personnelle hors périmètre. Aucun arbre source ciblé par cette tranche n'a changé. La fondation actuelle fournit déjà : - un registre process-local de runs exacts ; - un buffer ordonné complet de `AiStreamedChunk` ; - des leases de viewer exactes ; - les barrières `sourceSettled` et `handoffReady` ; - l'ouverture live-safe sans reset de mémoire Orch ; - le replay backlog + tail ; - une protection anti-ABA par époque de locator ; - le terminal exact et la distinction `TOOL\_USE` / vraie fin de tour ; - les transports Slimbk et message bus génériques ; - un hook frontend générique réutilisant Queue A, Queue B, les locks et les sanitizers Prompt ; - un adaptateur EasyAgents préservant le comportement produit historique. Le problème utilisateur est qu'un renderer Prompt peut devenir gris, notamment après saturation mémoire Electron. Aujourd'hui, fermer le panneau exécute explicitement `Abort → AwaitStop → ClearMemory`, détruit la génération et oblige à relancer le dernier message. La tranche finale doit permettre à une génération Prompt normale de continuer dans l'extension, d'être visible dans Activité, d'être rouverte sur sa source exacte et de reconstruire l'historique puis le stream sans perte ni duplication. La tranche est atomique : capture de source, fermeture, projection Activité, réouverture, replay, Stop, son, finalisation frontend, persistance terminale, nettoyage et protections destructives doivent être livrés ensemble. Aucun état intermédiaire incomplet ne doit être introduit. Les invariants de conservation restent obligatoires : - aucun changement d'algorithme dans Queue A, Queue B, `PromptStreamLockStore` ou les sanitizers ; seules leurs entrées, sorties et mutations sont gardées par l'identité de tentative ; - aucun doublon de timer Activité ni de son lors d'un replay ; - aucun second bouton Stop dans Prompt ; - aucun panneau « invisible » : un panneau est `open` ou `closed` ; - aucun `\*.generated.\*` modifié manuellement ; - aucun fichier de documentation personnelle interdit chargé ou modifié ; - code, tests, commentaires et identifiants ajoutés ou modifiés en anglais ; - aucun `any`, `eslint-disable`, TODO, cast inutile, import dynamique de production ou compatibilité implicite non demandée. ## Objectifs décrits par l'utilisateur ### Besoin initial exprimé par l'utilisateur - Livrer toute la tranche finale en une seule implémentation cohérente. - Permettre à une génération Prompt normale de continuer lorsque son panneau est fermé. - Conserver l'item dans Activité lorsque le panneau est ouvert, ou fermé avec une source encore active. - Supprimer l'item d'Activité lorsque le panneau est fermé et la source terminée. - Conserver l'item Idle si le panneau est ouvert lorsque la source se termine, y compris si le record `liveSession` est retiré faute de baseline ou de vrai terminal. - Ajouter une troisième ligne Activité avec un bouton Stop uniquement pour `Running/source/closed`. - Utiliser l'icône Stop existante, `size="small"`, `variant="danger"`, label `Stop`, avec désactivation immédiate après clic. - Faire disparaître la ligne Stop détachée dès réouverture du panneau ou fin de la source. - Réutiliser dans le panneau reconnecté le Stop normal du compositeur. - Restaurer Prompt normal au plus tôt : après installation de la baseline et armement du replay, autoriser la saisie du prochain brouillon et la préparation des attachments, tout en gardant Send verrouillé. - Étendre SidebarStore pour représenter des activités de session et non seulement des fenêtres ouvertes. - Conserver le buffer raw-live complet, sans compression ni limite ajoutée. - Tester tout code créé, déplacé ou modifié, notamment les courses fermeture/réouverture/Stop/retry transport/sources successives/delete/Retry EasyAgents. - Mettre à jour les programming docs génériques `liveSession`, EasyAgents, isomorphisme, thinking, Queue B et fallback persist. - Respecter l'architecture modulaire, le typage strict et toutes les règles de qualité du projet. ### Clarifications implicites ajoutées par AICode - `LiveSessionRunRef` devient `{ locator, turnId, sourceId }`. `turnId` reste la corrélation logique persistante ; `sourceId` identifie une invocation backend précise. - `streamInstanceId` est frontend-only, change pour chaque stream effectivement accepté et remplace `streamKey`. `replayOpenAttemptId` est également frontend-only, change pour chaque séquence Open initiale ou Retry, et n'est jamais fabriqué pour représenter un stream qui n'a pas démarré. - Le descripteur live privé comprend exactement `openMode`, `liveSessionTurnId`, `liveSessionViewerLeaseId`, `liveSessionClosePolicy`, `liveSessionSourceId`, `liveSessionAdapter` et `easyAgentsBatchLineNumber`. Toute clé inconnue préfixée `liveSession`, toute présence partielle et tout doublon critique rendent la query invalide. - Le contrat shared expose une union stricte de descripteurs complets : `easyAgentsBatch + stopOnLastViewerClose + easyAgentsBatchLineNumber + sourceId===turnId`, ou `normalPrompt + continueOnPanelClose` sans numéro de ligne. Extension et frontend utilisent cette même union après détection des doublons. - Une query live explicite valide conserve toujours priorité. Une query live explicite invalide est rejetée avant le low-level opener et ne se dégrade jamais en ouverture classique. La validation frontend reste une défense en profondeur. - `Prompt.tsx` est l'unique propriétaire du snapshot brut figé de query. Le dispatcher est pur, utilise `URLSearchParams.getAll(...)`, ne lit jamais la globale router et retourne la classification ainsi qu'un locator strict lorsqu'il est univoque. - Une frontière React externe traite les queries invalides avant de monter le runtime Prompt. Aucun locator sentinelle ni hook conditionnel n'est autorisé. - Toute query live explicite, valide ou invalide, désactive l'hydratation classique et tout auto-run. L'injection normalPrompt retire également tous les paramètres `autoRun\*`. - Le contexte

d'origine d'un stream est une donnée extension-only non sérialisée : le message bus transmet au handler le `HtmlPage` et la webview exacte ayant émis la demande. Le guard de démarrage compare cette webview à celle actuellement enregistrée pour le locator. Un simple test par locator est insuffisant, car un ancien message en file ne doit jamais démarrer sur un nouveau panneau rouvert avec le même locator. - `useAiStream.start()` retourne synchroniquement `{ accepted: false }` | `{ accepted: true; runRef; streamInstanceId }`. Aucun buffer, lock, queue, scaffold, placeholder, `uiPending` ou ref de continuation n'est muté sur refus. - Send, SPECIF, Retry, Edit, Compact context, Reply et Continue appliquent tous une préparation pure, appellent `start()`, puis committent exactement un scaffold uniquement après acceptation. - Les barrières Prompt de clear/focus sont corrélées à `{ turnId, streamInstanceId }`. Si `fallback\_persisted` arrive pour cette tentative, le compositeur est effacé comme aujourd'hui. Si la tentative se finalise sans cet ack, le masque transactionnel est retiré mais le texte, les attachments, `nextTurnId`, `pendingEditTurnId` et le mode maximisé restent intacts. - Les callbacks `onData`, `onComplete`, `onError`, leurs continuations après `await` et le `runMerged` déjà sorti de Queue A sont gardés par `sourceId + streamInstanceId`. - `completed` ou `transportError` n'est publié, et le verrou d'acceptation n'est libéré, qu'après Queue A idle, Queue B idle et la dernière barrière de commit. - Si un vrai terminal data a déjà été consommé puis qu'une erreur transport/auxiliaire arrive, le terminal reste unique ; aucun second `meta.error` ni suffixe n'est injecté et l'outcome devient `transportError`. - Le handoff direct normalPrompt exige terminal data exact, transport normalement complété, Queue A/Queue B drainées et `finalizedStreamInstanceId` correspondant. Il passe directement à `normal` sans second Open ni seconde hydratation du compositeur. - Toute réponse Open est gardée par `replayOpenAttemptId`. Une branche `terminal` n'alloue aucun `streamInstanceId`. - Les retries normalPrompt utilisent une hydratation `preserveLocal` qui ne lit, ne valide, ne normalise ni n'enregistre `promptDraft`, `promptDraftAttachments` ou `meta.nextTurnId`. Les tests utilisent aussi des getters piégés pour prouver l'absence réelle de lecture. - Le wrapper normalPrompt persiste idempotemment tout `meta.error` reçu depuis Orch avant de l'appendre au registre ou de l'exposer. Cette barrière couvre notamment le chemin auto-accept où Orch peut avoir construit un terminal alors que sa première tentative de patch durable a échoué. - Tout terminal wrapper synthétique après throw/EOF est créé uniquement sans terminal antérieur, patché durablement avec suffixe canonique, puis appendu. Si la persistance échoue, aucun terminal n'est appendu et le transport échoue. - Le protocole privé V1 ne concerne que le premier chunk source `meta.error`. Après un header `200\_OK`, un `meta.error` reste un body chunk normal. - Le framework Slimbk actuel sérialise une exception survenue après le header `200\_OK` sous forme d'un `SlimbkResult` non-200 dans le body. Le client généré le transmet actuellement au callback `onChunk`. Les handlers extension doivent reconnaître cette enveloppe avant toute logique `AiStreamedChunk`, la transformer en erreur transport et ne jamais la bufferiser, l'émettre ou tenter de la décoder comme protocole source V1. - Une réservation source revendique le pointeur actif et avance l'époque avant tout `await`. Sa promotion conserve la même époque. Annulation, handoff ou retraite retirent le pointeur, avancent l'époque puis mémorisent l'issue avec l'époque post-retrait. - Une réservation exacte dupliquée est refusée. Le handler qui n'a pas obtenu la réservation ne peut ni appeler l'endpoint source, ni annuler, promouvoir ou finaliser la réservation du premier handler. - La finalisation post-forwarding est réservée à `ownership='source'` et retourne exactement `reservationCancelled` | `handoffReady` | `retiredMissingBaseline` | `retiredMissingTerminal` | `stale`. - Un record sans baseline ou sans vrai terminal est retiré exactement ; aucun record `sourceSettled` non handoffable ne reste bloqué. Si les deux manquent, `retiredMissingBaseline` a priorité. - Les retraites structurées, utilisées pour fallback/idempotence, restent distinctes des tombstones exactes process-lifetime empêchant la résurrection d'un run key observé. - Le registre émet après mutation complète deux wake-ups profondément immuables : observer devenu inactif et claim destructif libéré. Les listeners sont isolés et ces événements ne portent aucune décision métier. - Les suppressions durables et Retry EasyAgents utilisent un claim destructif opaque de locator. Acquisition et libération avancent une génération destructive distincte de l'époque de run. L'éviction publique exige le handle exact et ne libère jamais le claim. - Un dispose sous claim conserve le mapping détaché et diffère `Abort/Await/ClearMemory`, orphan cleanup, son et cleanup trivial jusqu'au wake-up de release, puis revalidation. - L'acceptation d'un Stop source met aussi synchroniquement à jour `stopRequested` dans le runtime Activité avant l'abort. Ainsi, si le panneau se ferme juste après le Stop du compositeur, le Stop détaché apparaît déjà désactivé. - EasyAgents conserve `sourceId=turnId`, son elapsed viewer historique, `last viewer → kill`, son settlement/handoff métier et son UX read-only. ## Vue d'ensemble de la méthode choisie ### Identité et démarrage transactionnel frontend 1. Chaque commande construit un plan pur sans mutation visible. 2. `useAiStream.start()` acquiert synchroniquement le single-flight ; sur refus il retourne immédiatement sans mutation. 3. Sur acceptation, il génère `streamInstanceId` et, pour une source Prompt, `sourceId`, construit le runRef exact et retourne l'identité. 4. Le caller committe alors le scaffold, les refs de thinking/assistant, `uiPending`, le scroll et les barrières locales. 5. L'effet transport démarre seulement au rendu suivant ; les barrières sont donc armées dans la même pile avant tout callback réseau. 6. Les handlers extension reçoivent un contexte d'origine exact non sérialisé. Ils vérifient que la webview émettrice est encore celle enregistrée pour le locator avant la première mutation. ### Cycle backend d'une source Prompt normale 1. Guard exact de panneau par webview d'origine. 2. Réservation source exacte comme première mutation synchrone ; revendication du pointeur actif, bump d'époque et capture de `sourceStartedAtMs`. 3. Allocation synchrone du runtime/timer Activité et de `activityRunId` ; la publication est démarrée séparément et reste auxiliaire. 4. Armement du son exact et récupération de `soundAttemptId`. 5. Persistance éventuelle de la préférence SPECIF Code, chargement catalogue, puis capture sans `await` de la baseline de continuation immédiatement avant `askStream(...)`. 6. Priming lazy du `Readable`, promotion réservation → record par `registerRun`, application d'un abort précoce. 7. `continueAssistant` publie immédiatement sa baseline de messages ; `replaceTurnAssistants` attend `fallback\_persisted`. 8. Chaque chunk ordinaire est parsé, éventuellement persisté s'il s'agit de `meta.error`, appendu au registre puis exposé. 9. Un throw/EOF sans terminal tente de persister un terminal canonique avant append ; un throw après terminal reste transport. 10. Le service source ne finalise jamais le registre. 11. L'endpoint encode uniquement un premier `meta.error` dans le header V1. Après `200\_OK`, les chunks restent dans le body ; une exception post-header devient une enveloppe Slimbk d'erreur dans le body. 12. Le handler extension maintient une FIFO unique pour les chunks ordinaires et le terminal décodé depuis le

premier header. Une enveloppe Slimbk d'erreur post-header est interceptée avant la FIFO et classée transport. 13. Les erreurs auxiliaires sont mémorisées dans l'ordre sans empêcher les étapes restantes du chunk ni les chunks suivants. 14. Après retour Slimbk, drainage FIFO, finalisation atomique du registre, arrêt du son exact, puis délégation au coordinateur pour Activité/cleanup. 15. Le frontend publie l'outcome seulement après Queue A, Queue B et commit ; le contrôleur Prompt publie ensuite `finalizedStreamInstanceId` après sa propre finalisation UI. ### Replay et handoff - `replaceTurnAssistants` recharge l'enveloppe persistée récente et retire uniquement les assistants du tour exact. - `continueAssistant` superpose la baseline de messages immuable sur l'enveloppe persistée la plus récente. - `not\_ready` normalPrompt est retryé avec une seule requête Open en vol et un délai annulable. - `not\_found`, transportError, completion sans terminal ou éviction sans terminal conduisent à `liveSessionReplayFailed` pour normalPrompt. - Le handoff normalPrompt est direct sur l'état replayé ; EasyAgents et les bootstraps déjà terminaux utilisent le snapshot terminal. - Un retry normalPrompt remplace messages et stores d'hydratation mais conserve intégralement le compositeur local, y compris si Open répond directement `terminal`. #### Registre, retraites, tombstones et claims Le registre distingue : - réservation source active ; - record actif ; - retraite structurée avec issue + époque post-retrait ; - tombstone exacte process-lifetime ; - claim destructif opaque de locator ; - époque de run et génération destructive indépendantes. Les Open capturent et revalident les deux générations. Un cycle acquire/release invalide uniquement l'Open en vol. Une nouvelle source rend toute répétition tardive de finalisation ancienne `stale`. L'éviction sous claim retire réservations, records, leases et retraites replayables, conserve les tombstones et ne relâche pas le claim. ### Panneaux, Activité, son et Stop - Le public `openCentralPanel(...)` devient async. Il délègue Prompt au coordinateur et garde le low-level opener privé pour les autres pages. - Le coordinateur est une classe injectée qui n'importe jamais le low-level opener. - Open, dispose, sourceSettled et wake-ups sont sérialisés par locator dans des queues supprimées à idle. - Le low-level dispose supprime synchroniquement le panneau du registre avant d'appeler le coordinateur. - Le coordinateur est l'unique propriétaire de la transition terminale Activité d'une source normale. - `SidebarActivityQueryItem` est une union stricte : Idle/open ; Running/observer/open ; Running/source/open/closed. - Le runtime Activité garde runRef, ownership, runId, timestamps, timer, usage et `stopRequested` ; les transitions sont exactes par runRef + runId. - Le Stop détaché existe uniquement pour Running/source/closed. Le sous-arbre est keyé par `sourceId`. - Le son utilise runRef exact pour la déduplication et `soundAttemptId` pour isoler les tentatives viewer successives. #### Suppression et Retry destructifs 1. Préchecks UI/Git historiques. 2. Acquisition synchrone du claim destructif juste avant fermeture. 3. Fermeture du panneau ; le dispose constate le claim et diffère tout cleanup concurrent. 4. Suppression durable : éviction seulement après succès disque. 5. Retry EasyAgents : éviction pré-delete historique sous le même claim. 6. Libération en `finally`, bump de génération et wake-up du coordinateur. ## Liste exacte des fichiers avec détails d'implémentation ### Contrats shared ### 1) `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionReplay.type.ts` — créé - Définir `ZLiveSessionReplayMode` / `LiveSessionReplayMode` : `replaceTurnAssistants | continueAssistant`. - Documenter la reconstruction frontend de chaque mode. ### 2) `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.type.ts` — modifié - Ajouter `LiveSessionSourceId` strict trim/non vide et `sourceId` obligatoire au runRef. - Inclure `sourceId` dans la run key, sans modifier la locator key. - Documenter la différence `turnId` logique / `sourceId` exécutif. ### 3) `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionPanel.type.ts` — modifié - Ajouter `LIVE\_SESSION\_SOURCE\_ID\_PARAM`, `LIVE\_SESSION\_ADAPTER\_PARAM` et centraliser `EASY\_AGENTS\_BATCH\_LINE\_NUMBER\_PARAM`. - Définir `ZLiveSessionAdapter` : `easyAgentsBatch | normalPrompt`. - Définir la liste canonique readonly des paramètres privés. - Définir une union stricte `ZLiveSessionPanelDescriptor` : EasyAgents exige `stopOnLastViewerClose`, ligne positive et `sourceId===turnId` ; normalPrompt exige `continueOnPanelClose` et interdit la ligne EasyAgents. - Réutiliser les schémas runRef/sourceId/lease ; aucun parseur extension/frontend ne redéfinit ces couples. ### 4) `AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionOpenResult.type.ts` — modifié - Ajouter `sourceId` aux branches live/terminal et `replayMode` uniquement à live. - Étendre le transport et `superRefine`, y compris les propriétés présentes à `undefined`. - Conserver les schémas canoniques complets sans rendre `ZChatSessionFile` deep-strict. - Ajouter `ZLiveSessionOpenResultEnvelope` avec `session: unknown`, strict sur discriminant et identité, réservé au parse frontend policy-aware. ### 5) `AICode/src/sharedlib/src/types/liveSessionTypes/\_\_test\_\_/LiveSessionContracts.test.ts` — modifié - Couvrir run keys, sourceId, replay modes, paramètres, liste privée, adaptateurs, couples complets, égalité EasyAgents source/turn, ligne requise/interdite et propriétés `undefined`. - Comparer schéma Open complet et enveloppe sélective, sans resserrement historique accidentel. ### 6) `AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts` — modifié - Ajouter fabriques canoniques SYSTEM\_ERROR/INTERRUPTED. - Définir encodeur/décodeur V1 exact : `500\_SERVER\_ERROR`, code/préfixe `AICODE\_LIVE\_SESSION\_SOURCE\_ERROR\_V1`, payload strict `{ version: 1, value }`. - Le décodeur accepte chaîne, Error et objet structurel ; il respecte strictement `status` et la présence propre de `errorCode`. - Extraire une ligne utile de stack et reconstruire exactement `AiErrorMetaValue`. ### 7) `AICode/src/sharedlib/src/types/aiTypes/\_\_test\_\_/aiStopSuffix.util.test.ts` — modifié - Tester réversibilité, statuts/codes, propriété présente à `undefined`, stacks, détails multilignes, extras invalides et chunks canoniques. ### Persistance terminale Orch ### 8) `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/OrchMessagesMemoryAssistantPatch.ts` — modifié - Ajouter le suffixe canonique idempotent pour SYSTEM\_ERROR comme pour INTERRUPTED. - Préserver blocs, métadonnées, usages et sink ; ne jamais dupliquer un suffixe existant. ### 9) `AICode/src/extension/src/aiTools/orch/context/memory/core/msg/\_\_test\_\_/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts` — créé - Couvrir SYSTEM\_ERROR, INTERRUPTED, texte absent/présent, suffixe existant, sink et hydratation isomorphe. ### Contrats Sidebar ### 10) `AICode/src/sharedlib/src/types/busTypes/ui/SidebarStore.ts` — modifié - Remplacer `SidebarActivityQueryWindowItem` par l'union stricte `SidebarActivityQueryItem`. - Branches : Idle/open ; Running/observer/open ; Running/source/open]

closed. - Conserver titre, elapsed, usage, `updatedAt`, `openedAt`; inclure runRef exact et `stopRequested` seulement pour source. - Renommer `queryState.windows` en `queryState.items`. ### 11)

`AICode/src/sharedlib/src/types/busTypes/ui/SidebarUiMessages.ts` — modifié - Ajouter `sidebar->ext:openQueryActivity(locator)` et `sidebar->ext:stopDetachedQuery(runRef) -> { accepted }`. ### 12)

`AICode/src/sharedlib/src/types/busTypes/global/MessageRegistry.ts` — modifié - Ajouter sourceId aux streams normal et liveSession. - Ne jamais transporter `streamInstanceId`, `replayOpenAttemptId` ni le contexte extension-only de webview. ### Backend Orch `liveSession` ### 13)

`AICode/src/extension/src/aiTools/orch/liveSession/liveSession.programming-doc-readme.md` — modifié - Documenter identité, réservations, époques, replay modes, protocoles premier-header et erreur post-header-body, barrière de persistance de tout `meta.error`, finalisation post-forwarding, retraites/tombstones, claims, wake-ups, panel-origin exact, Activité, son et identités frontend-only. ### 14)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRecord.type.ts` — modifié - Ajouter `baselineOnly`, replayMode, baseline messages, timestamp, abortRequested, ownership source/observer, réservations, retraites et tombstones. - Définir `LiveSessionPostForwardingFinalizationResult` et le handle opaque `LiveSessionLocatorDestructionClaim`. - Ajouter génération destructive aux permits Open. - Définir événements immuables observer-inactif et claim-libéré. - Définir snapshots actifs/réservés sans exposer de mutable. ### 15)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRegistry.service.ts` — modifié - Implémenter réservation source exacte, annulation, promotion atomique via `registerRun`, refus de duplication et application d'abort précoce. - La réservation revendique le pointeur et bump l'époque; la promotion ne rebump pas. - Implémenter listings/snapshots, claims exacts d'abort source/viewer et suppression de `canAbortSource`. - Implémenter claims destructifs, génération destructive, éviction exigeant le handle et blocage reserve/register/Open. - Finaliser source après forwarding; retirer tout record sans baseline/terminal, priorité missingBaseline, idempotence dans l'époque puis stale. - Conserver tombstones après éviction et émettre les deux wake-ups après mutation complète avec listeners isolés. ### 16)

`AICode/src/extension/src/aiTools/orch/liveSession/registry/\_\_test\_\_/LiveSessionRegistry.service.test.ts` — modifié - Couvrir réservations, duplicate owner, promotion, abort, baseline modes/messages, finalisation cinq résultats, stale, retraites/tombstones, sources successives, claims, générations, événements, éviction sous handle et waiter réveillé sans terminal. ### 17)

`AICode/src/extension/src/aiTools/orch/liveSession/core/LiveSessionService.ts` — modifié - Exposer toutes les nouvelles opérations/snapshots/claims/souscriptions/finalisation. - Supprimer `canAbortSource` et rendre l'éviction destructive accessible uniquement par claim exact. ### 18)

`AICode/src/extension/src/aiTools/orch/liveSession/core/\_\_test\_\_/LiveSessionService.test.ts` — modifié - Couvrir forwarding complet, réservations, sources successives, stale, missing terminal, claims, générations, signaux et tombstones. ### 19)

`AICode/src/extension/src/aiTools/orch/liveSession/open/LiveSessionOpen.service.ts` — modifié - Retourner not\_ready pour réservation exacte et refuser sous claim. - Revalider époque de run + génération destructive autour de toute lecture. - Appliquer replaceTurnAssistants ou continueAssistant sur l'enveloppe récente. - Contrôler fallback terminal par retraite + époque post-retrait et retourner sourceId/replayMode. ### 20)

`AICode/src/extension/src/aiTools/orch/liveSession/open/\_\_test\_\_/LiveSessionOpen.service.test.ts` — modifié - Couvrir réservations, baselines, continuation, sourceId, retraites, source successive, claim, Open en vol invalidé et record retiré sans terminal. ### 21)

`AICode/src/extension/src/aiTools/orch/liveSession/replay/\_\_test\_\_/LiveSessionReplay.service.test.ts` — modifié - Ajouter sourceId, isolation des subscriptions, attente de finalisation, éviction sans terminal et éviction publique sous claim. ### 22)

`AICode/src/extension/src/aiTools/orch/liveSession/source/LiveSessionSourceLifecycle.service.ts` — créé - Extraire le lifecycle lazy `Readable` OOP générique: priming, first next, callback start sync, consommation, return/destroy, cleanup et agrégation déterministe. - Permettre au caller d'interdire l'abort thread-level sur échec post-return. ### 23)

`AICode/src/extension/src/aiTools/orch/liveSession/source/\_\_test\_\_/LiveSessionSourceLifecycle.service.test.ts` — créé - Couvrir priming, cleanup, settlement unique, erreurs combinées et mode sans abort tiers. ### 24)

`AICode/src/extension/src/aiTools/orch/liveSession/source/NormalPromptLiveSessionSource.service.ts` — créé - Mapper AiQuery → replayMode et readiness baselineOnly. - Capturer la baseline continue sans `await` avant `askStream`, valider l'assistant, primer puis promouvoir la réservation. - Publier baseline continue immédiatement ou baseline replace sur `fallback\_persisted`. - Avant tout append d'un `meta.error` Orch, appeler idempotemment `patchLastAssistantMeta` avec reason/message/details et suffixe; append/yield seulement après succès. - Cette barrière doit corriger le cas où Orch a émis un terminal auto-accept malgré un premier patch durable échoué. - Append-before-yield pour tous les chunks; suivre le vrai terminal. - Synthétiser SYSTEM\_ERROR/INTERRUPTED uniquement sans terminal et après persistance exacte; sinon propager le transport. - Ne jamais finaliser le registre. ### 25)

`AICode/src/extension/src/aiTools/orch/liveSession/source/\_\_test\_\_/NormalPromptLiveSessionSource.service.test.ts` — créé - Couvrir tous les query types, continuation, baseline, ordre, abort, premier error/header, error tardive, incoming meta.error déjà persistée ou nécessitant un second patch, échec persistance auto-accept, throw/EOF, throw après terminal et absence de settlement prématuré. ### Adaptation EasyAgents ### 26)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/EasyAgentsBatchLiveSessionAdapter.service.ts` — modifié - `replayMode='replaceTurnAssistants'`, `sourceId=turnId`, timestamp. - Remplacer `canAbortSource` par claims exacts. - Exposer autorité source/réservation, claim destructif et éviction sous handle, sans dépendre du coordinateur. ### 27)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/\_\_test\_\_/EasyAgentsBatchLiveSessionAdapter.service.test.ts` — modifié - Tester identité, replayMode, timestamp, claims, handle d'éviction et wake-up générique. ### 28)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunExecution.service.ts` — modifié - Construire tous les runRefs avec `sourceId=liveTurnId`, sans changer les transitions Batch. ### 29)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.live-activity-and-

gating.service.test.ts` — modifié - Étendre les assertions d'identité/replayMode en préservant Watchable/Handoff. ### 30)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.commit-and-recovery-failures.service.test.ts` — modifié - Ajouter sourceId aux assertions handoff sans changer l'ordre commit/cleanup/publication. ### 31)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.post-commit-persistence.service.test.ts` — modifié - Ajouter sourceId et préserver l'ordre patch parent → publication → handoff. ### 32)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunChildExecution.service.ts` — modifié - Remplacer le lifecycle local par `LiveSessionSourceLifecycleService` injecté ; conserver parsing/classification/callbacks. ### 33)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunChildExecution.service.test.ts` — modifié - Conserver les tests Batch et déplacer les tests lifecycle purs. ### 34)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunManager.service.ts` — modifié - Instancier/injecter le lifecycle partagé. - Migrer `items` et compléter le précheck Retry avec l'autorité registre. - Passer claims destructifs et éviction sous handle au Retry. ### 35)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.testHarness.ts` — modifié - Injecter lifecycle et sourceId. ### 36)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunLifecycle.service.test.ts` — modifié - Adapter la construction au lifecycle partagé. ### 37)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchLastViewerStop.service.test.ts` — modifié - Ajouter sourceId et conserver les courses last-viewer/ancien-nouveau run. ### 38)

`AICode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md` — modifié - Documenter sourceId=turnId, lifecycle partagé, observer/open, elapsed historique, last viewer kill, claim Retry et wake-ups observer/claim. ### Endpoints et message bus ### 39)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/open.endpoint.ts` — modifié - Exiger sourceId et transporter sourceId/replayMode. ### 40)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/open.endpoint.test.ts` — modifié - Couvrir champs, branches et mismatches. ### 41)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts` — modifié - Exiger sourceId et relayer runRef complet. ### 42)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/stream.endpoint.test.ts` — modifié - Couvrir identité, terminal, éviction sans terminal, attente finalisation et erreur post-header body. ### 43)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abort.endpoint.ts` — modifié - Exiger sourceId et utiliser le claim viewer atomique avec lease active. ### 44)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abort.endpoint.test.ts` — modifié - Couvrir viewer, stale, double Stop, remplacement et idempotence. ### 45)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abortSource.endpoint.ts` — créé - Stop exact source Prompt par locator+turn+source, réservation ou record, claim puis abort sans `await` intermédiaire. - Refuser observer, settled, stale, double claim et absent. - Après claim accepté, notifier synchroniquement le runtime Activité `sourceStopRequested` avant l'abort ; la publication Sidebar reste auxiliaire. ### 46)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abortSource.endpoint.test.ts` — créé - Couvrir réservation, record, stale, double clic, observer, ordre claim → stopRequested → abort et fermeture immédiate du panneau après Stop. ### 47)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/stream.endpoint.ts` — modifié - Ajouter sourceId, préserver ordre SPECIF/catalogue et déléguer au service source. - Encoder uniquement le premier `meta.error` sous header V1. - Après `200\_OK`, envoyer les `meta.error` comme body chunks normaux. - Ne jamais tenter un second header applicatif. Une exception post-header est laissée au framework Slimbk, dont l'enveloppe body sera filtrée par les handlers extension. - Ne pas finaliser le registre. ### 48)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/\_\_test\_\_/stream.endpoint.test.ts` — modifié - Tester délégation, premier header V1, meta.error tardif body 200, persistance préalable, absence de synthèse après terminal, absence de finalisation et exception post-200 sérialisée comme enveloppe Slimbk body. ### 49)

`AICode/src/extension/src/messageBus/handlers/ia/liveSessionStreamHandlers.ts` — modifié - Relayer sourceId et recevoir le contexte d'origine extension-only. - Refuser une demande dont la webview d'origine n'est plus le panneau exact enregistré, y compris si un nouveau panneau du même locator existe. - Source ownership : aucun timer/son supplémentaire. Observer : ressources uniquement pour ce panneau exact. - Filtrer toute enveloppe Slimbk non-200 post-header avant la FIFO et la classer transport. - Utiliser runRef + soundAttemptId, allocation Activité synchrone et cleanup exact. ### 50)

`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/liveSessionStreamHandlers.test.ts` — modifié - Couvrir sourceId, ownership, contexte exact, ancien panneau remplacé, observer fermé, publication Activity échouée, son stale, body-error post-header, terminal et éviction. ### 51)

`AICode/src/extension/src/messageBus/handlers/ia/iaStreamHandlers.ts` — modifié - Recevoir sourceId et contexte d'origine exact. - Vérifier webview + locator avant réservation ; un ancien message d'un panneau disposé ne doit pas utiliser un panneau rouvert du même locator. - Réserver, allouer Activité et son synchroniquement avant tout `await` métier. - Démarrer publication Activity comme tâche auxiliaire sans empêcher l'endpoint. - Maintenir FIFO unique ; `onChunk` filtre d'abord l'enveloppe Slimbk post-header, sinon enfile le vrai chunk. - `onError` décode d'abord le protocole premier-header V1, sinon mémorise transport sans throw ; callback idempotent. - Chaque tâche FIFO tente Activity, son et emit séparément, poursuit les chunks et agrège les erreurs dans l'ordre. - Après call + FIFO : déterminer completed/transport, finaliser seulement si réservation possédée, arrêter le son par attempt et appeler `notifyPromptSourceSettled(...)` une seule fois. ### 52)

`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.queryDoneSound.test.ts` — modifié -

Couvrir origine exacte, panneau remplacé, sourceId/soundAttemptId, Activity auxiliaire, FIFO, premier header source-error, meta.error body, body Slimbk error post-header, terminal puis transportError, erreurs auxiliaires non court-circuitantes, onError idempotent et conflit de réservation. ### 53)

`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts` — modifié - Vérifier ordre réservation → allocation → endpoint/FIFO → finalisation → son → coordinateur, terminal header dans FIFO et transport sans terminal retirant le run. ### 54)

`AICode/src/extension/src/extension/panels/prompt/PromptQueryDoneSound.manager.ts` — modifié - Corréler par runRef exact ; `onAiStreamStart` retourne `soundAttemptId` monotone. - Exiger runRef+attempt pour chunk/stop, ignorer stale, ne jamais créer implicitement un état. - Dédupliquer terminal par runRef sans réarmer lors d'une reconnexion. - Conserver l'état tant que source/observer actif ; cleanup uniquement via coordinateur. ### 55)

`AICode/src/extension/src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts` — modifié - Couvrir sources successives, attempts du même run, stale, dedup, fermeture et conservation source. ### Activité runtime extension ### 56)

`AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts` — modifié - Renommer la classe interne `SidebarQueryActivityRuntimeService`. - Stocker runRef, ownership, runId, startMs/sourceStartedAtMs, timer, usage et stopRequested par locator. - Exposer une allocation de début synchrone `{ runId, publication }`. - Exposer `onSourceStopRequested(runRef)` : mutation interne/store synchrone, broadcast auxiliaire, guard runRef+runId. - Exposer snapshot readonly pour rehydrate et transition terminale source appelée uniquement par coordinateur. - Source elapsed depuis sourceStartedAtMs ; observer depuis tentative viewer. ### 57)

`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.race.test.ts` — modifié - Couvrir sources successives, stale, open/closed, allocation avant broadcast, échec broadcast, stopRequested exact et fermeture après Stop sans réactivation du bouton. ### 58)

`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.usage.test.ts` — modifié - Couvrir usage, Idle/suppression, missing terminal, publication échouée et arrêt timer exact. ### Coordination panneau Prompt ### 59)

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts` — créé - Parser occurrences brutes, détecter toute query explicite, inconnus `liveSession\*`, doublons locator/live et valider avec `ZLiveSessionPanelDescriptor` shared. - Préserver toute query explicite valide ; rejeter invalide. - Construire normalPrompt avec lease fraîche, sourceId/adaptateur/policy exacts et suppression complète `autoRun\*`. ### 60)

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionPanelQuery.util.test.ts` — créé - Couvrir couples shared, égalité source/turn EasyAgents, ligne requise/interdite, doublons, inconnus, ordre, lease, invalid-before-opener et autoRun. ### 61)

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelCoordinator.service.ts` — créé - Classe injectée sans import low-level, FIFO par locator. - `openPromptPanel`, `onPanelDisposed`, `onSourceSettled`, wake-ups observer/claim. - Préserver explicite valide, rejeter invalide, injecter normalPrompt seulement pour source active/réservée. - Releases adapter-specific ; normalPrompt libère uniquement lease. - Unique propriétaire de la transition terminale Activity et des cleanups revalidés. - Sous claim, conserver mapping détaché et différer tous les cleanups. ### 62)

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionPanelCoordinator.service.test.ts` — créé - Couvrir priorités query, races open/close/settle, missing terminal, Idle ouvert, observer wake-up, claim wake-up, réouverture/nouvelle source et cleanup unique. ### 63)

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionDetachedStop.service.ts` — créé - Valider synchroniquement projection source/closed exacte, source active/réservée et claim d'abort disponible. - Partager la transition `onSourceStopRequested` du runtime Activité, puis abort sans await intermédiaire. - Ne jamais abort une source différente depuis une projection stale. ### 64)

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionDetachedStop.service.test.ts` — créé - Couvrir accept/refus, reopened, remplacement, double clic, réservation, store stale, claim destructif et ordre stopRequested/abort. ### 65)

`AICode/src/extension/src/extension/panels/prompt/PromptThreadCleanup.manager.ts` — modifié - Indexer par locator complet ; séparer register, detach, cleanup historique, cleanup terminal et orphan cleanup. - Ne jamais abort/clear une source ou un observer actif. - Conserver mapping détaché pendant les cleanups différés et sous claim ; revalider après chaque await. ### 66)

`AICode/src/extension/src/extension/panels/prompt/\_\_test\_\_/PromptThreadCleanup.manager.attachmentsCleanup.test.ts` — modifié - Couvrir scopes, detach, terminal, nouvelle source, observer wake-up, claim wake-up, attachments et idempotence. ### 67)

`AICode/src/extension/src/extension/panels/core/openPanel.util.ts` — modifié - Conserver le nom public `openCentralPanel(...)` mais le rendre `async`. - Extraire un low-level opener privé synchrone ; construire le coordinateur module-scoped avec ce port injecté. - Prompt passe par le coordinateur ; autres pages passent directement par le low-level. - Exporter `notifyPromptSourceSettled(...)` pour le handler source. - Capturer le descripteur initial ; au dispose, supprimer synchroniquement le panneau du registre avant toute délégation. - Adapter titres/Activity à `items/status`, injecter subscriptions et runtime, préserver absence de cycle. ### 68)

`AICode/src/extension/src/extension/panels/core/\_\_test\_\_/openPanel.util.test.ts` — modifié - Await façade, wiring sans cycle, exact descriptor, reveal, dispose order, sourceSettled, subscriptions, claim et pages non-Prompt. ### 69)

`AICode/src/extension/src/messageBus/handlers/ui/uiGenericHandlers.ts` — modifié - Await `openCentralPanel` ; Prompt est coordonné, autres pages inchangées. ### 70)

`AICode/src/extension/src/messageBus/handlers/ui/\_\_test\_\_/uiGenericHandlers.openExternalUrl.test.ts` — modifié - Mock facade async et préserver URL. ### 71)

`AICode/src/extension/src/extension/commands/NewChat.command.ts` — modifié - Await ouverture coordonnée. ### 72)

`AICode/src/extension/src/extension/commands/\_\_test\_\_/NewChat.command.test.ts` — modifié - Vérifier await et propagation erreur. ### 73)

`AICode/src/extension/src/extension/uriHandlers/registerDeepLinkUriHandler.ts` — modifié -

Deep-link chat via façade coordonnée en fire-and-forget avec catch explicite. ### 74)

`AICode/src/extension/src/extension/uriHandlers/\_\_test\_\_/registerDeepLinkUriHandler.scope.test.ts` — modifié - Couvrir scopes, façade et guards. ### SidebarStore et Activité ### 75)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.ts` — modifié - Transitions atomiques `panelOpened`, `sourceStarted`, `sourceStopRequested`, `sourceSettled`. - SourceStarted reçoit run/runtime déjà alloué ; publication séparée. - Mapping final non-stale : open → Idle, closed → remove ; stale/source différente no-op. ### 76)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.mutations.manager.ts` — modifié - `windows → items`, `panelClosed`, `forceRemove`, elapsed et stopRequested exacts. - Conserver uniquement Running/source fermé ; lors de panelClosed, dériver stopRequested du snapshot registre/runtime exact. ### 77)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.core.manager.ts` — modifié - Initialiser items et appeler `rehydrateQueryActivities`. ### 78)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.rehydrate.manager.ts` — modifié - Fusionner panneaux, observers actifs et sources/réservations ; dédupliquer. - Interdire observer/closed et Idle/closed. - Conserver terminal bufferisé non finalisé en Running ; panneau ouvert sans record en Idle. - Reconstituer title/openedAt par runtime/store/panneau/persistence/default dans cet ordre, sans faire du SidebarStore l'autorité des timers. ### 79)

`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.rehydrate.manager.test.ts` — modifié - Couvrir toutes projections, publication manquante, réservation, observer fermé, terminal pré-forwarding, missing terminal et stopRequested. ### 80)

`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.manager.queryRunningUpsert.test.ts` — modifié - Tester union, transitions, publication auxiliaire, outcomes stricts et sourceStopRequested. ### 81)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.chatTitles.manager.ts` — modifié - Lire items et préserver fallback title source fermée/running. ### 82)

`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.handlers.manager.ts` — modifié - Enregistrer open activity et Stop détaché ; déléguer coordinateur/service. ### 83)

`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts` — créé - Vérifier payloads, accepted et délégation. ### 84) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts` — modifié - Lire items et autorité liveSession. - Acquérir claim avant fermeture ; fermer/retraiter projection sous claim. - Delete durable sans cleanup concurrent ; éviction seulement après succès avec handle exact. - Release en finally ; pas d'éviction préventive sur échec disque. ### 85) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts` — créé - Couvrir store stale, réservation, claim indisponible, start concurrent refusé, handle exact, succès/échec, tombstones et cleanup différé. ### 86)

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/list.endpoint.windowsLock.eperm.test.ts` — modifié - Adapter fixtures items. ### 87) `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/diskAliases.lastWins.endpoints.test.ts` — modifié - Adapter items et préserver delete disque. ### 88)

`AICode/src/extension/src/aiTools/orch/context/persistence/\_\_test\_\_/ChatPersistenceWriter.windowsFallback.test.ts` — modifié - Adapter fixture items. ### 89) `AICode/src/extension/src/aiTools/orch/context/memory/core/mem/\_\_test\_\_/OrchMemory.naive\_title.user\_prompt\_specif\_multiline.e2e.repro.test.ts` — modifié - Adapter fixture items. ### 90)

`AICode/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx` — modifié - Consommer items/union, ouvrir par nouveau RPC et afficher Stop troisième ligne uniquement source/closed. - `SquareStop`, small, danger, label Stop, disableAfterClick, enabled `!stopRequested`, testIds stables. - Callbacks via `useCallback`, objets via `useMemo`, aucun hook dans map ; sous-composant Stop keyé par sourceId. ### 91) `AICode/src/frontend/src/app/sidebar/sidebar-activity/\_\_test\_\_/SidebarActivityQuery.test.tsx` — modifié - Couvrir projections, open RPC, Stop exact, désactivation, disparition et remount source A → B sans labels fragiles. ### Frontend `liveSession` et Prompt ### 92)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/NormalPromptLiveSession.adapter.ts` — créé - Binder descripteur normalPrompt shared et politiques not\_ready/replayFailed/direct handoff/composer interactif. - Aucun Stop de bannière ; Stop compositeur unique. - Initial baseline = hydratation complète ; retries = preserveLocal. ### 93)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/NormalPromptLiveSession.adapter.test.ts` — créé - Couvrir parsing/binding, politiques, surfaces, bannière, handoff et preserveLocal. ### 94)

`AICode/src/frontend/src/app/prompt/liveSession/adapters/PromptLiveSessionBootstrap.adapter.ts` — créé - Dispatcher pur recevant snapshot brut ; getAll sur paramètres critiques. - Importer liste/union shared ; détecter partiels, inconnus, doublons et locator ambigu. - Retourner classic/valid/invalid, locator strict éventuel et bool explicite ; aucune lecture globale. ### 95) `AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/PromptLiveSessionBootstrap.adapter.test.ts` — créé - Couvrir dispatch, doublons identiques/différents, inconnus, adaptateur, égalité EasyAgents, invalid sans locator et pureté. ### 96)

`AICode/src/frontend/src/app/prompt/liveSession/core/PromptLiveSession.types.ts` — modifié - Ajouter `liveSessionReplayFailed`, sourceId et politiques explicites : composer, Send, historique, cleanup, Stop, handoff, snapshot policy, bannière. - Séparer `PromptLiveSessionBannerViewModel` de la surface composer. - Représenter `directAfterReplay` vs `terminalSnapshot` et replayOpenAttemptId vs streamInstanceId. ### 97)

`AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx` — modifié - Consommer le modèle de bannière dédié, compatible viewer EasyAgents et bannière normalPrompt au-dessus du composer. - Conserver status/details/Stop/Retry, callbacks stables et testIds. ### 98)

`AICode/src/frontend/src/app/prompt/liveSession/components/\_\_test\_\_/PromptLiveSessionBanner.test.tsx` — modifié - Adapter fixtures et ajouter Retry normalPrompt sans Stop. ### 99)

`AICode/src/frontend/src/app/prompt/liveSession/core/usePromptLiveSession.hook.ts` — modifié - Maintenir

replayOpenAttemptId pour bootstrap/retry/timers/unmount. - Parser d'abord enveloppe ; full parse pour hydrateFromSnapshot, session opaque pour preserveLocal. - Retry not\_ready normalPrompt ;  
 not\_found/transport/completion sans terminal/éviction → replayFailed. - Démarrer observer uniquement après start accepté et conserver streamInstanceId exact. - Handoff direct sans second Open après terminal data + completed + queues + finalization exacte. - Retry terminal sans stream fictif ; preserveLocal pour messages/stores seulement. - Gérer transportStopAvailable et suppression explicite sur terminal/not\_found/reset/Stop. ### 100)  
 `AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.hydration.test.tsx` — modifié - Couvrir politiques adaptateur, full vs preserveLocal, draft malformé/getters non lus et stale Open/timer/unmount. ### 101)  
 `AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.handoff.test.tsx` — modifié - Couvrir EasyAgents terminal, normal direct, Queue B, stale finalization, terminal data + transportError, source error data et absence de second Open. ### 102)  
 `AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.normalPrompt.test.tsx` — créé - Couvrir not\_ready/not\_found, surface, draft/attachments/nextTurnId, Stop, retry, terminal direct, retry terminal sans stream, enveloppe draft opaque, replayFailed abortable et stale attempts. ### 103)  
 `AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.normalPromptIsomorphism.test.tsx` — créé - Comparer stream/replay/hydratation pour tous query types, thinking, tools, usage, edits, erreurs, EOF et transport après terminal. ### 104)  
 `AICode/src/frontend/src/app/prompt/liveSession/adapters/EasyAgentsBatchLiveSession.adapter.ts` — modifié - Importer constante shared, ajouter sourceId=turnId/adaptateur, valider union shared et préserver UX/read-only/elapsed. ### 105)  
 `AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/EasyAgentsBatchLiveSession.adapter.test.ts` — modifié - Adapter query/sourceId/dispatcher/constante et policy full hydration. ### 106)  
 `AICode/src/frontend/src/messageBus/AiStreamStartArgs.type.ts` — modifié - Définir état interne runRef+streamInstanceId+outcome, union de retour start et type `AiStreamFallbackPersistedAck { turnId; streamInstanceId }`. - Observer transporte runRef complet, lease, replayMode et policy Stop. - Aucun identifiant frontend-only au backend hors sourceId du runRef. ### 107)  
 `AICode/src/frontend/src/messageBus/useAiStream.hook.ts` — modifié - Single-flight ref synchrone ; IDs générés seulement après acceptation. - Retourner union accept/refuse et ne rien muter sur refus. - Utiliser streamInstanceId comme listener key ; exposer runRef, IDs, outcome, terminal data, transportStopAvailable, Stop state et ack fallback corrélé à streamInstanceId. - Garder callbacks/runMerged avant/après awaits ; invalider avant reset. - Attendre Queue A, Queue B puis commit avant outcome/libération. - onError sans terminal crée SYSTEM\_ERROR ; après terminal conserve terminal unique et outcome transportError. - Stop source via abortSource, observer via abort lease ; disponibilité consommée avant premier await. ### 108)  
 `AICode/src/frontend/src/messageBus/\_\_test\_\_/useAiStream.hook.liveSessionObserver.test.ts` — modifié - Couvrir IDs, routes, stale callbacks/runMerged, Queue B, outcomes, terminal+error sans doublon, transportStopAvailable et ack corrélé. ### 109)  
 `AICode/src/frontend/src/messageBus/\_\_test\_\_/useAiStream.hook.promptSourceIdentity.test.ts` — créé - Couvrir génération IDs après acceptation, double start même tick, refus sans mutation, observer sourceId fourni, Stop result et callbacks stale. ### 110)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.hook.ts` — modifié - Propager start results, runRef/streamInstance/outcome/terminal/transportStopAvailable/fallback ack corrélé. - StartLiveObserver prépare/committe placeholders seulement après acceptation. - Lier base thinking à streamInstanceId accepté. - Exposer `finalizedStreamInstanceId`, clear transport availability et reset complet. ### 111)  
 `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunController.liveSessionObserver.test.tsx` — modifié - Couvrir accept/refuse, aucun placeholder sur refus, IDs, base thinking, Stop post-transport et finalization ack. ### 112)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptCancelController.hook.ts` — modifié - Séparer isStreaming et transportStopAvailable. - UI-only cancel seulement si les deux faux ; sinon Stop exact unique. - Pour replayFailed hors transport, finir le handshake après réponse Stop et conserver disponibilité false. ### 113)  
 `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptCancelController.stopCore.test.tsx` — modifié - Ajouter cas `isStreaming=false + transportStopAvailable=true`, double clic et absence de mutation source UI-only. ### Continuation et thinking ### 114)  
 `AICode/src/frontend/src/app/prompt/Prompt/preparePromptContinuationStreaming.util.ts` — créé - Préparation pure : assistant non-thinking exact, bases assistant/thinking, UID réutilisé, placeholder thinking unique, séparateur et resets planifiés sans mutation. ### 115)  
 `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/preparePromptContinuationStreaming.util.test.ts` — créé - Couvrir bases, assistant manquant, séparateur, sources successives, aucune duplication/mutation. ### 116)  
 `AICode/src/frontend/src/app/prompt/Prompt/preparePromptStreamingTurn.util.ts` — modifié - Séparer plan pur et commit ; reset base thinking/séparateur pour replace. ### 117)  
 `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/preparePromptStreamingTurn.util.test.ts` — modifié - Vérifier plan non committé sur refus et resets replace. ### 118)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptThinkingLifecycle.hook.ts` — modifié - Initialiser base thinking par streamInstanceId, append/séparateur unique, reset explicite et cleanup seulement si base vide. ### 119)  
 `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptThinkingLifecycle.continuation.test.tsx` — créé - Couvrir base, append, tool/text avant thinking, réinsertion, reset et retry même source/nouvelle tentative. ### 120)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptToolUse.hook.ts` — modifié - Ne retirer la bulle que si baseThinking vide ; éviter doubles séparateurs. ### 121)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.finalization.hook.ts` — modifié - Garder toutes mutations par streamInstanceId, reset bases et publier finalizedStreamInstanceId après finalisation complète. ### 122)  
 `AICode/src/frontend/src/app/prompt/Prompt/usePromptRunCommands.hook.ts` — modifié - Plans purs + start synchrone

+ commit après acceptation pour Send/SPECIF/Retry/Edit/Compact/Reply/Continue. - Retourner le résultat strict au caller ; supprimer async artificiels. - Sur refus, zéro mutation durable et `uiPending=false` dans la même pile. ### 123)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.compactContextNow.test.tsx` — modifié - Adapter acceptation et confirmer replace sans base héritée. ### 124)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.retryQuery.attachmentOnly.test.tsx` — modifié - Adapter acceptation et préserver attachment-only. ### 125)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.retryQuery.compactContextNow.test.tsx` — modifié - Adapter acceptation et préserver mapping compact. ### 126)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.continueResponse.test.tsx` — créé - Vérifier continuation, thinking unique, même turn/nouvelle tentative et refus sans mutation. ### Intégration Prompt et documentation ### 127)

`AICode/src/frontend/src/app/prompt/Prompt/Prompt.tsx` — modifié - Séparer frontière bootstrap externe et runtime interne résolu. - Lire une seule fois le snapshot brut ; aucune décision live via hooks scalaires. - Invalid sans locator : dialog/close sans contrôleur, hydratation ni autoRun. - Dédire séparément composer, Send, Stop, historique, cleanup et bannière. - Autoriser draft/attachments pendant normalPrompt replay selon policy. - Utiliser transportStopAvailable dans locks/hotkeys/cancel. - Après Send/Edit accepté, stocker `{ turnId, streamInstanceId }` dans la barrière de fallback et seulement alors armer submitTxn/focus. - Effet unique : ack matching → clear historique ; finalization matching sans ack → démasquer/restaurer focus sans effacer texte, attachments, nextTurnId, pendingEditTurnId ou maximized. - Toute query live explicite désactive hydratation classique et autoRun. ### 128)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.liveSessionBootstrap.test.tsx` — modifié - Couvrir snapshot brut, doublons, invalid sans runtime, autoRun neutralisé, surface/replay failure et refus start sans barrières. ### 129)

`AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.tsx` — modifié - Supporter composer seul, viewer banner remplaçant composer, ou composer + bannière normalPrompt. - Bannière normalPrompt au-dessus du textarea, sans second Stop ; VscTextarea reste Stop unique. - Séparer showComposer, edit, Send lock et banner model. ### 130)

`AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.module.css` — modifié - Ajouter wrapper/espacement du bandeau normalPrompt, variables VS Code uniquement, aucune CSS inline. ### 131)

`AICode/src/frontend/src/app/prompt/PromptInput/\_\_test\_\_/PromptInput.batchLiveViewerMode.test.tsx` — modifié - Préserver viewer EasyAgents ; ajouter composer normalPrompt + Retry banner sans Stop, attachments et Send lock. ### 132)

`AICode/src/frontend/src/app/easy-agents/easy-agents-batch/core/\_\_test\_\_/EasyAgentsBatch.liveWatch.test.tsx` — modifié - Adapter query source/adaptateur et constante shared. ### 133)

`AICode/src/frontend/src/app/easy-agents/easy-agents-item/core/batch/\_\_test\_\_/EasyAgentBatchSurface.test.tsx` — modifié - Adapter query exacte sans changement UX. ### 134)

`AICode/src/frontend/src/app/prompt/hydration/isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md` — modifié - Documenter fermeture/réouverture, continuation/thinking, finalization, erreur premier-header, meta.error tardive, enveloppe Slimbk post-header filtrée, persistance de tout meta.error avant append, terminal+transportError unique, direct handoff et preserveLocal opaque. ### 135)

`AICode/src/frontend/src/app/prompt/thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md` — modifié - Documenter base thinking continuation, séparateur unique, guards text/tool et streamInstanceId. ### Contrats transverses frontend ### 136)

`AICode/src/frontend/src/app/prompt/Prompt/applyHydratedPromptSessionState.util.ts` — modifié - Union discriminée `hydrateFromSnapshot | preserveLocal`. - Full : comportement actuel. - Preserve : aucun setter composer, aucune lecture/validation/normalisation de draft/attachments/nextTurnId ; construire un FrontChatSession assaini uniquement depuis champs consommés. - Getters piégés sur champs ignorés ne doivent pas être exécutés. - Retour `void` ; doc refs messages toujours enregistrées. ### 137)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/applyHydratedPromptSessionState.util.test.ts` — modifié - Adapter full et couvrir preserveLocal, getters non lus, messages/stores/thread/reset/scroll et retour void. ### 138)

`AICode/src/frontend/src/app/prompt/Prompt/usePromptSessionHydration.hook.ts` — modifié - Appeler `hydrateFromSnapshot` ; désactiver pour toute query live explicite. ### 139)

`AICode/src/frontend/src/app/prompt/hydration/frontSessionHydration.type.ts` — modifié - Généraliser `allowOrphanToolResults=true` aux baselines live-safe EasyAgents et normalPrompt, sans modifier pairing. ### 140)

`AICode/src/frontend/src/app/prompt/lib/usePromptSessionLocator.hook.ts` — modifié - Ajouter `usePromptRouterQueryStringRawSnapshot()` via `useMemo(..., [])`. - Conserver parseurs locator existants pour autres consommateurs ; documenter que le snapshot est la voie bootstrap Prompt. ### 141)

`AICode/src/frontend/src/app/prompt/lib/\_\_test\_\_/usePromptSessionLocator.hook.test.tsx` — modifié - Ajouter harness prouvant snapshot figé après mutation globale/rerender. ### 142)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx` — modifié - Mock snapshot brut classique et union d'acceptation ; préserver autoRun classique. ### 143)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.editLastPrompt.noThumbTurnIdDrift.test.tsx` — modifié - Mock snapshot ; acceptation synchrone Edit ; préserver turnId thumbnails jusqu'à ack. ### 144)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.editLastPrompt.restoreAttachments.test.tsx` — modifié - Mock snapshot et préserver restauration attachments. ### 145)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.ensureNextTurnId.stability.test.tsx` — modifié - Mock snapshot/acceptation et préserver nextTurnId. ### 146)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.fallbackPersistAck.clearAttachments.test.tsx` — modifié - Vérifier clear seulement sur ack exact et conservation attachments/nextTurnId/pending edit lors d'une finalization sans ack. ### 147)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.fallbackPersistAck.clearInput.test.tsx` — modifié - Vérifier masque armé après acceptation, clear sur ack exact, démasquage sans clear sur finalization sans ack et focus

restauré. ### 148) `AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.promptDraft.hydraton.test.tsx` — modifié - Mock snapshot/acceptation et préserver debounce/hydraton. ### 149)

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.startAcceptance.test.tsx` — créé - Harness commun sept commandes ; refus sans mutation, acceptation avec un seul scaffold et résultat synchrone exact. ### 150)

`AICode/src/frontend/src/messageBus/prompt-stream-dispatch-lock.programming-doc-readme.md` — modifié - Documenter streamInstance guards, Queue A+B idle, commit final, reset stale interdit et terminal+transportError sans doublon. ### 151)

`AICode/src/extension/src/aiTools/orch/ask/stream/core/fallback-persist-and-title.programming-doc-readme.md` — modifié - Documenter barrières armées après acceptation. - Ajouter le cas accepted puis finalization sans fallback ack : démasquer sans effacer le brouillon/attachments/nextTurnId/edit state. ### Atomicité destructive Retry EasyAgents ### 152)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunRetry.service.ts` — modifié - Après guards non destructifs, acquérir claim exact juste avant fermeture. - Tenir le claim pendant fermeture, éviction pré-delete, UI remove et delete. - Passer handle exact à éviction ; release en finally sur toute issue. - Préserver toutes règles Git/reconstruction/viewer. ### 153)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunRetry.testHarness.ts` — modifié - Ajouter mocks stricts claim/release/autorité/evict(handle), sans casts inutiles. ### 154)

`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunRetry.guards.service.test.ts` — modifié - Couvrir claim indisponible, ordre, handle exact, release succès/erreurs, source concurrente refusée et cleanup différé. ### Barrières de transport et origine exacte révélées par la réinspection ### 155)

`AICode/src/sharedlib/src/slimbk/framework/parseSlimbkStreamBodyError.util.ts` — créé - Détecter avant toute logique métier une enveloppe `SlimbkResult` non-200 apparue dans le body après un header 200. - Valider status connu, errorMessage string et errorCode optionnel ; toute forme avec un champ top-level `status` incohérent devient une erreur de protocole, jamais un `AiStreamedChunk`. - Ne jamais interpréter cette voie comme protocole source V1, même si texte/code coïncident. ### 156)

`AICode/src/sharedlib/src/slimbk/framework/\_\_test\_\_/parseSlimbkStreamBodyError.util.test.ts` — créé - Couvrir enveloppes valides, 200 ignoré, statuts/codes malformés, objets/chunks ordinaires et collision volontaire avec le préfixe V1. ### 157)

`AICode/src/extension/src/messageBus/core/extTypedMessageBus.util.ts` — modifié - Définir un `StreamHandlerOriginContext` extension-only contenant `htmlPage` et la webview exacte. - Étendre `onStreamed`, `dispatchStreamedCall` et `getStreamHandler` pour le transmettre comme troisième argument sans toucher aux contrats sérialisés. ### 158)

`AICode/src/extension/src/messageBus/core/extMessageBus.util.ts` — modifié - Construire le contexte d'origine dans `initializeViewWithMessageBus` et le transmettre au chemin stream. - Exiger ce contexte pour un stream webview ; les appels RPC non-stream restent inchangés. ### 159)

`AICode/src/extension/src/messageBus/core/\_\_test\_\_/extMessageBus.streamOriginContext.test.ts` — créé - Prouver que le handler reçoit la webview émettrice exacte, que deux panneaux Prompt ont des contextes distincts et qu'aucune donnée n'est ajoutée au payload sérialisé. ### 160)

`AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.module.css` — modifié - Mettre à jour le commentaire de maintenance : le bandeau n'est plus uniquement read-only ni forcément rendu à la place du compositeur. - Conserver les variables VS Code et le style existant, sauf ajustement strictement nécessaire au modèle réutilisable. ## Récapitulatif exact des chemins de fichiers ### Créés -

`AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionReplay.type.ts` -

`AICode/src/extension/src/aiTools/orch/context/memory/core/msg/\_\_test\_\_/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts` -

`AICode/src/extension/src/aiTools/orch/liveSession/source/LiveSessionSourceLifecycle.service.ts` -

`AICode/src/extension/src/aiTools/orch/liveSession/source/\_\_test\_\_/LiveSessionSourceLifecycle.service.test.ts` -

`AICode/src/extension/src/aiTools/orch/liveSession/source/NormalPromptLiveSessionSource.service.ts` -

`AICode/src/extension/src/aiTools/orch/liveSession/source/\_\_test\_\_/NormalPromptLiveSessionSource.service.test.ts` -

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abortSource.endpoint.ts` -

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abortSource.endpoint.test.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelQuery.util.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionPanelQuery.util.test.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionPanelCoordinator.service.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionPanelCoordinator.service.test.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/PromptLiveSessionDetachedStop.service.ts` -

`AICode/src/extension/src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionDetachedStop.service.test.ts` -

`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts` -

`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts` -

`AICode/src/frontend/src/app/prompt/liveSession/adapters/NormalPromptLiveSession.adapter.ts` -

`AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/NormalPromptLiveSession.adapter.test.ts` -

`AICode/src/frontend/src/app/prompt/liveSession/adapters/PromptLiveSessionBootstrap.adapter.ts` -

`AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/PromptLiveSessionBootstrap.adapter.test.ts` -

`AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.normalPrompt.test.tsx` -

`AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.normalPromptIsomorphism.test.tsx` -

`AICode/src/frontend/src/messageBus/\_\_test\_\_/useAiStream.hook.promptSourceIdentity.test.ts` -

`AICode/src/frontend/src/app/prompt/Prompt/preparePromptContinuationStreaming.util.ts` -

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/preparePromptContinuationStreaming.util.test.ts` -

`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptThinkingLifecycle.continuation.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.continueResponse.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.startAcceptance.test.tsx` -  
`AICode/src/sharedlib/src/slimbk/framework/parseSlimbkStreamBodyError.util.ts` -  
`AICode/src/sharedlib/src/slimbk/framework/\_\_test\_\_/parseSlimbkStreamBodyError.util.test.ts` -  
`AICode/src/extension/src/messageBus/core/\_\_test\_\_/extMessageBus.streamOriginContext.test.ts` ### Modifiés -  
`AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionRun.type.ts` -  
`AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionPanel.type.ts` -  
`AICode/src/sharedlib/src/types/liveSessionTypes/LiveSessionOpenResult.type.ts` -  
`AICode/src/sharedlib/src/types/liveSessionTypes/\_\_test\_\_/LiveSessionContracts.test.ts` -  
`AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts` -  
`AICode/src/sharedlib/src/types/aiTypes/\_\_test\_\_/aiStopSuffix.util.test.ts` -  
`AICode/src/extension/src/aiTools/orch/context/memory/core/msg/OrchMessagesMemoryAssistantPatch.ts` -  
`AICode/src/sharedlib/src/types/busTypes/ui/SidebarStore.ts` -  
`AICode/src/sharedlib/src/types/busTypes/ui/SidebarUiMessages.ts` -  
`AICode/src/sharedlib/src/types/busTypes/global/MessageRegistry.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/liveSession.programming-doc-readme.md` -  
`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRecord.type.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/registry/LiveSessionRegistry.service.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/registry/\_\_test\_\_/LiveSessionRegistry.service.test.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/core/LiveSessionService.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/core/\_\_test\_\_/LiveSessionService.test.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/open/LiveSessionOpen.service.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/open/\_\_test\_\_/LiveSessionOpen.service.test.ts` -  
`AICode/src/extension/src/aiTools/orch/liveSession/replay/\_\_test\_\_/LiveSessionReplay.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/EasyAgentsBatchLiveSessionAdapter.service.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/\_\_test\_\_/EasyAgentsBatchLiveSessionAdapter.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunExecution.service.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.live-activity-and-gating.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.commit-and-recovery-failures.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.post-commit-persistence.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunChildExecution.service.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunChildExecution.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunManager.service.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunExecution.testHarness.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunLifecycle.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchLastViewerStop.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/EasyAgentsBatchRunRetry.service.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunRetry.testHarness.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/batch/run/\_\_test\_\_/EasyAgentsBatchRunRetry.guards.service.test.ts` -  
`AICode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/open.endpoint.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/open.endpoint.test.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/stream.endpoint.test.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/abort.endpoint.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abort.endpoint.test.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/stream.endpoint.ts` -  
`AICode/src/extension/src/slimbk-local/endpoints/ai/orch/query/ask/\_\_test\_\_/stream.endpoint.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/liveSessionStreamHandlers.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/liveSessionStreamHandlers.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/iaStreamHandlers.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.queryDoneSound.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts` -  
`AICode/src/extension/src/extension/panels/prompt/PromptQueryDoneSound.manager.ts` -  
`AICode/src/extension/src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.race.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.usages.test.ts` -  
`AICode/src/extension/src/extension/panels/prompt/PromptThreadCleanup.manager.ts` -  
`AICode/src/extension/src/extension/panels/prompt/\_\_test\_\_/PromptThreadCleanup.manager.attachmentsCleanup.test.ts` -

`AICode/src/extension/src/extension/panels/core/openPanel.util.ts` -  
`AICode/src/extension/src/extension/panels/core/\_\_test\_\_/openPanel.util.test.ts` -  
`AICode/src/extension/src/messageBus/handlers/ui/uiGenericHandlers.ts` -  
`AICode/src/extension/src/messageBus/handlers/ui/\_\_test\_\_/uiGenericHandlers.openExternalUrl.test.ts` -  
`AICode/src/extension/src/extension/commands/NewChat.command.ts` -  
`AICode/src/extension/src/extension/commands/\_\_test\_\_/NewChat.command.test.ts` -  
`AICode/src/extension/src/extension/uriHandlers/registerDeepLinkUriHandler.ts` -  
`AICode/src/extension/src/extension/uriHandlers/\_\_test\_\_/registerDeepLinkUriHandler.scope.test.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.queryWindow.mutations.manager.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.core.manager.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.rehydrate.manager.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.rehydrate.manager.test.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.manager.queryRunningUpsert.test.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.chatTitles.manager.ts` -  
`AICode/src/extension/src/extension/panels/sidebar/SidebarStore.handlers.manager.ts` - `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts` - `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/list.endpoint.windowsLock.eperm.test.ts` - `AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/diskAliases.lastWins.endpoints.test.ts` - `AICode/src/extension/src/aiTools/orch/context/persistence/\_\_test\_\_/ChatPersistenceWriter.windowsFallback.test.ts` -  
`AICode/src/extension/src/aiTools/orch/context/memory/core/mem/\_\_test\_\_/OrchMemory.naive\_title.user\_prompt\_specif\_multiline.e2e.repro.test.ts` - `AICode/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx` -  
`AICode/src/frontend/src/app/sidebar/sidebar-activity/\_\_test\_\_/SidebarActivityQuery.test.tsx` -  
`AICode/src/frontend/src/app/prompt/liveSession/core/PromptLiveSession.types.ts` -  
`AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx` -  
`AICode/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.module.css` -  
`AICode/src/frontend/src/app/prompt/liveSession/components/\_\_test\_\_/PromptLiveSessionBanner.test.tsx` -  
`AICode/src/frontend/src/app/prompt/liveSession/core/usePromptLiveSession.hook.ts` -  
`AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.hydratation.test.tsx` -  
`AICode/src/frontend/src/app/prompt/liveSession/core/\_\_test\_\_/usePromptLiveSession.handoff.test.tsx` -  
`AICode/src/frontend/src/app/prompt/liveSession/adapters/EasyAgentsBatchLiveSession.adapter.ts` -  
`AICode/src/frontend/src/app/prompt/liveSession/adapters/\_\_test\_\_/EasyAgentsBatchLiveSession.adapter.test.ts` -  
`AICode/src/frontend/src/messageBus/AiStreamStartArgs.type.ts` -  
`AICode/src/frontend/src/messageBus/useAiStream.hook.ts` -  
`AICode/src/frontend/src/messageBus/\_\_test\_\_/useAiStream.hook.liveSessionObserver.test.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunController.liveSessionObserver.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptCancelController.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptCancelController.stopCore.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/preparePromptStreamingTurn.util.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/preparePromptStreamingTurn.util.test.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptThinkingLifecycle.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptToolUse.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptRunController.finalization.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptRunCommands.hook.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.compactContextNow.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.retryQuery.attachmentOnly.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/usePromptRunCommands.retryQuery.compactContextNow.test.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt.tsx` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/Prompt.liveSessionBootstrap.test.tsx` -  
`AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.tsx` -  
`AICode/src/frontend/src/app/prompt/PromptInput/PromptInput.module.css` -  
`AICode/src/frontend/src/app/prompt/PromptInput/\_\_test\_\_/PromptInput.batchLiveViewerMode.test.tsx` -  
`AICode/src/frontend/src/app/easy-agents/easy-agents-batch/core/\_\_test\_\_/EasyAgentsBatch.liveWatch.test.tsx` -  
`AICode/src/frontend/src/app/easy-agents/easy-agents-item/core/batch/\_\_test\_\_/EasyAgentBatchSurface.test.tsx` -  
`AICode/src/frontend/src/app/prompt/hydration/isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md` - `AICode/src/frontend/src/app/prompt/thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md` -  
`AICode/src/frontend/src/app/prompt/Prompt/applyHydratedPromptSessionState.util.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/\_\_test\_\_/applyHydratedPromptSessionState.util.test.ts` -  
`AICode/src/frontend/src/app/prompt/Prompt/usePromptSessionHydration.hook.ts` -  
`AICode/src/frontend/src/app/prompt/hydration/frontSessionHydration.type.ts` -  
`AICode/src/frontend/src/app/prompt/lib/usePromptSessionLocator.hook.ts` -  
`AICode/src/frontend/src/app/prompt/lib/\_\_test\_\_/usePromptSessionLocator.hook.test.tsx` -

```
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.noThumbTurnIdDrift.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.editLastPrompt.restoreAttachments.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.ensureNextTurnId.stability.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearAttachments.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.fallbackPersistAck.clearInput.test.tsx` -
`AICode/src/frontend/src/app/prompt/Prompt/__test__/Prompt.promptDraft.hydrate.test.tsx` -
`AICode/src/frontend/src/messageBus/prompt-stream-dispatch-lock.programming-doc-readme.md` -
`AICode/src/extension/src/aiTools/orch/ask/stream/core/fallback-persist-and-title.programming-doc-readme.md` -
`AICode/src/extension/src/messageBus/core/extTypedMessageBus.util.ts` -
`AICode/src/extension/src/messageBus/core/extMessageBus.util.ts` #### Déplacés Aucun. #### Supprimés Aucun. ##
Facteur de risque de régression 1. **Risque calculé : 92 %.** Criticité `5`, couplage `5`, blast radius `5`, incertitude `4`,
faible confiance `3`. Base = `10×5 + 3×5 + 2×5 + 2×4 + 3×3 = 92` ; risque = `round(92 × 5/5) = 92 %`. 2. **Zones
complexes : ** start transactionnel, exactitude webview d'origine, sources successives, réservations/époques, Queue A/B,
callbacks stale, terminal puis transportError, enveloppe Slimbk post-header, protocole V1 premier-header, persistance de
tout meta.error avant append, finalisation post-forwarding, retraites/tombstones, claims destructifs, wake-ups,
continuation/thinking, parse query brut, frontière React, preserveLocal sans lecture de draft, transition Activité unique,
stopRequested partagé, son par attempt, delete/Retry sous claim et démasquage du compositeur sans fallback ack. 3.
**Justification : ** la tranche modifie le chemin normal de toutes les requêtes Prompt et plusieurs coordinations
asynchrones. Les corrections ajoutées par cette réinspection ferment quatre courses particulièrement dangereuses : un
ancien message de stream ne peut plus démarrer sur un nouveau panneau du même locator ; une exception post-header
Slimbk ne peut plus être prise pour un `AiStreamedChunk` ; un `meta.error` ne devient handoffable qu'après persistance
exacte, y compris le cas auto-accept ; et une tentative acceptée terminée avant `fallback_persisted` ne laisse plus le
compositeur masqué indéfiniment. Les identités exactes, finalisations, claims, wake-ups et tests d'isomorphisme restent
indispensables pour éviter perte, duplication, résurrection ou cleanup concurrent. ## Résumé - Étendre `liveSession` aux
sources Prompt normales sans tuer la source à la fermeture. - Ajouter sourceId backend exact, streamInstanceId frontend et
replayOpenAttemptId Open-only. - Rendre start synchrone/transactionnel et toutes les commandes sans mutation sur refus.
- Vérifier la webview d'origine exacte, pas seulement le locator, avant toute réservation. - Réserver avant await, bufferiser
avant forwarding, finaliser seulement après FIFO extension et Queue A/B frontend. - Persister tout meta.error avant append
; encoder seulement le premier meta.error dans le header V1. - Filtrer les enveloppes Slimbk d'erreur post-header avant
qu'elles ne contaminent la pipeline de chunks. - Conserver un terminal data unique lorsqu'un transportError suit. -
Finaliser en handoff uniquement avec baseline + terminal durable ; retirer sinon. - Séparer retraites, tombstones, époque de
run, claim destructif et génération destructive. - Sérialiser Activité/cleanup par locator ; Idle si panneau ouvert, suppression
si fermé. - Propager stopRequested à Activité pour que le Stop détaché reste désactivé après fermeture. - Reconnecter
Prompt avec composer interactif, Send verrouillé, Stop unique et Retry banner sans Stop. - Handoff direct sans second
Open ; retries preserveLocal sans lire les champs draft. - Démasquer sans effacer le compositeur si une tentative acceptée
se finalise sans fallback ack. - Préserver EasyAgents, timers, son, queues, locks, sanitizers et fichiers générés. ## Nom du
commit `feat(live-session): keep prompt runs alive across panel close and reopen`
</SPECIF_FINALISER_LIVESESSION_POUR_LES_SESSIONS_PROMPT_NORMALES_20260712_204731>
</USER>
<ASSISTANT>
```

**Prêt, Joel** 

J'ai **lu intégralement la spécification** et assimilé ses invariants, son architecture, ses contrats d'identité, ses courses asynchrones, ainsi que son périmètre de **160 fichiers**.

Comme demandé, je n'ai effectué **aucune vérification anticipée** du dépôt, de la compilation, du lint ou des tests.

Je suis prêt pour la **feedback loop** : nous traiterons factuellement chaque erreur jusqu'à obtenir **compilation, lint et tests entièrement verts, sans tricher ni dégrader les invariants**. 

</ASSISTANT>

<USER>

```
----- Linting sharedlib ... ----- src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(87,24): error
TS2339: Property 'status' does not exist on type 'never'. src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(88,24): error
TS2339: Property 'errorCode' does not exist on type 'never'. src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(104,17):
error TS2698: Spread types may only be created from object types.
src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(111,39): error TS2339: Property 'errorMessage' does not exist on type
'never'. src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(116,17): error TS2698: Spread types may only be created from
object types. src/types/aiTypes/__test__/aiStopSuffix.util.test.ts(122,17): error TS2698: Spread types may only be created
from object types. src/types/aiTypes/aiStopSuffix.util.ts(161,5): error TS2322: Type '{ status: string; errorCode:
"AICODE_LIVE_SESSION_SOURCE_ERROR_V1"; errorMessage: string; }' is not assignable to type 'never'.
</USER>
```

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧 ✅ scanning problems

🔧 ✅ reading project: 📁 g "[Docs techniques](#)

[proches](#)",

📄 [aiStopSuffix.util.ts](#), 📄 [aiStopSuffix.util.test.ts](#), 📄 [aiStopSuffix.util.ts](#), 📄 [SlimbkResult.type.ts](#), 🔍 e "[Typages](#)

[SlimbkResult](#)", 🔍 e "[Statut serveur](#)

[Slimbk](#)", 📄 [parseSlimbkStreamBodyError.util.ts](#), 📄 [parseSlimbkStreamBodyError.util.test.ts](#)

🔧 ✅ patching file: 📄 [aiStopSuffix.util.ts](#)

[AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts](#)

Copy

Review

Accept

PatchFull file

```
- 'AICODE_LIVE_SESSION_SOURCE_ERROR_V1' as const;
-
+ 'AICODE_LIVE_SESSION_SOURCE_ERROR_V1' as const;
+
+ /** Exact non-success Slimbk envelope emitted by the private first-header source error
+ protocol. */
+ type LiveSessionSourceErrorV1Envelope = Exclude<
+   SlimbkResult,
+   { status: '200_OK' }
+ > & {
+   readonly status: '500_SERVER_ERROR';
+   readonly errorCode: typeof LIVE_SESSION_SOURCE_ERROR_V1_CODE;
+ };
+
```

---

```
- value: AiErrorMetaValue,
- ): Extract<SlimbkResult, { status: '500_SERVER_ERROR' }> {
-
+ value: AiErrorMetaValue,
+ ): LiveSessionSourceErrorV1Envelope {
+
```

[AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts](#)

Copy

Review

Accept

🔧 ✅ reading project: 📄 [aiStopSuffix.util.ts](#) [L45-L180]

### Correctif appliqué 🔧

J'ai appliqué **LLM-DEBUG-3** : les tests révélaient ici un défaut de typage dans le code source, et non une erreur dans leurs assertions.

### Cause racine

`SlimbkResult` regroupe tous les statuts d'erreur dans **une seule branche d'union** dont la propriété `status` contient plusieurs littéraux. Par conséquent, le type :

```
Extract<SlimbkResult, { status: '500_SERVER_ERROR' }>
```

était évalué à `never` : la branche globale des erreurs n'est pas entièrement assignable à un objet exigeant uniquement le statut `500_SERVER_ERROR`.

Cela expliquait directement les sept diagnostics :

- retour de la fonction incompatible avec `never` ;
- propriétés `status`, `errorCode` et `errorMessage` absentes sur `never` ;
- spreads impossibles depuis une valeur de type `never`.

J'ai corrigé [aiStopSuffix.util.ts](#) en définissant une enveloppe dédiée qui :

- exclut proprement la branche `200_OK` ;
- restreint ensuite `status` à `500_SERVER_ERROR` ;
- rend le code protocolaire V1 obligatoire et littéral ;

- reste structurellement compatible avec `SlimbkResult` ;
- n'utilise aucun cast, `any`, contournement ou affaiblissement du test.

Le test `aiStopSuffix.util.test.ts` reste inchangé : ses assertions sont légitimes et doivent maintenant être correctement typées.

Relance le lint de `sharedlib` et envoie-moi la prochaine sortie, même si elle est longue : nous poursuivrons jusqu'au vert complet.

Résumé des changements

| Élément                         | Changement                                                                                 | Complétude |
|---------------------------------|--------------------------------------------------------------------------------------------|------------|
| Typage de l'enveloppe Slimbk V1 | Remplacement de l' <code>Extract</code> impossible par une branche d'erreur Slimbk affinée | ✓          |
| Statut protocolaire             | <code>500_SERVER_ERROR</code> demeure un littéral exact                                    | ✓          |
| Code protocolaire               | <code>AICODE_LIVE_SESSION_SOURCE_ERROR_V1</code> demeure obligatoire et exact              | ✓          |
| Tests existants                 | Aucun test assoupli ou modifié pour masquer le problème                                    | ✓          |
| Dette technique                 | Aucun cast, <code>any</code> , fallback ou suppression de contrôle                         | ✓          |
| Nom du commit                   |                                                                                            |            |

text

Copy

fix(live-session): type source error envelope without never extraction

🎵 Le `never` quitte enfin le serveur 🚫 Le protocole garde toute sa rigueur 🎵

</ASSISTANT>

<USER>

C:\dev\AICode\src\sharedlib\src\types\aiTypes\aiStopSuffix.util.ts 104:23 error 'value' may use Object's default stringification format ('[object Object]') when stringified @typescript-eslint/no-base-to-string ✖ 1 problem (1 error, 0 warnings)

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧 ✓ scanning problems

🔧 ✓ reading project: [aiStopSuffix.util.ts](#) [L85-L120], [aiStopSuffix.util.test.ts](#) [L65-L135], [safeJSONStringify.util.ts](#), [Normalisation erreurs](#), [JSON stringify sûr](#), [aiStopSuffix.util.test.ts](#) [L1-L70]

🔧 ✓ patching file: [aiStopSuffix.util.ts](#)  
[AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts](#)

Copy

Review

Accept

PatchFull file

```
-     } catch {
-         return String(value);
-
+     } catch {
+         // BigInt is a valid diagnostic primitive, but JSON.stringify rejects it.
+         if (typeof value === 'bigint') return value.toString();
+
+         // Avoid Object's opaque "[object Object]" fallback for cyclic or throwing
+         // values.
+         return '[unserializable]';
+

```

[AICode/src/sharedlib/src/types/aiTypes/aiStopSuffix.util.ts](#)

Copy  
Review  
Accept

**Correction appliquée** 

La cause était le fallback `String(value)` dans `aiStopSuffix.util.ts`. Comme `value` est de type `unknown`, ESLint détectait qu'un objet pouvait produire la chaîne inutile `"[object Object]"`.

Le fallback est maintenant strict et déterministe :

- un `bigint`, que `JSON.stringify` refuse nativement, utilise son `toString()` sûr ;
- un objet cyclique ou dont la sérialisation lève une exception produit `"[unserializable]"` ;
- aucun cast, `any`, contournement ESLint ou affaiblissement du typage.

| Élément                                                                     | Résultat                                        | Complétude                                                                          |
|-----------------------------------------------------------------------------|-------------------------------------------------|-------------------------------------------------------------------------------------|
| Erreur <code>no-base-to-string</code>                                       | Suppression de la conversion ambiguë d'un objet |  |
| Support de <code>bigint</code>                                              | Valeur textuelle préservée explicitement        |  |
| Objet non sérialisable                                                      | Fallback déterministe et informatif             |  |
| Qualité du typage                                                           | Aucun cast ni assouplissement                   |  |
| Portée                                                                      | Uniquement le bloc signalé par le lint          |  |
| Relance le lint de <code>sharedlib</code> et colle-moi la prochaine sortie. |                                                 |                                                                                     |
| <b>Nom du commit</b>                                                        |                                                 |                                                                                     |

text

Copy

```
fix(live-session): safely normalize unserializable error details
```

 Les objets cycliques restent sages  Et le lint repart sans naufrage 

</ASSISTANT>

<USER>

```
src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunLifecycle.service.test.ts(388,13): error TS2322: Type
'{ onSourceStarted: Mock<() => void>; appendChunk: Mock<() => { becameReplayReady: boolean; }>;
markBaselineReady: Mock<() => { becameReplayReady: boolean; }>; markSourceSettled: Mock<...>;
markHandoffReady: Mock<...>; }' is not assignable to type 'Pick<EasyAgentsBatchLiveSessionAdapterService,
"appendChunk" | "markBaselineReady" | "markSourceSettled" | "markHandoffReady" | "onSourceStarted">'. The types
returned by 'onSourceStarted(...)' are incompatible between these types. Type 'void' is not assignable to type
'LiveSessionSourceReservationHandle'. src/aiTools/easyAgents/batch/run/__test__/EasyAgentsBatchRunRetry.live-
restart.service.test.ts(107,13): error TS2345: Argument of type 'unknown' is not assignable to parameter of type
'AsymmetricMatcher_2 | { readonly token: symbol | AsymmetricMatcher_2; }'.
src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts(78,64): error
TS2554: Expected 0 arguments, but got 1.
src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts(87,62): error
TS2554: Expected 0 arguments, but got 1.
src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts(104,68): error
TS2554: Expected 0 arguments, but got 1.
src/aiTools/easyAgents/batch/run/liveSession/__test__/EasyAgentsBatchLiveSessionAdapter.service.test.ts(117,13): error
TS2554: Expected 0 arguments, but got 2. src/extension/panels/core/__test__/openPanel.util.test.ts(19,18): error TS2352:
Conversion of type '{ postMessage: () => Promise<true>; }' to type 'Webview' may be a mistake because neither type
sufficiently overlaps with the other. If this was intentional, convert the expression to 'unknown' first. Type '{ postMessage:
() => Promise<true>; }' is missing the following properties from type 'Webview': options, html, onDidReceiveMessage,
asWebviewUri, cspSource src/extension/panels/core/__test__/openPanel.util.test.ts(156,61): error TS2554: Expected 0
arguments, but got 1. src/extension/panels/core/__test__/openPanel.util.test.ts(176,61): error TS2554: Expected 0
arguments, but got 1. src/extension/panels/prompt/__test__/PromptQueryDoneSound.manager.test.ts(47,13): error TS2322:
Type '{ turnId: string; type: "meta"; value: { metaType: "stop_reason"; value: { reason: AiStopReason; }; }; }' is not
assignable to type 'AiStreamedChunk'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop_reason"; value:
{ reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType:
"thinking_signature"; value: string; }; } | { turnId: string; type: "meta"; value: { metaType: "thinking_section_break";
value: true; }; } | { turnId: string; type: "meta"; value: { ...; }; } | ... 25 more ... | { ...; }'. Type '{ turnId: string; type: "meta";
value: { metaType: "stop_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type:
```

"meta"; value: { metaType: "interact\_gate\_required"; value: { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { ...; }; }; }'. The types of 'value.metaType' are incompatible between these types. Type ""stop\_reason"" is not assignable to type ""interact\_gate\_required"". src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts(52,13): error TS2322: Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type 'AiStreamedChunk'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "thinking\_signature"; value: string; }; } | { turnId: string; type: "meta"; value: { metaType: "thinking\_section\_break"; value: true; }; } | { turnId: string; type: "meta"; value: { ...; }; } | ... 25 more ... | { ...; }'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "interact\_gate\_required"; value: { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { ...; }; }; }'. The types of 'value.metaType' are incompatible between these types. Type ""stop\_reason"" is not assignable to type ""interact\_gate\_required"". src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts(64,13): error TS2322: Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type 'AiStreamedChunk'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "thinking\_signature"; value: string; }; } | { turnId: string; type: "meta"; value: { metaType: "thinking\_section\_break"; value: true; }; } | { turnId: string; type: "meta"; value: { ...; }; } | ... 25 more ... | { ...; }'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "interact\_gate\_required"; value: { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { ...; }; }; }'. The types of 'value.metaType' are incompatible between these types. Type ""stop\_reason"" is not assignable to type ""interact\_gate\_required"". src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts(70,13): error TS2322: Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type 'AiStreamedChunk'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "thinking\_signature"; value: string; }; } | { turnId: string; type: "meta"; value: { metaType: "thinking\_section\_break"; value: true; }; } | { turnId: string; type: "meta"; value: { ...; }; } | ... 25 more ... | { ...; }'. Type '{ turnId: string; type: "meta"; value: { metaType: "stop\_reason"; value: { reason: AiStopReason; }; }; }' is not assignable to type '{ turnId: string; type: "meta"; value: { metaType: "interact\_gate\_required"; value: { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { ...; }; }; }'. The types of 'value.metaType' are incompatible between these types. Type ""stop\_reason"" is not assignable to type ""interact\_gate\_required"".

src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionDetachedStop.service.test.ts(60,23): error TS2352: Conversion of type '{ sidebarContentState: { activityState: { queryState: { items: never[]; }; }; }' to type 'SidebarStore' may be a mistake because neither type sufficiently overlaps with the other. If this was intentional, convert the expression to 'unknown' first. Type '{ sidebarContentState: { activityState: { queryState: { items: never[]; }; }; }' is missing the following properties from type 'SidebarStore': sidebarLoadingState, sidebarUiState

src/extension/panels/prompt/liveSession/\_\_test\_\_/PromptLiveSessionDetachedStop.service.test.ts(67,28): error TS2322: Type 'Mock<UnknownFunction>' is not assignable to type '(runRef: { locator: { sessionId: string; storageScopeRelPath: string; }; turnId: string; sourceId: string; }) => LiveSessionAbortClaim | undefined'. Type 'unknown' is not assignable to type 'LiveSessionAbortClaim | undefined'.

src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts(29,34): error TS2769: No overload matches this call. Overload 1 of 4, '(iterable?: Iterable<readonly [unknown, unknown]> | null | undefined): Map<unknown, unknown>', gave the following error. Argument of type 'unknown[][]' is not assignable to parameter of type 'Iterable<readonly [unknown, unknown]>'. The types returned by '[Symbol.iterator]()>.next(...)' are incompatible between these types. Type 'IteratorResult<unknown[], undefined>' is not assignable to type 'IteratorResult<readonly [unknown, unknown], any>'. Type 'IteratorYieldResult<unknown[]>' is not assignable to type 'IteratorResult<readonly [unknown, unknown], any>'. Type 'IteratorYieldResult<unknown[]>' is not assignable to type 'IteratorYieldResult<readonly [unknown, unknown]>'. Type 'unknown[]' is not assignable to type 'readonly [unknown, unknown]'. Target requires 2 element(s) but source may have fewer. Overload 2 of 4, '(entries?: readonly (readonly [unknown, unknown])[] | null | undefined): Map<unknown, unknown>', gave the following error. Argument of type 'unknown[][]' is not assignable to parameter of type 'readonly (readonly [unknown, unknown])[]'. Type 'unknown[]' is not assignable to type 'readonly [unknown, unknown]'. Target requires 2 element(s) but source may have fewer.

src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts(34,15): error TS2349: This expression is not

src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts(35,59): error TS2554: Expected 0 arguments, but got 2. src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts(44,13): error TS2349: This expression is not callable. Type '{}' has no call signatures.

src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts(46,47): error TS2554: Expected 0 arguments, but got 1. src/extension/uriHandlers/\_\_test\_\_/registerDeepLinkUriHandler.scope.test.ts(106,59): error TS2554: Expected 0 arguments, but got 2. src/extension/uriHandlers/\_\_test\_\_/registerDeepLinkUriHandler.scope.test.ts(134,59): error TS2554: Expected 0 arguments, but got 2. src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts(18,46): error TS2554: Expected 0 arguments, but got 1.

src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts(26,13): error TS2554: Expected 0 arguments, but got 1. src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.race.test.ts(58,56): error TS2554: Expected 0 arguments, but got 1. src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.usage.test.ts(34,26): error TS2739: Type '{ duration\_ms: number; }' is missing the following properties from type '{ start\_ts\_utc: number; end\_ts\_utc: number; duration\_ms: number; }': start\_ts\_utc, end\_ts\_utc

src/messageBus/handlers/ia/\_\_test\_\_/SidebarQueryRunningBridge.usage.test.ts(50,26): error TS2739: Type '{ duration\_ms: number; }' is missing the following properties from type '{ start\_ts\_utc: number; end\_ts\_utc: number; duration\_ms: number; }': start\_ts\_utc, end\_ts\_utc

src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts(134,13): error TS2322: Type 'string | true | { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { gateKind: "mcp\_tool\_authorization"; ... 5 more ...; mcpToolCallArgs?: Record<...> | undefined; } | ... 24 more ... | { ...; }' is not assignable to type '{ input\_tokens?: number | undefined; pruned\_input\_tokens?: number | undefined; output\_tokens?: number | undefined; total\_tokens?: number | undefined; cache\_read\_input\_tokens?: number | undefined; cache\_write\_input\_tokens?: number | undefined; tool\_usage?: AiToolUsageSnakeCase | undefined; query\_time: { ...; }; llm?:?...'. Type 'string' is not assignable to type '{ input\_tokens?: number | undefined; pruned\_input\_tokens?: number | undefined; output\_tokens?: number | undefined; total\_tokens?: number | undefined; cache\_read\_input\_tokens?: number | undefined; cache\_write\_input\_tokens?: number | undefined; tool\_usage?: AiToolUsageSnakeCase | undefined; query\_time: { ...; }; llm?:?...'. Property 'query\_time' does not exist on type 'string | true | { gateKind: "mcp\_oauth"; gateId: string; deadlineEpochMs: number; defaultDecision: "continue" | "auth"; mcpConnectorId: string; mcpExactToolName: string; } | { gateKind: "mcp\_tool\_authorization"; ... 5 more ...; mcpToolCallArgs?: Record<...> | undefined; } | ... 24 more ... | { ...; }'. Property 'query\_time' does not exist on type 'string'.

src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts(51,23): error TS2352: Conversion of type '{ sidebarContentState: { activityState: { queryState: { items: never[]; }; }; chatState: { items: never[]; selectedSessionLocator: undefined; }; }' to type 'SidebarStore' may be a mistake because neither type sufficiently overlaps with the other. If this was intentional, convert the expression to 'unknown' first. Type '{ sidebarContentState: { activityState: { queryState: { items: never[]; }; }; chatState: { items: never[]; selectedSessionLocator: undefined; }; }' is missing the following properties from type 'SidebarStore': sidebarLoadingState, sidebarUiState

src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts(58,21): error TS2352: Conversion of type '{ sidebarContentState: { chatState: { items: never[]; selectedSessionLocator: undefined; }; }; }' to type 'SidebarStore' may be a mistake because neither type sufficiently overlaps with the other. If this was intentional, convert the expression to 'unknown' first. Type '{ sidebarContentState: { chatState: { items: never[]; selectedSessionLocator: undefined; }; }; }' is missing the following properties from type 'SidebarStore': sidebarLoadingState, sidebarUiState

src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts(97,13): error TS2554: Expected 0 arguments, but got 2. src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts(113,50): error TS2554: Expected 0 arguments, but got 1.

src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abort.endpoint.test.ts(49,59): error TS2554: Expected 0 arguments, but got 1. src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/open.endpoint.test.ts(15,16): error TS2345: Argument of type '() => Mock<() => Promise<LiveSessionOpenResult>>' is not assignable to parameter of type 'TestFn'. Type '() => Mock<() => Promise<LiveSessionOpenResult>>' is not assignable to type 'PromiseReturningTestFn'. Type 'Mock<() => Promise<{ session: { meta: { formatVersionNumber: number; sessionId: string; title: string; createdAt: number; updatedAt: number; firstSendState: "preSend" | "postSend" | "notApplicable"; sessionKind?: "chat" | ... 1 more ... | undefined; nextTurnId?: string | undefined; }; messages: unknown[]; promptDraf...' is not assignable to type 'TestReturnValue'.

src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/open.endpoint.test.ts(51,47): error TS2554: Expected 0 arguments, but got 1. src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/stream.endpoint.test.ts(31,49): error TS2554: Expected 0 arguments, but got 1.

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧  scanning problems

🔧  reading project:  [liveSession.programming-doc-readme.md](#),  [easyAgents.programming-doc-readme.md](#),  [EasyAgentsBatchRunLifecycle.service.test.ts](#) [L340-L420],  [EasyAgentsBatchRunRetry.live-restart.service.test.ts](#) [L70-L125],  [EasyAgentsBatchLiveSessionAdapter.service.test.ts](#) [L1-L145],  [openPanel.util.test.ts](#) [L1-L205],  [PromptQueryDoneSound.manager.test.ts](#) [L1-L95],  [PromptLiveSessionDetachedStop.service.test.ts](#) [L1-L100],  [SidebarStore.handlers.manager.test.ts](#),  [registerDeepLinkUriHandler.scope.test.ts](#) [L85-L150],  [iaStreamHandlers.runIdPropagation.test.ts](#) [L1-L65],  [SidebarQueryRunningBridge.race.test.ts](#) [L1-L85],  [SidebarQueryRunningBridge.usage.test.ts](#) [L1-L75],  [SidebarQueryRunningBridge.ts](#) [L1-L175],  [delete.endpoint.liveSession.test.ts](#) [L1-L135],  [abort.endpoint.test.ts](#) [L1-L70],  [open.endpoint.test.ts](#),  [stream.endpoint.test.ts](#) [L1-L55],  [EasyAgentsBatchRunLifecycle.service.test.ts](#) [L1-L45],  [EasyAgentsBatchRunRetry.live-restart.service.test.ts](#) [L1-L75],  
 [EasyAgentsBatchRunRetry.testHarness.ts](#),  
 [EasyAgentsBatchLiveSessionAdapter.service.ts](#),  [LiveSessionService.ts](#) [L1-L240],  
 [AiStreamedChunk.type.ts](#),  
 [PromptQueryDoneSound.manager.ts](#),  
 [SidebarStore.ts](#),  
 [PromptLiveSessionDetachedStop.service.ts](#),  
 [SidebarStore.handlers.manager.ts](#),  [registerDeepLinkUriHandler.scope.test.ts](#) [L1-L90],  e ["Chunks usage existants"](#),  e ["Valeurs query time"](#),  [open.endpoint.ts](#),  [stream.endpoint.ts](#),  [abort.endpoint.ts](#),  [LiveSessionRecord.type.ts](#) [L1-L220],  [SidebarStore.core.manager.ts](#) [L1-L145],  [delete.endpoint.ts](#) [L1-L130],  [openPanel.util.ts](#) [L1-L180],  [PromptLiveSessionPanelCoordinator.service.ts](#) [L1-L95],  [SidebarQueryRunningBridge.ts](#) [L175-L270],  e ["Mocks SidebarStore complets"](#),  e ["Mocks WebviewPanel"](#),  ["Fixtures SidebarStore"](#),  [SidebarStore.queryWindow.listeners.manager.ts](#) [L1-L190],  [SidebarStore.queryWindow.mutations.manager.ts](#) [L1-L190]

🔧  patching file:  [EasyAgentsBatchRunLifecycle.service.test.ts](#)

[AICode/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunLifecycle.service.test.ts](#)

Copy

Review

Accept

PatchFull file

```
-      const liveSessionAdapter = {
-        onSourceStarted: jest.fn(() => {}),
-      }
+      const liveSessionAdapter = {
+        // The execution path must retain the opaque reservation returned by source
+        // registration.
+        onSourceStarted: jest.fn(() =>
+          Object.freeze({
+            token: Symbol('lifecycle-source-reservation'),
+          })),
+      },
+
```

[AICode/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunLifecycle.service.test.ts](#)

Copy

Review

Accept

🔧  patching file:  [EasyAgentsBatchRunRetry.testHarness.ts](#)

[AICode/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunRetry.testHarness.ts](#)

Copy  
Review  
Accept  
PatchFull file

```
-      store: EasyAgentsBatchRunStoreService;  
-  
+      store: EasyAgentsBatchRunStoreService;  
+      destructionClaim: LiveSessionLocatorDestructionClaim;  
+
```

---

```
-      service,  
-      store,  
-  
+      service,  
+      store,  
+      destructionClaim,  
+
```

[AI Code/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunRetry.testHarness.ts](#)

Copy  
Review  
Accept

  patching file:  [EasyAgentsBatchRunRetry.live-restart.service.test.ts](#)

[AI Code/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunRetry.live-restart.service.test.ts](#)

Copy  
Review  
Accept  
PatchFull file

```
-      service,  
-      closePromptPanelForLocatorMock,  
-  
+      service,  
+      destructionClaim,  
+      closePromptPanelForLocatorMock,  
+
```

---

```
-      evictLocatorMock,  
-      acquireLocatorDestructionClaimMock,  
-  
+      evictLocatorMock,  
+
```

---

```
-      CHILD_LOCATOR,  
-      acquireLocatorDestructionClaimMock.mock.results[0]?.value,  
-  
+      CHILD_LOCATOR,  
+      destructionClaim,  
+
```

[AI Code/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunRetry.live-restart.service.test.ts](#)

Copy  
Review  
Accept

  patching file:  [EasyAgentsBatchLiveSessionAdapter.service.test.ts](#)

[AI Code/src/extension/src/aiTools/easyAgents/batch/run/liveSession/ test /  
EasyAgentsBatchLiveSessionAdapter.service.test.ts](#)

Copy  
Review  
Accept

#### PatchFull file

```
- } from '../../../orch/liveSession/registry/LiveSessionRecord.type.js';
-
+ } from '../../../orch/liveSession/registry/LiveSessionRecord.type.js';
+ import type { LiveSessionService } from
+ '../../../orch/liveSession/core/LiveSessionService.js';
+
-
- const liveSession = {
-   reserveSource: jest.fn(() => reservation),
-   registerRun: jest.fn(() => ({ abortRequested: false })),
-   cancelSourceReservation: jest.fn(() => true),
-   appendChunk: jest.fn(() => ({ becameReplayReady: true })),
-   markBaselineReady: jest.fn(() => ({ becameReplayReady: false })),
-   markSourceSettled: jest.fn(() => true),
-   markHandoffReady: jest.fn(() => {}),
-   releaseViewerLease: jest.fn(() => undefined),
-   hasViewerForGroup: jest.fn(() => false),
-   claimViewerAbort: jest.fn(() => undefined),
-   claimSourceAbort: jest.fn(() => undefined),
-   acquireLocatorDestructionClaim: jest.fn(() => destructionClaim),
-   releaseLocatorDestructionClaim: jest.fn(() => true),
-   evictLocator: jest.fn(() => {}),
-   sweep: jest.fn(() => {}),
-
+ const liveSession = {
+   reserveSource: jest.fn<LiveSessionService['reserveSource']>(  
+     () => reservation,  
+   ),  
+   registerRun: jest.fn<LiveSessionService['registerRun']>(() => ({  
+     abortRequested: false,  
+   })),  
+   appendChunk: jest.fn<LiveSessionService['appendChunk']>(() => ({  
+     becameReplayReady: true,  
+   })),  
+   markBaselineReady: jest.fn<LiveSessionService['markBaselineReady']>(  
+     () => ({ becameReplayReady: false })),  
+   ),  
+   markSourceSettled: jest.fn<LiveSessionService['markSourceSettled']>(  
+     () => true,  
+   ),  
+   markHandoffReady: jest.fn<LiveSessionService['markHandoffReady']>(),  
+   releaseViewerLease: jest.fn<LiveSessionService['releaseViewerLease']>(),  
+   hasViewerForGroup: jest.fn<LiveSessionService['hasViewerForGroup']>(  
+     () => false,  
+   ),  
+   claimViewerAbort: jest.fn<LiveSessionService['claimViewerAbort']>(),  
+   acquireLocatorDestructionClaim: jest.fn<  
+     LiveSessionService['acquireLocatorDestructionClaim']  
+   >(() => destructionClaim),  
+   releaseLocatorDestructionClaim: jest.fn<  
+     LiveSessionService['releaseLocatorDestructionClaim']  
+   >(() => true),  
+   evictLocator: jest.fn<LiveSessionService['evictLocator']>(),  
+   sweep: jest.fn<LiveSessionService['sweep']>(),  
+ }
```

[AICode/src/extension/src/aiTools/easyAgents/batch/run/liveSession/\\_test\\_/](#)

[EasyAgentsBatchLiveSessionAdapter.service.test.ts](#)

Copy

Review

Accept

  patching file:  [openPanel.util.test.ts](#)

[AI/Code/src/extension/src/extension/panels/core/ test /openPanel.util.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import type * as vscode from 'vscode';
-
+ import type * as vscode from 'vscode';
+ import type { LiveSessionPostForwardingFinalizationResult } from
+ '../../../../../aiTools/orch/liveSession/registry/LiveSessionRecord.type.js';
+ import type { ChatSessionLocator } from
+ '../../../../../sharedlib/types/chatTypes/ChatSessionLocator.type.js';
+ import type { LiveSessionRunRef } from
+ '../../../../../sharedlib/types/liveSessionTypes/LiveSessionRun.type.js';
+
- const loadPanelMock = jest.fn();
- const coordinatorOpenMock = jest.fn(
-   async (args: {
-
+ const loadPanelMock = jest.fn();
+ const coordinatorOpenMock = jest.fn<
+   (args: {
+     locator: ChatSessionLocator;
+
-   port: { open: (query: { name: string; value: string }[]) => void };
-   }) => args.port.open(args.query),
- );
- const coordinatorDisposeMock = jest.fn(async () => {});
- const coordinatorSettledMock = jest.fn(async () => {});
-
+   port: { open: (query: { name: string; value: string }[]) => void };
+   }) => Promise<void>
+ >(async (args) => {
+   args.port.open(args.query);
+ });
+ const coordinatorDisposeMock = jest.fn<
+   (locator: ChatSessionLocator) => Promise<void>
+ >(async () => {});
+ const coordinatorSettledMock = jest.fn<
+   (args: {
+     runRef: LiveSessionRunRef;
+     runId: number;
+     finalization: LiveSessionPostForwardingFinalizationResult;
+   }) => Promise<void>
+ >(async () => {});
+
-   active: false,
-   webview: { postMessage: async () => true } as vscode.Webview,
-
+   active: false,
+   webview: { postMessage: async () => true },
+

```

[AI/Code/src/extension/src/extension/panels/core/ test /openPanel.util.test.ts](#)

Copy

Review

Accept

  patching file:  [PromptQueryDoneSound.manager.test.ts](#)

[AICode/src/extension/src/extension/panels/prompt/ test /PromptQueryDoneSound.manager.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import { beforeEach, describe, expect, it, jest } from '@jest/globals';  
-  
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';  
+  
+ import type { AiStreamedChunk } from  
'../../../../../sharedlib/types/aiTypes/AiStreamedChunk.type.js';  
+
```

---

```
- function endTurn(turnId: string) {  
-  
+  
+ function endTurn(turnId: string): AiStreamedChunk {  
+
```

[AICode/src/extension/src/extension/panels/prompt/ test /PromptQueryDoneSound.manager.test.ts](#)

Copy

Review

Accept



patching file: [registerDeepLinkUriHandler.scope.test.ts](#)

[AICode/src/extension/src/extension/uriHandlers/ test /registerDeepLinkUriHandler.scope.test.ts](#)

Copy

Review

Accept

PatchFull file

```
-  
- const openCentralPanelMock = jest.fn(async () => {});  
-  
+  
+ const openCentralPanelMock = jest.fn(  
+   (  
+     page: string,  
+     query?: { name: string; value: string }[],  
+   ) => Promise<void>  
+ >(async () => {}));  
+
```

[AICode/src/extension/src/extension/uriHandlers/ test /registerDeepLinkUriHandler.scope.test.ts](#)

Copy

Review

Accept



patching file: [iaStreamHandlers.runIdPropagation.test.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/ test /iaStreamHandlers.runIdPropagation.test.ts](#)

Copy

Review

Accept

PatchFull file

```
-  
- const allocateStartMock = jest.fn(() => ({  
-   runId: 7,  
-   publication: Promise.resolve(),  
- }));  
-
```

```

+
+ import type { LiveSessionRunRef } from
+ '../../../../../sharedlib/types/liveSessionTypes/LiveSessionRun.type.js';
+
+ const allocateStartMock = jest.fn<
+   (args: {
+     runRef: LiveSessionRunRef;
+     ownership: 'source' | 'observer';
+     sourceStartedAtMs: number;
+     panelState: 'open' | 'closed';
+   }) => { runId: number; publication: Promise<void> }
+ >(() => ({
+   runId: 7,
+   publication: Promise.resolve(),
+ })));
+
AICode/src/extension/src/messageBus/handlers/ia/ test /iaStreamHandlers.runIdPropagation.test.ts
Copy
Review
Accept

```

  **patching file:**  [SidebarQueryRunningBridge.race.test.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/ test /SidebarQueryRunningBridge.race.test.ts](#)

Copy  
Review  
Accept

PatchFull file

```

-
- const sourceStartedMock = jest.fn(async () => {});
- const sourceSettledMock = jest.fn(async () => {});
- const sourceStopRequestedMock = jest.fn(() => true);
- const panelClosedMock = jest.fn(async () => {});
- const setElapsedMock = jest.fn(async () => {});
-
+
+ import type { ChatSessionLocator } from
+ '../../../../../sharedlib/types/chatTypes/ChatSessionLocator.type.js';
+ import type { LiveSessionRunRef } from
+ '../../../../../sharedlib/types/liveSessionTypes/LiveSessionRun.type.js';
+
+ const sourceStartedMock = jest.fn<
+   (args: {
+     runRef: LiveSessionRunRef;
+     ownership: 'source' | 'observer';
+     panelState: 'open' | 'closed';
+     sourceStartedAtMs: number;
+     stopRequested: boolean;
+   }) => Promise<void>
+ >(async () => {});
+ const sourceSettledMock = jest.fn<
+   (runRef: LiveSessionRunRef) => Promise<void>
+ >(async () => {});
+ const sourceStopRequestedMock = jest.fn<
+   (runRef: LiveSessionRunRef) => boolean
+ >(() => true);
+ const panelClosedMock = jest.fn<
+   (args: { locator: ChatSessionLocator }) => Promise<void>
+ >(async () => {});
+ const setElapsedMock = jest.fn<
+   (locator: ChatSessionLocator, ms: number | undefined) => Promise<void>
+ >(async () => {});

```

+  
[AICode/src/extension/src/messageBus/handlers/ia/ test /SidebarQueryRunningBridge.race.test.ts](#)

Copy

Review

Accept

  patching file:  [PromptLiveSessionDetachedStop.service.test.ts](#)

[AICode/src/extension/src/extension/panels/prompt/liveSession/ test /PromptLiveSessionDetachedStop.service.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import { describe, expect, it, jest } from '@jest/globals';
-
-
+ import { describe, expect, it, jest } from '@jest/globals';
+
+ import type { LiveSessionService } from
+ '../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+
- const RUN = { locator: LOCATOR, turnId: 'turn', sourceId: 'source' } as const;
-
+ const RUN = { locator: LOCATOR, turnId: 'turn', sourceId: 'source' } as const;
+
+ /** Build an isolated complete Sidebar store around the Activity items under test. */
+ function createSidebarStore(
+   items: SidebarStore['sidebarContentState']['activityState']['queryState']['items'],
+ ): SidebarStore {
+   return {
+     sidebarLoadingState: {
+       status: { mode: 'ready' },
+       steps: [],
+       showResetAi: false,
+     },
+     sidebarContentState: {
+       llmState: { aiCatalog: { engines: [] } },
+       activityState: {
+         indexState: {
+           isRunning: false,
+           eventType: 'cycle-stop',
+           percent: 0,
+         },
+         queryState: { items },
+       },
+       chatState: { items: [] },
+     },
+     sidebarUiState: {
+       viewScrollTop: {},
+       workspace: { status: 'present' },
+       chatUiState: {
+         viewMode: 'list',
+         openItem: null,
+         renameDrafts: {},
+         load: { text: '' },
+         list: {
+           isLoading: false,
+           hasMore: false,
+           didInitialFetch: true,
+         },
+         search: {
```

```

+         q: '',
+         status: 'idle',
+         runId: null,
+         result: null,
+         error: null,
+         seq: 0,
+       },
+     },
+   },
+ };
+ }
+

```

---

```

-   const order: string[] = [];
-   const store = {
-     sidebarContentState: {
-       activityState: {
-         queryState: {
-           items: [
-             {
-               status: 'running',
-               ownership: 'source',
-               panelState: 'closed',
-               runRef: RUN,
-               sessionId: LOCATOR.sessionId,
-               storageScopeRelPath: '',
-               title: 'Chat',
-               liveQueryElapsedMs: 1,
-               stopRequested: false,
-               updatedAt: 1,
-               openedAt: 1,
-             },
-           ],
-         },
-       },
-     },
-   },
-

```

```

+   const order: string[] = [];
+   const store = createSidebarStore([
+     {
+       status: 'running',
+       ownership: 'source',
+       panelState: 'closed',
+       runRef: RUN,
+       sessionId: LOCATOR.sessionId,
+       storageScopeRelPath: '',
+       title: 'Chat',
+       liveQueryElapsedMs: 1,
+       stopRequested: false,
+       updatedAt: 1,
+       openedAt: 1,
+     },
+   ],
+

```

---

```

-       openedAt: 1,
-     },
-   } as SidebarStore;
-
+       openedAt: 1,
+     },
+   ]);
+

```

---

```

-   it('refuses reopened, stale, and already-requested projections', () => {
-     const claim = jest.fn();
-     const store = {
-       sidebarContentState: {

```

```

-         activityState: { queryState: { items: [] } },
-     },
-     } as SidebarStore;
-
+     it('refuses reopened, stale, and already-requested projections', () => {
+         const claim = jest.fn<LiveSessionService['claimSourceAbort']>(>());
+         const store = createSidebarStore([]);
+

```

[AI Code/src/extension/src/extension/panels/prompt/liveSession/ test /PromptLiveSessionDetachedStop.service.test.ts](#)

Copy

Review

Accept



**patching file:** [SidebarStore.handlers.manager.test.ts](#)

[AI Code/src/extension/src/extension/panels/sidebar/ test /SidebarStore.handlers.manager.test.ts](#)

Copy

Review

Accept

PatchFull file

```

-
- const onMock = jest.fn();
- const openCentralPanelMock = jest.fn(async () => {});
- const stopMock = jest.fn(() => ({ accepted: true }));
-
+
+ import type { ChatSessionLocator } from
+ '../../../../../sharedlib/types/chatTypes/ChatSessionLocator.type.js';
+ import type { LiveSessionRunRef } from
+ '../../../../../sharedlib/types/liveSessionTypes/LiveSessionRun.type.js';
+
+ type SidebarHandlerPayload =
+   | { locator: ChatSessionLocator }
+   | { runRef: LiveSessionRunRef };
+ type SidebarHandler = (payload: SidebarHandlerPayload) => Promise<unknown>;
+
+ const onMock = jest.fn<
+   (messageName: string, handler: SidebarHandler) => void
+ >();
+ const openCentralPanelMock = jest.fn<
+   (
+     page: string,
+     query: { name: string; value: string }[],
+   ) => Promise<void>
+ >(async () => {});
+ const stopMock = jest.fn<
+   (runRef: LiveSessionRunRef) => { accepted: boolean }
+ >(() => ({ accepted: true }));
+
-
-   registerSidebarStoreHandlers();
-   const handlers = new Map(onMock.mock.calls);
-
+   registerSidebarStoreHandlers();
+   const handlers = new Map<string, SidebarHandler>(onMock.mock.calls);
+
-
-   };
-   await handlers.get('sidebar->ext:openQueryActivity')?.({ locator });
-
+   };
+   const openHandler = handlers.get('sidebar->ext:openQueryActivity');
+   if (!openHandler) throw new Error('Missing open activity handler');
+

```

```

+      await openHandler({ locator });
+
-      const runRef = { locator, turnId: 'turn', sourceId: 'source' };
-
+      const runRef = { locator, turnId: 'turn', sourceId: 'source' };
+      const stopHandler = handlers.get('sidebar->ext:stopDetachedQuery');
+      if (!stopHandler) throw new Error('Missing detached Stop handler');
+
-      await expect(
-        handlers.get('sidebar->ext:stopDetachedQuery')?.({ runRef }),
-
+      await expect(
+        stopHandler({ runRef }),
+

```

[AI Code/src/extension/src/extension/panels/sidebar/ test /SidebarStore.handlers.manager.test.ts](#)

Copy

Review

Accept

  patching file:  [SidebarQueryRunningBridge.usage.test.ts](#)

[AI Code/src/extension/src/messageBus/handlers/ia/ test /SidebarQueryRunningBridge.usage.test.ts](#)

Copy

Review

Accept

PatchFull file

```

-      metaType: 'usage',
-      value: { query_time: { duration_ms: 42 } },
-
+      metaType: 'usage',
+      value: {
+        query_time: {
+          start_ts_utc: 10,
+          end_ts_utc: 52,
+          duration_ms: 42,
+        },
+      },
+
-      lastQueryElapsedMs: 42,
-      usage: { query_time: { duration_ms: 42 } },
-
+      lastQueryElapsedMs: 42,
+      usage: {
+        query_time: {
+          start_ts_utc: 10,
+          end_ts_utc: 52,
+          duration_ms: 42,
+        },
+      },
+
-      metaType: 'usage',
-      value: { query_time: { duration_ms: 99 } },
-
+      metaType: 'usage',
+      value: {
+        query_time: {
+          start_ts_utc: 10,
+          end_ts_utc: 109,
+          duration_ms: 99,
+        },
+      },
+

```

+  
+  
+                    },

[AICode/src/extension/src/messageBus/handlers/ia/ test /SidebarQueryRunningBridge.usage.test.ts](#)

Copy

Review

Accept

  patching file:  [SidebarQueryRunningBridge.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts](#)

Copy

Review

Accept

PatchFull file

```
-         if (chunk.type !== 'meta' || chunk.value.metaType !== 'usage') return;
-
+         if (chunk.type !== 'meta' || chunk.value.metaType !== 'usage') return;
+
+         // Capture the discriminated payload before entering the mutation callback so
TypeScript
+         // cannot widen the nested chunk union across the asynchronous closure
boundary.
+         const usage = chunk.value.value;
+
```

---

```
-         if (!item || item.status !== 'running') return;
-         item.usage = chunk.value.value;
-         const elapsed = chunk.value.value.query_time?.duration_ms;
-
+         if (!item || item.status !== 'running') return;
+         item.usage = usage;
+         const elapsed = usage.query_time.duration_ms;
+
```

[AICode/src/extension/src/messageBus/handlers/ia/SidebarQueryRunningBridge.ts](#)

Copy

Review

Accept

  patching file:  [delete.endpoint.liveSession.test.ts](#)

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/ test /delete.endpoint.liveSession.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import { beforeEach, describe, expect, it, jest } from '@jest/globals';
- import type { SidebarStore } from
'../../../../../../../../sharedlib/types/busTypes/ui/SidebarStore.js';
-
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';
+ import type { LiveSessionService } from
'../../../../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+ import type { LiveSessionLocatorDestructionClaim } from
'../../../../../../../../aiTools/orch/liveSession/registry/LiveSessionRecord.type.js';
+ import type { ChatSessionLocator } from
'../../../../../../../../sharedlib/types/chatTypes/ChatSessionLocator.type.js';
+
```

---

```
- const order: string[] = [];
- const claim = Object.freeze({ token: Symbol('delete-claim') });
- const acquireMock = jest.fn(() => claim);
- const evictMock = jest.fn(() => order.push('evict'));
```

```

- const releaseMock = jest.fn(() => {
-
+ const order: string[] = [];
+ const claim: LiveSessionLocatorDestructionClaim = Object.freeze({
+   token: Symbol('delete-claim'),
+ });
+ const acquireMock = jest.fn<
+   LiveSessionService['acquireLocatorDestructionClaim']
+ >(() => claim);
+ const evictMock = jest.fn<LiveSessionService['evictLocator']>(() => {
+   order.push('evict');
+ });
+ const releaseMock = jest.fn<
+   LiveSessionService['releaseLocatorDestructionClaim']
+ >(() => {
+
- });
- const deleteMock = jest.fn(async () => {
-
+ });
+ const deleteMock = jest.fn<
+   (locator: ChatSessionLocator) => Promise<void>
+ >(async () => {
+
-   order.push('delete');
- });
-
+   order.push('delete');
+ });
+ const sidebarStore = {
+   sidebarContentState: {
+     activityState: { queryState: { items: [] } },
+     chatState: { items: [], selectedSessionLocator: undefined },
+   },
+ };
+
-   () => ({
-     sidebarStore: {
-       sidebarContentState: {
-         activityState: { queryState: { items: [] } },
-         chatState: { items: [], selectedSessionLocator: undefined },
-       },
-     } as SidebarStore,
-     updateSidebarStore: async (mutator: (store: SidebarStore) => void) => {
-       mutator({
-         sidebarContentState: {
-           chatState: {
-             items: [],
-             selectedSessionLocator: undefined,
-           },
-         },
-       } as SidebarStore);
-     },
-   }) => ({
+     sidebarStore,
+     updateSidebarStore: async () => {},
+   }) => ({

```

[AI Code/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\\_test\\_/delete.endpoint.liveSession.test.ts](#)

Copy

Review

Accept

  patching file:  [abort.endpoint.test.ts](#)

[AI/Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /abort.endpoint.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import { beforeEach, describe, expect, it, jest } from '@jest/globals';
- import type { LiveSessionAbortClaim } from
'../../../../../../../../aiTools/orch/liveSession/registry/LiveSessionRecord.type.js';
-
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';
+ import type { LiveSessionService } from
'../../../../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+
```

---

```
- const abortMock = jest.fn(() => order.push('abort'));
- const claimViewerAbortMock = jest.fn<() => LiveSessionAbortClaim | undefined>(
-   () => ({
-
+ const abortMock = jest.fn(() => order.push('abort'));
+ const claimViewerAbortMock = jest.fn<LiveSessionService['claimViewerAbort']>(
+   () => ({
+
```

[AI/Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /abort.endpoint.test.ts](#)

Copy

Review

Accept

  patching file:  [open.endpoint.test.ts](#)

[AI/Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /open.endpoint.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- import { beforeEach, describe, expect, it, jest } from '@jest/globals';
- import type { LiveSessionOpenResult } from
'../../../../../../../../sharedlib/types/liveSessionTypes/LiveSessionOpenResult.type.js';
-
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';
+ import type { LiveSessionService } from
'../../../../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+
```

---

```
- const openMock = jest.fn<() => Promise<LiveSessionOpenResult>>(async () => ({
-
+
+ const openMock = jest.fn<LiveSessionService['open']>(async () => ({
+
```

---

```
- describe('liveSession open endpoint', () => {
-   beforeEach(() => openMock.mockClear());
-
+ describe('liveSession open endpoint', () => {
+   beforeEach(() => {
+     openMock.mockClear();
+   });
+
```

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /open.endpoint.test.ts](#)

Copy

Review

Accept

  patching file:  [stream.endpoint.test.ts](#)

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /stream.endpoint.test.ts](#)

Copy

Review

Accept

PatchFull file

```
-
- const replayMock = jest.fn(async function* () {
-   yield { turnId: 'turn', type: 'token' as const, value: 'backlog' };
-   yield { turnId: 'turn', type: 'token' as const, value: 'tail' };
- });
-
+
+ import type { LiveSessionService } from
+ '../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+
+ const replayMock = jest.fn<LiveSessionService['replay']>(async function* () {
+   yield { turnId: 'turn', type: 'token' as const, value: 'backlog' };
+   yield { turnId: 'turn', type: 'token' as const, value: 'tail' };
+ });
+
+
```

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /stream.endpoint.test.ts](#)

Copy

Review

Accept

  reading project:  [EasyAgentsBatchLiveSessionAdapter.service.test.ts](#) [L1-L75],  [openPanel.util.test.ts](#) [L1-L45],  [PromptLiveSessionDetachedStop.service.test.ts](#) [L1-L125],  [SidebarStore.handlers.manager.test.ts](#) [L1-L90],  [SidebarQueryRunningBridge.ts](#) [L112-L155],  [delete.endpoint.liveSession.test.ts](#) [L1-L85],  [open.endpoint.test.ts](#) [L1-L30],  [stream.endpoint.test.ts](#) [L1-L22],  [EasyAgentsBatchRunRetry.live-restart.service.test.ts](#) [L10-L35]

  patching file:  [EasyAgentsBatchRunRetry.live-restart.service.test.ts](#)

[AICode/src/extension/src/aiTools/easyAgents/batch/run/ test /EasyAgentsBatchRunRetry.live-restart.service.test.ts](#)

Grouped

([what is it?](#))

  reading project:  [SidebarQueryRunningBridge.usage.test.ts](#) [L18-L60]

  patching file:  [SidebarQueryRunningBridge.usage.test.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/ test /SidebarQueryRunningBridge.usage.test.ts](#)

Grouped

([what is it?](#))

Lot corrigé 

J'ai appliqué **LLM-DEBUG-3** et lu les documents d'architecture [liveSession](#) et [EasyAgents](#) avant les corrections.

TypeScript nous avait préparé un petit buffet à volonté : **la majorité des erreurs provenaient de mocks Jest dont la signature avait été inférée comme () => ...**, alors que les contrats réels attendent désormais les identités et handles exacts. 😊

Corrections effectuées

Catégorie

Corrections

**Mocks Jest strictement**

Les mocks utilisent maintenant les signatures réelles des services, avec le format `jest.fn<(args) =>`

[EasyAgentsBatchLiveSessionAda](#)

| Catégorie                          | Corrections                                                                                                                                        |                                                     |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| typés                              | <code>Return&gt;()</code> attendu par le projet                                                                                                    |                                                     |
| Réservation source opaque          | <code>onSourceStarted</code> retourne désormais un vrai <code>LiveSessionSourceReservationHandle</code> dans le harness de lifecycle               | <a href="#">EasyAgentsBatchRunLifecycle.se</a>      |
| Claim destructif exact             | Le claim du harness est exposé explicitement ; l’assertion ne lit plus une valeur <code>unknown</code> dans <code>mock.results</code>              | <a href="#">EasyAgentsBatchRunRetry.testHa</a>      |
| Mocks du coordinateur de panneau   | Signatures exactes pour ouverture, dispose et settlement ; suppression du cast direct invalide du faux <code>Webview</code>                        | <a href="#">openPanel.util.test.ts</a>              |
| Chunk terminal du son              | Le helper <code>endTurn</code> est maintenant contextuellement typé comme un vrai <code>AiStreamedChunk</code>                                     | <a href="#">PromptQueryDoneSound.manage</a>         |
| SidebarStore de test complet       | Le test du Stop détaché construit un store strictement valide sans double cast ni objet partiel trompeur                                           | <a href="#">PromptLiveSessionDetachedStop</a>       |
| Handlers Sidebar typés             | Les callbacks enregistrés, l’ouverture Prompt et le Stop détaché ont des signatures appelables et exactes                                          | <a href="#">SidebarStore.handlers.manager.te</a>    |
| Propagation des identités          | Les mocks d’allocation et de settlement acceptent maintenant le <code>runRef</code> et les paramètres attendus                                     | <a href="#">iaStreamHandlers.runIdPropagati</a>     |
| Usage complet                      | Les fixtures fournissent maintenant <code>start_ts_utc</code> , <code>end_ts_utc</code> et <code>duration_ms</code> , conformément au contrat réel | <a href="#">SidebarQueryRunningBridge.usag</a>      |
| Narrowing TypeScript en production | Le payload <code>usage</code> discriminé est capturé avant la closure de mutation, empêchant son élargissement vers toute l’union des métadonnées  | <a href="#">SidebarQueryRunningBridge.ts</a>        |
| Delete atomique                    | Mocks des claims, de l’éviction, de la release et de la suppression durable strictement typés ; suppression des casts partiels de store            | <a href="#">delete.endpoint.liveSession.test.ts</a> |
| Deep link asynchrone               | La façade mockée accepte maintenant la page et la query transmises par le handler                                                                  | <a href="#">registerDeepLinkUriHandler.scop</a>     |

Garanties conservées

- Aucun test n’a été désactivé ou affaibli.
- Aucun `any`, `eslint-disable`, fallback de compatibilité ou cast artificiel ajouté.
- Aucun fichier `*.generated.*` modifié.
- Les assertions d’identité exacte `locator + turnId + sourceId` restent intactes.
- L’assertion du **handle destructif exact** a été renforcée plutôt que remplacée par un matcher vague.
- La correction de production dans le bridge Activité ne change aucun comportement : elle stabilise uniquement le narrowing TypeScript avant la closure asynchrone.
- Le hook Jest de l’endpoint Open ne retourne plus accidentellement l’objet `Mock` depuis `beforeEach`.  
Relance maintenant la même compilation/lint et colle-moi **toute la prochaine sortie**. Nous continuerons erreur par erreur jusqu’au vert complet, sans détourner les contrats.

Nom du commit

text

Copy

fix(live-session): align tests and activity typing with exact contracts

🎵 Les mocks retrouvent leurs arguments 🧑‍🔧 Les handles restent exacts et les types sont contents 🎶

</ASSISTANT>

<USER>

----- Linting frontend ... -----

src/app/sidebar/sidebar-activity/\_\_test\_\_/SidebarActivityQuery.test.tsx(36,22): error TS2322: Type 'Mock<(channel: string, payload: unknown) => Promise<unknown>>' is not assignable to type 'void | TestReturnValuePromise | TestReturnValueGenerator'. Type 'Mock<(channel: string, payload: unknown) => Promise<unknown>>' is missing the following properties from type 'Generator<void, unknown, void>': next, return, throw, [Symbol.iterator], and 12 more.  
src/app/sidebar/sidebar-activity/\_\_test\_\_/SidebarActivityQuery.test.tsx(98,25): error TS2322: Type '{ status: "running"; ownership: "source"; panelState: "open"; sessionId: string; storageScopeRelPath: string; title: string; lastQueryElapsedMs?: number | undefined; usage?: { input\_tokens?: number | undefined; ... 7 more ...; llm?: { ...; } | undefined; } | undefined; updatedAt: number; openedAt: number; } | { ...; }' is not assignable to type 'SidebarActivityQueryItem'. Type '{ status: "running"; ownership: "source"; panelState: "open"; sessionId: string; storageScopeRelPath: string; title: string; lastQueryElapsedMs?: number; usage?: { input\_tokens?: number; pruned\_input\_tokens?: number; ... 6 more ...; llm?: { engine\_id: string; model\_id: string; reasoning?: string; verbosity?: string; ... }' is not assignable to type 'SidebarActivityQueryItem'. Type '{ status: "running"; ownership: "source"; panelState: "open"; sessionId: string; storageScopeRelPath: string; title: string; lastQueryElapsedMs?: number | undefined; usage?: { input\_tokens?: number | undefined; ... 7 more ...; llm?: { ...; } | undefined; } | undefined; updatedAt: number; openedAt: number; }' is missing the following properties from type 'SidebarActivityQuerySourceItem': runRef, liveQueryElapsedMs, stopRequested  
src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx(44,26): error TS2339: Property 'runRef' does not exist on type 'never'.  
src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx(46,14): error TS2339: Property 'runRef' does not exist on type 'never'.  
src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx(55,32): error TS2339: Property 'stopRequested' does not exist on type 'never'.  
src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx(58,53): error TS2339: Property 'runRef' does not exist on type 'never'.  
src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx(178,33): error TS2322: Type 'SidebarActivityQuerySourceItem' is not assignable to type 'never'.

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

  scanning problems

  reading project:  g "[Docs Activité proches](#)",

 [SidebarActivityQuery.tsx](#),

 [SidebarActivityQuery.test.tsx](#),

 [SidebarStore.ts](#),  [VscButton.tsx](#),  [SidebarActivity.tsx](#),  [SidebarActivityIndex.tsx](#)

  patching file:  [SidebarActivityQuery.tsx](#)

[AI/Code/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx](#)

Copy

Review

Accept

PatchFull file

```
- import type {
-   SidebarActivityQueryItem,
-
+ import type {
+   SidebarActivityQueryItem,
+   SidebarActivityQuerySourceItem,
+
- const EMPTY_ITEMS: readonly SidebarActivityQueryItem[] = [];
-
+ const EMPTY_ITEMS: readonly SidebarActivityQueryItem[] = [];
+
+ /** Source activity whose panel is detached while the exact backend source remains
+  * active. */
+ type SidebarActivityQueryDetachedSourceItem =
+   SidebarActivityQuerySourceItem & {
+     readonly panelState: 'closed';
+   };
+ 
```

```

+
-   queryState?: SidebarActivityQueryState;
- }
-
+   queryState?: SidebarActivityQueryState;
+ }
+
+ /** Narrow the activity union to the only projection allowed to expose detached Stop.
+ */
+ function isDetachedSourceItem(
+   item: SidebarActivityQueryItem,
+ ): item is SidebarActivityQueryDetachedSourceItem {
+   // All three discriminants are required: observers never survive close and open
+   sources
+   // must keep using the Stop control already owned by the Prompt composer.
+   return (
+     item.status === 'running' &&
+     item.ownership === 'source' &&
+     item.panelState === 'closed'
+   );
+ }
+
- }: {
-   item: Extract<
-     SidebarActivityQueryItem,
-     { status: 'running'; ownership: 'source'; panelState: 'closed' }
-   >;
-
+ }: {
+   item: SidebarActivityQueryDetachedSourceItem;
+
-   );
-   const detachedSource =
-     item.status === 'running' &&
-     item.ownership === 'source' &&
-     item.panelState === 'closed'
-     ? item
-     : undefined;
-
+   );
+   // Preserve the strict detached-source subtype through JSX so the Stop
+ subtree
+   // cannot accidentally accept an idle, observer-owned, or reopened
+ activity.
+   const detachedSource = isDetachedSourceItem(item)
+     ? item
+     : undefined;
+

```

[AI Code/src/frontend/src/app/sidebar/sidebar-activity/SidebarActivityQuery.tsx](#)

Copy

Review

Accept



patching file: [SidebarActivityQuery.test.tsx](#)

[AI Code/src/frontend/src/app/sidebar/sidebar-activity/ test /SidebarActivityQuery.test.tsx](#)

Copy

Review

Accept

PatchFull file

```

- import { act, fireEvent, render, screen } from '@testing-library/react';
- import type { SidebarActivityQueryState } from
'../../../../../sharedlib/types/busTypes/ui/SidebarStore.js';
-
+ import { act, fireEvent, render, screen } from '@testing-library/react';
+ import type {
+   SidebarActivityQuerySourceItem,
+   SidebarActivityQueryState,
+ } from ' ../../../../../sharedlib/types/busTypes/ui/SidebarStore.js';
+


---


- describe('SidebarActivityQuery', () => {
-   beforeEach(() => busCallMock.mockClear());
-
+ describe('SidebarActivityQuery', () => {
+   beforeEach(() => {
+     // Hooks must not return Jest's mock object as a test completion value.
+     busCallMock.mockClear();
+   });
+


---


-     sourceId: 'source-a',
-   };
-
+     sourceId: 'source-a',
+   };
+   const closedSource: SidebarActivityQuerySourceItem = {
+     status: 'running',
+     ownership: 'source',
+     panelState: 'closed',
+     runRef,
+     ...LOCATOR,
+     title: 'Detached source',
+     liveQueryElapsedMs: 10,
+     stopRequested: false,
+     updatedAt: 2,
+     openedAt: 1,
+   };
+


---


-   const state: SidebarActivityQueryState = {
-     items: [
-       {
-         status: 'running',
-         ownership: 'source',
-         panelState: 'closed',
-         runRef,
-         ...LOCATOR,
-         title: 'Detached source',
-         liveQueryElapsedMs: 10,
-         stopRequested: false,
-         updatedAt: 2,
-         openedAt: 1,
-       },
-     ],
-
+   const state: SidebarActivityQueryState = {
+     items: [closedSource],
+


---


-     expect(stop).toBeDisabled();
-
-
+     expect(stop).toBeDisabled();
+

```

```
+      // Reopening keeps the same source identity but removes the detached Stop
projection.
+      const reopenedSource: SidebarActivityQuerySourceItem = {
+        ...closedSource,
+        panelState: 'open',
+      };
+
```

```
-      queryState={{
-        items: [
-          {
-            ...state.items[0],
-            status: 'running',
-            ownership: 'source',
-            panelState: 'open',
-          },
-        ],
-      },
+      queryState={{
+        items: [reopenedSource],
+      },
```

[AI/Code/src/frontend/src/app/sidebar/sidebar-activity/ test /SidebarActivityQuery.test.tsx](#)

Copy

Review

Accept

### Correctif frontend appliqué

J'ai poursuivi avec **LLM-DEBUG-3**. Les diagnostics provenaient de **deux problèmes de test** et d'un **mauvais affinage de l'union discriminée** dans le composant.

#### Causes racines

##### 1. Extract<...> produisait never

Dans [SidebarActivityQuery.tsx](#), la branche source est déclarée avec :

- ownership: 'source'
- panelState: 'open' | 'closed'

Un Extract demandant directement panelState: 'closed' ne peut pas extraire cette branche entière, car celle-ci accepte également open. TypeScript produisait donc never, rendant ensuite runRef et stopRequested inaccessibles. La correction introduit :

- un sous-type strict représentant une source détachée ;
- un véritable type guard vérifiant les trois discriminants :
- activité running ;
- ownership source ;
- panneau closed ;
- la propagation de ce sous-type jusqu'au composant Stop.

Cette solution maintient l'invariant produit : **le Stop détaché ne peut être rendu ni pour un observer, ni pour une activité Idle, ni pour une source dont le panneau est ouvert.**

##### 2. Le hook beforeEach retournait le mock Jest

Dans [SidebarActivityQuery.test.tsx](#), la fonction fléchée concise retournait implicitement le résultat de mockClear(), donc l'objet Mock lui-même.

Le hook utilise maintenant un bloc explicite et retourne normalement void.

##### 3. Le spread partait de l'union complète

state.items[0] était typé comme l'union globale SidebarActivityQueryItem. En le spreadant pour simuler la réouverture, TypeScript devait envisager qu'il puisse s'agir d'une activité Idle ou observer, dépourvue des champs source obligatoires.

Le test définit désormais explicitement :

- une projection source fermée correctement typée ;
- une projection source ouverte correctement typée ;
- la même identité runRef, sans cast ni reconstruction ambiguë.

#### Résumé des changements

| Élément                   | Correction                                     | État                                                                                  |
|---------------------------|------------------------------------------------|---------------------------------------------------------------------------------------|
| Sous-type source détachée | Intersection stricte avec panelState: 'closed' |  |

| Élément                                                                                            | Correction                                                           | État |
|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|------|
| Affinage de l'union                                                                                | Type guard dédié sur les trois discriminants                         | ✓    |
| Stop détaché                                                                                       | Impossible pour Idle, observer ou source ouverte                     | ✓    |
| Hook Jest                                                                                          | Suppression du retour implicite de <code>mockClear()</code>          | ✓    |
| Fixture source                                                                                     | Objets fermés et rouverts strictement typés                          | ✓    |
| Assertions existantes                                                                              | Stop immédiat et disparition à la réouverture conservés              | ✓    |
| Contournements                                                                                     | Aucun cast, <code>any</code> , désactivation ESLint ou test affaibli | ✓    |
| Relance le lint frontend et transmets-moi la prochaine sortie : on continue jusqu'au vert total. 🔄 |                                                                      |      |
| <b>Nom du commit</b>                                                                               |                                                                      |      |

text

Copy

fix(sidebar): narrow detached source activities without never

🎵 Le `never` quitte la Sidebar 🏠 Le Stop détaché garde le bon territoire 🎵

</ASSISTANT>

<USER>

```

----- Linting frontend ... ----- C:\dev\AICode\src\frontend\src\app\prompt\Prompt\Prompt.tsx
264:11 warning The 'liveSessionBootstrap' conditional could make the dependencies of useMemo Hook (at line 282)
change on every render. To fix this, wrap the initialization of 'liveSessionBootstrap' in its own useMemo() Hook react-
hooks/exhaustive-deps 264:11 warning The 'liveSessionBootstrap' conditional could make the dependencies of useEffect
Hook (at line 314) change on every render. To fix this, wrap the initialization of 'liveSessionBootstrap' in its own
useMemo() Hook react-hooks/exhaustive-deps 856:17 error Unexpected `await` of a non-Promise (non-"Thenable") value
@typescript-eslint/await-thenable 879:13 error Unexpected `await` of a non-Promise (non-"Thenable") value @typescript-
eslint/await-thenable 964:13 error Unexpected `await` of a non-Promise (non-"Thenable") value @typescript-eslint/await-
thenable C:\dev\AICode\src\frontend\src\app\prompt\lib\__test__\usePromptSessionLocator.hook.test.tsx 19:9 error Error:
Cannot reassign variables declared outside of the component/hook Variable `observed` is declared outside of the
component/hook. Reassigning this value during render is a form of side effect, which can cause unpredictable behavior
depending on when the component happens to re-render. If this variable is used in rendering, use useState instead.
Otherwise, consider updating it in an effect. (https://react.dev/reference/rules/components-and-hooks-must-be-pure#side-
effects-must-run-outside-of-render). C:
\dev\AICode\src\frontend\src\app\prompt\lib\__test__\usePromptSessionLocator.hook.test.tsx:19:9 17 | let observed: string
| undefined; 18 | function Harness(): null { > 19 | observed = usePromptRouterQueryStringRawSnapshot(); | ^^^^^^^
`observed` cannot be reassigned 20 | return null; 21 | } 22 | const { rerender } = render(<Harness />); react-hooks/globals
C:\dev\AICode\src\frontend\src\app\prompt\liveSession\components\PromptLiveSessionBanner.tsx 18:58 warning React
Hook useCallback has a missing dependency: 'props'. Either include it or remove the dependency array. However, 'props'
will change when *any* prop changes, so the preferred fix is to destructure the 'props' object outside of the useCallback
call and refer to those specific props inside useCallback react-hooks/exhaustive-deps 19:60 warning React Hook
useCallback has a missing dependency: 'props'. Either include it or remove the dependency array. However, 'props' will
change when *any* prop changes, so the preferred fix is to destructure the 'props' object outside of the useCallback call
and refer to those specific props inside useCallback react-hooks/exhaustive-deps C:
\dev\AICode\src\frontend\src\app\prompt\liveSession\components\__test__\PromptLiveSessionBanner.test.tsx 27:9 error
Promises must be awaited, end with a call to .catch, end with a call to .then with a rejection handler or be explicitly marked
as ignored with the `void` operator @typescript-eslint/no-floating-promises 48:9 error Promises must be awaited, end with
a call to .catch, end with a call to .then with a rejection handler or be explicitly marked as ignored with the `void` operator
@typescript-eslint/no-floating-promises ✖ 10 problems (6 errors, 4 warnings)

```

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧 ✓ scanning problems

🔧 ✓ reading project: 📄 [thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md](#), 📄 [isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-](#)

[readme.md](#), [prompt-stream-dispatch-lock.programming-doc-readme.md](#), [Prompt.tsx](#) [L235-L330], [Prompt.tsx](#) [L820-L990],  
[usePromptSessionLocator.hook.test.tsx](#),  
[usePromptSessionLocator.hook.ts](#),  
[PromptLiveSessionBanner.tsx](#),  
[PromptLiveSessionBanner.test.tsx](#), [PromptLiveSession.types.ts](#), [usePromptRunCommands.hook.ts](#) [L1-L260], ["Définition\\_gateAction"](#), ["Usages\\_gateAction"](#), [Prompt.tsx](#) [L1-L235], [Prompt.tsx](#) [L990-L1110], [PromptLiveSessionBootstrap.adapter.ts](#), [Prompt.liveSessionBootstrap.test.tsx](#) [L1-L180], [usePromptGlobalHotkeys.hook.ts](#), ["Gate action Prompt"](#), [Prompt.tsx](#) [L600-L820], [Prompt.tsx](#) [L430-L610], [Prompt.tsx](#) [L330-L435], [usePromptIndexPending.hook.tsx](#)

  patching file: [Prompt.tsx](#)

[AICode/src/frontend/src/app/prompt/Prompt/Prompt.tsx](#)

Copy

Review

Accept

PatchFull file

```
- import { parsePromptAutoRunUserPromptAugmentationFromQueryString } from
'../lib/promptAutoRunUserPromptAugmentationQuery.util.js';
- import type {
-
+ import { parsePromptAutoRunUserPromptAugmentationFromQueryString } from
'../lib/promptAutoRunUserPromptAugmentationQuery.util.js';
+ import type {
+   PromptLiveSessionBootstrap,
+
-
-   const sessionId: string = sessionLocator.sessionId;
-   const liveSessionBootstrap =
-     dispatch.kind === 'valid'
-       ? dispatch.bootstrap
-       : dispatch.kind === 'invalid'
-       ? {
-         kind: 'invalid' as const,
-         explicitLiveQuery: true as const,
-         dialogTitle: 'Live session bootstrap failed',
-         message: dispatch.message,
-       }
-       : {
-         kind: 'inactive' as const,
-         explicitLiveQuery: false as const,
-       };
-
+   const sessionId: string = sessionLocator.sessionId;
+   const liveSessionBootstrap = useMemo<PromptLiveSessionBootstrap>(() => {
+     // Stabilize the derived descriptor so downstream effects cannot restart merely
because
+     // a defensive inactive/invalid object was reconstructed during an unrelated
render.
+     return dispatch.kind === 'valid'
+       ? dispatch.bootstrap
+       : dispatch.kind === 'invalid'
+       ? {
+         kind: 'invalid',
+         explicitLiveQuery: true,
+         dialogTitle: 'Live session bootstrap failed',
+         message: dispatch.message,
+       }
+       : {
+         kind: 'inactive',
+         explicitLiveQuery: false,
```

```

+         };
+     }, [dispatch]);
+
-
-         gateAction(async () => {
-
+
+         gateAction(() => {
+
-
-         // Start the SPECIF action in the current (new) session.
-         await handleSendSpecifAction({
-
+
+         // Start the SPECIF action in the current (new) session.
+         handleSendSpecifAction({
+
-
-         gateAction(async () => {
-             const autoRunAiQueryOptions: AiQueryOptions = {
-
+
+         gateAction(() => {
+             const autoRunAiQueryOptions: AiQueryOptions = {
+
-
-         // Submit the rendered prompt through the standard Prompt send path.
-         await handleSend(
-
+
+         // Submit the rendered prompt through the standard Prompt send path.
+         handleSend(
+
-
-         const onCompactContextNow = useCallback(): void => {
-             gateAction(async () => {
-                 await handleSend(
-
+
+         const onCompactContextNow = useCallback(): void => {
+             gateAction(() => {
+                 handleSend(
+
+

```

[AI Code/src/frontend/src/app/prompt/Prompt/Prompt.tsx](#)

Copy

Review

Accept

  patching file:  [usePromptSessionLocator.hook.test.tsx](#)

[AI Code/src/frontend/src/app/prompt/lib/ test /usePromptSessionLocator.hook.test.tsx](#)

Error

([what is it?](#))

  patching file:  [PromptLiveSessionBanner.tsx](#)

[AI Code/src/frontend/src/app/prompt/liveSession/components/PromptLiveSessionBanner.tsx](#)

Copy

Review

Accept

PatchFull file

```

- ): JSX.Element {
-     const handleStop = useCallback(() => props.onStop(), [props.onStop]);
-     const handleRetry = useCallback(() => props.onRetry(), [props.onRetry]);
-
+ ): JSX.Element {

```

```
+   const { viewModel, retryCallbackAvailable, onStop, onRetry } = props;
+   const handleStop = useCallback(() => onStop(), [onStop]);
+   const handleRetry = useCallback(() => onRetry(), [onRetry]);
+
```

---

```
-       <div className={styles.label}>
-         {props.viewModel.statusLabel}
-
```

```
+       <div className={styles.label}>
+         {viewModel.statusLabel}
+
```

---

```
-     </div>
-     {props.viewModel.failureDetails ? (
-
```

```
+     </div>
+     {viewModel.failureDetails ? (
+
```

---

```
-       <div className={styles.details}>
-         {props.viewModel.failureDetails}
-
```

```
+       <div className={styles.details}>
+         {viewModel.failureDetails}
+
```

---

```
-     </div>
-     {props.viewModel.showStopAction ? (
-
```

```
+     </div>
+     {viewModel.showStopAction ? (
+
```

---

```
-       <VscButton
-         label={props.viewModel.stopActionLabel}
-
```

```
+       <VscButton
+         label={viewModel.stopActionLabel}
+
```

---

```
-         variant="danger"
-         enabled={props.viewModel.stopEnabled}
-
```

```
+         variant="danger"
+         enabled={viewModel.stopEnabled}
+
```

---

```
-       ) : null}
-       {props.viewModel.showRetryAction ? (
-
```

```
+       ) : null}
+       {viewModel.showRetryAction ? (
+
```

---

```
-       <VscButton
-         label={props.viewModel.retryActionLabel}
-
```

```
+       <VscButton
+         label={viewModel.retryActionLabel}
+
```

---

```
-         enabled={
-           props.viewModel.retryEnabled &&
-           props.retryCallbackAvailable
-
```

```
+         enabled={
+           viewModel.retryEnabled && retryCallbackAvailable
+
```



```

-     });
-     let observed: string | undefined;
-     function Harness(): null {
-         observed = usePromptRouterQueryStringRawSnapshot();
-         return null;
-     }
-     const { rerender } = render(<Harness />);
-     Object.defineProperty(globalThis, '__AICODE_ROUTER_QUERY_STRING__', {
-         configurable: true,
-         value: 'sessionId=bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb&storageScopeRelPath=',
-     });
-     rerender(<Harness />);
-     expect(observed).toBe(
-         'sessionId=aaaaaaaaaaaaaaaaaaaaaaaaaaaa&storageScopeRelPath=',
-     );
- });
-
+ } from '../usePromptSessionLocator.hook.js';
+
-     );
- }
-
+     );
+ }
+
+ /** Render the frozen raw query snapshot without mutating test-owned state during
+ render. */
+ function RawQuerySnapshotHarness(): JSX.Element {
+     const rawQuerySnapshot = usePromptRouterQueryStringRawSnapshot();
+     return <output data-testid="raw-query-snapshot">{rawQuerySnapshot}</output>;
+ }
+
- describe('usePromptSessionLocator', () => {
-
+ describe('usePromptSessionLocator', () => {
+     it('freezes the raw router query snapshot across global mutation and rerender', ()
=> {
+         const initialRawQuery =
+             'sessionId=aaaaaaaaaaaaaaaaaaaaaaaaaaaa&storageScopeRelPath=';
+         setRouterQueryString(initialRawQuery);
+         const { rerender } = render(<RawQuerySnapshotHarness />);
+
+         setRouterQueryString(
+             'sessionId=bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb&storageScopeRelPath=',
+         );
+         act(() => {
+             rerender(<RawQuerySnapshotHarness />);
+         });
+
+         expect(screen.getByTestId('raw-query-snapshot').textContent).toBe(
+             initialRawQuery,
+         );
+     });
+ });

```

[AI Code/src/frontend/src/app/prompt/lib/ test /usePromptSessionLocator.hook.test.tsx](#)

Copy  
Review  
Accept

🔧  reading project:  [Prompt.tsx](#) [L55-L85],  [Prompt.tsx](#) [L255-L320],  [Prompt.tsx](#) [L840-L975],  [usePromptSessionLocator.hook.test.tsx](#) [L1-L90],  [PromptLiveSessionBanner.tsx](#),  [PromptLiveSessionBanner.test.tsx](#)

### Correctifs frontend appliqués

J'ai poursuivi avec **LLM-DEBUG-3** et relu les trois documents d'architecture concernés :

- [invariants thinking et continuation](#)
- [isomorphisme streaming et hydratation](#)
- [verrou de dispatch Queue B](#)

Les dix diagnostics ont été traités sans modifier les invariants fonctionnels. React avait simplement décidé de jouer au contrôleur fiscal des dépendances de hooks. 😊

### Détail des corrections

#### Bootstrap `liveSession` stable

Dans [Prompt.tsx](#) :

- le descripteur dérivé `liveSessionBootstrap` est maintenant créé dans un `useMemo` ;
- les objets défensifs `inactive` et `invalid` ne sont plus reconstruits lors de chaque rendu ;
- les dépendances de `liveSessionReadyBootstrap` et de l'effet de fermeture sont donc stables ;
- aucune ouverture, hydratation ou transition `liveSession` supplémentaire n'est déclenchée.

#### Suppression des `await` invalides

Les commandes suivantes sont désormais **synchrones et transactionnelles** :

- `handleSend`;
- `handleSendSpecifAction`;
- l'envoi `compactContextNow`.

Elles retournent immédiatement `AiStreamStartResult`, conformément au nouveau contrat `start()` synchrone. Les callbacks transmis à `gateAction` sont donc redevenus de simples fonctions `() => void`, sans wrapper `async` artificiel ni `await` sur une valeur non-Thenable.

Cela préserve l'ordre critique :

1. préparation de l'action ;
2. acquisition synchrone du single-flight ;
3. commit du scaffold uniquement si le transport est accepté.

#### Test du snapshot brut React pur

Dans [usePromptSessionLocator.hook.test.tsx](#) :

- la variable externe modifiée pendant le rendu a été supprimée ;
  - le harness rend désormais le snapshot dans un élément observable ;
  - le test vérifie la valeur directement depuis le DOM ;
  - le `rerender` reste exécuté dans `act(...)` ;
  - la mutation de la globale entre les deux rendus reste intacte afin de prouver que le snapshot est réellement figé.
- Le test valide donc toujours que la valeur initiale survit à une mutation ultérieure de la query globale, sans effet de bord pendant le rendu.

#### Callbacks stables du bandeau

Dans [PromptLiveSessionBanner.tsx](#) :

- les propriétés sont destructurées avant les `useCallback` ;
- les dépendances sont limitées à `onStop` et `onRetry` ;
- aucune dépendance globale à l'objet `props` n'est introduite ;
- les comportements Stop et Retry restent indépendants.

#### Promesses `act(...)` correctement attendues

Dans [PromptLiveSessionBanner.test.tsx](#) :

- le test est maintenant asynchrone ;
- les deux interactions utilisateur sont exécutées avec `await act(async () => ...)` ;
- aucune promesse React n'est laissée flottante ;
- les assertions Stop, disparition du Stop et Retry restent inchangées.

### Résumé des changements

| Zone             | Correction                                    | Invariant conservé                       | État                                                                                  |
|------------------|-----------------------------------------------|------------------------------------------|---------------------------------------------------------------------------------------|
| Bootstrap Prompt | Descripteur <code>liveSession</code> mémorisé | Aucun redémarrage sur rendu sans rapport |  |
| Auto-run SPECIF  | Suppression du faux traitement asynchrone     | Start synchrone et scaffold après        |  |

| Zone                                                                    | Correction                                                         | Invariant conservé                        | État |
|-------------------------------------------------------------------------|--------------------------------------------------------------------|-------------------------------------------|------|
| Auto-run Prompt                                                         | Suppression du faux traitement asynchrone                          | acceptation Options et payload identiques | ✓    |
| Compact context                                                         | Suppression du faux traitement asynchrone                          | Passage par le même pipeline Send         | ✓    |
| Test query brute                                                        | Observation via le DOM                                             | Snapshot figé réellement vérifié          | ✓    |
| Bandeau <code>liveSession</code>                                        | Dépendances précises des callbacks                                 | Stop et Retry indépendants                | ✓    |
| Test du bandeau                                                         | Promesses <code>act</code> attendues                               | Interactions React correctement drainées  | ✓    |
| Qualité                                                                 | Aucun cast, <code>any</code> , désactivation lint ou test affaibli | Typage strict maintenu                    | ✓    |
| Relance le lint frontend et transmets-moi la prochaine sortie complète. |                                                                    |                                           |      |
| Nom du commit                                                           |                                                                    |                                           |      |

text

Copy

fix(prompt): stabilize live bootstrap and synchronous run actions

🎵 Le bootstrap reste bien figé 🧱 Les commandes synchrones peuvent avancer 🎵

</ASSISTANT>

<USER>

Summary of all failing tests FAIL

```
src/aiTools/orch/context/memory/core/msg/__test__/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts •
OrchMessagesMemoryAssistantPatch error suffixes › appends system_error exactly once while preserving existing
assistant text expect(jest.fn()).toHaveBeenCalledTimes(expected) Expected number of calls: 2 Received number of calls: 4
74 | 75 | expect(readAssistantText(memory)).toBe(`partial answer${suffix}`); > 76 |
expect(sink).toHaveBeenCalledTimes(2); | ^ 77 | }, 78 | ); 79 | at
src/aiTools/orch/context/memory/core/msg/__test__/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts:76:26 •
OrchMessagesMemoryAssistantPatch error suffixes › appends interrupted exactly once while preserving existing assistant
text expect(jest.fn()).toHaveBeenCalledTimes(expected) Expected number of calls: 2 Received number of calls: 4 74 | 75 |
expect(readAssistantText(memory)).toBe(`partial answer${suffix}`); > 76 | expect(sink).toHaveBeenCalledTimes(2); | ^ 77
| }, 78 | ); 79 | at
src/aiTools/orch/context/memory/core/msg/__test__/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts:76:26 FAIL
src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abortSource.endpoint.test.ts • abortSource endpoint › orders exact
claim then Activity latch then abort expect(received).toEqual(expected) // deep equality - Expected - 2 + Received + 2 -
ObjectContaining { - "accepted": true, + Object { + "accepted": false, } 58 | }, 59 | } as Parameters<typeof
endpoint.syncHandler>[0]; > 60 | expect(result).toEqual(expect.objectContaining({ accepted: true })); | ^ 61 |
expect(order).toEqual(['stopRequested', 'abort']); 62 | }); 63 | at Object.<anonymous>
(src/slimbk-local/endpoints/ai/orch/liveSession/__test__/abortSource.endpoint.test.ts:60:24) FAIL
src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts • chat delete live-session atomicity ›
holds the exact claim through close, durable delete, and eviction expect(received).toEqual(expected) // deep equality -
Expected - 2 + Received + 0 Array [ "close", "removeActivity", "delete", - "evict", - "release", ] 91 | }, 92 | } as
Parameters<typeof endpoint.syncHandler>[0]; > 93 | expect(order).toEqual([ | ^ 94 | 'close', 95 | 'removeActivity', 96 |
'delete', at Object.<anonymous> (src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts:93:23)
• chat delete live-session atomicity › does not evict on durable delete failure and still releases the claim
expect(received).rejects.toThrow(expected) Expected substring: "disk failed" Received message: "Call VsCode.load() first
(extension host only); VS Code API unavailable in worker context" 61 | * Tell if the VS Code API has been loaded in this
process. 62 | */ > 63 | public static isLoading(): boolean { | ^ 64 | return !!VsCode._mod; 65 | } 66 | at VsCode.module
(src/extension/core/VsCode.ts:63:19) at VsCode.get window [as window] (src/extension/core/VsCode.ts:72:26) at
SlimbkEndpoint.handler [as _handler] (src/slimbk-local/endpoints/ai/orch/chat/delete.endpoint.ts:89:25) at
Object.<anonymous> (src/slimbk-local/endpoints/ai/orch/chat/__test__/delete.endpoint.liveSession.test.ts:108:9) 113 | },
```

114 | } as Parameters<typeof endpoint.syncHandler>[0]), > 115 | ).rejects.toThrow('disk failed'); | ^ 116 | expect(evictMock).not.toHaveBeenCalled(); 117 | expect(releaseMock).toHaveBeenCalledWith(claim); 118 | }); at Object.toThrow (node\_modules/expect/build/index.js:2155:20) at Object.<anonymous> (src/slimbk-local/endpoints/ai/orch/chat/\_\_test\_\_/delete.endpoint.liveSession.test.ts:115:19) FAIL

src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts • SidebarStore handlers › registers exact activity open and detached Stop RPCs expect(received).resolves.toEqual(expected) // deep equality Expected: {"accepted": true} Received: undefined 57 | const stopHandler = handlers.get('sidebar->ext:stopDetachedQuery'); 58 | if (!stopHandler) throw new Error('Missing detached Stop handler'); > 59 | await expect(stopHandler({ runRef })).resolves.toEqual({ | ^ 60 | accepted: true, 61 | }); 62 | expect(stopMock).toHaveBeenCalledWith(runRef); at Object.toEqual (node\_modules/expect/build/index.js:2140:20) at Object.<anonymous> (src/extension/panels/sidebar/\_\_test\_\_/SidebarStore.handlers.manager.test.ts:59:56) FAIL

src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abort.endpoint.test.ts • liveSession abort endpoint › claims and aborts the exact run synchronously expect(received).toEqual(expected) // deep equality - Expected - 2 + Received + 2 - ObjectContaining { - "accepted": true, + Object { + "accepted": false, } 46 | }, 47 | } as Parameters<typeof endpoint.syncHandler>[0]); > 48 | expect(result).toEqual(expect.objectContaining({ accepted: true })); | ^ 49 | expect(claimViewerAbortMock).toHaveBeenCalledWith({ 50 | runRef: { 51 | locator: { at Object.<anonymous> (src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/abort.endpoint.test.ts:48:24) FAIL

src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/stream.endpoint.test.ts • liveSession stream endpoint › relays ordered chunks for the exact source id TypeError: Cannot read properties of undefined (reading 'Symbol(Symbol.asyncIterator)') 34 | }); 35 | > 36 | for await (const chunk of getLiveSessionServiceForProd().replay({ | ^ 37 | runRef: { 38 | locator: { 39 | sessionId: args.sessionId, at SlimbkEndpoint.handler [as \_handler] (src/slimbk-local/endpoints/ai/orch/liveSession/stream.endpoint.ts:36:66) at Object.<anonymous> (src/slimbk-local/endpoints/ai/orch/liveSession/\_\_test\_\_/stream.endpoint.test.ts:19:24) FAIL

src/extension/panels/core/\_\_test\_\_/openPanel.util.test.ts • openPanel coordinated facade › routes Prompt through the async coordinator and non-Prompt directly through the low-level opener expect(jest.fn()).toHaveBeenCalledWith(...expected) Expected: "prompt", ArrayContaining [{ "name": "os", "value": "linux" }] Number of calls: 0 135 | }); 136 | expect(coordinatorOpenMock).toHaveBeenCalledTimes(1); > 137 | expect(loadPanelMock).toHaveBeenCalledWith( | ^ 138 | 'prompt', 139 | expect.arrayContaining([ { name: 'os', value: 'linux' } ]), 140 | ); at Object.<anonymous> (src/extension/panels/core/\_\_test\_\_/openPanel.util.test.ts:137:31) • openPanel coordinated facade › removes the exact panel mapping before delegating dispose to the coordinator expect(received).toBeDefined() Received: undefined 162 | const key = `prompt::sessionId=\${'a'.repeat(32)}&storageScopeRelPath=`; 163 | const panel = openedPanelsRegistry.get(key); > 164 | expect(panel).toBeDefined(); | ^ 165 | coordinatorDisposeMock.mockImplementationOnce(async () => { 166 | expect(openedPanelsRegistry.get(key)).toBeUndefined(); 167 | }); at Object.<anonymous> (src/extension/panels/core/\_\_test\_\_/openPanel.util.test.ts:164:23) FAIL

src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts • PromptQueryDoneSound manager › isolates attempts and deduplicates terminal playback by exact runRef TypeError: Cannot read properties of undefined (reading 'catch') 96 | ? { frequencyHz: 1200, durationMs: 120 } 97 | : { frequencyHz: 600, durationMs: 220 }; > 98 | void playBeep(beepArgs).catch((error: unknown) => { | ^ 99 | console.error(formatError(error)); 100 | }); 101 | } at Module.onAiStreamChunk (src/extension/panels/prompt/PromptQueryDoneSound.manager.ts:98:28) at Object.<anonymous> (src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts:45:17) • PromptQueryDoneSound manager › does not let a stale attempt or previous source affect a successor TypeError: Cannot read properties of undefined (reading 'catch') 96 | ? { frequencyHz: 1200, durationMs: 120 } 97 | : { frequencyHz: 600, durationMs: 220 }; > 98 | void playBeep(beepArgs).catch((error: unknown) => { | ^ 99 | console.error(formatError(error)); 100 | }); 101 | } at Module.onAiStreamChunk (src/extension/panels/prompt/PromptQueryDoneSound.manager.ts:98:28) at Object.<anonymous> (src/extension/panels/prompt/\_\_test\_\_/PromptQueryDoneSound.manager.test.ts:68:17) FAIL

src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts • iaStreamHandlers run identity propagation › allocates Activity with exact source identity and preserves the runId for coordinator finalization TypeError: Cannot read properties of undefined (reading 'runId') 31 | panelState: 'open', 32 | }); > 33 | expect(allocation.runId).toBe(7); | ^ 34 | expect(allocateStartMock).toHaveBeenCalledWith( 35 | expect.objectContaining({ runRef }), 36 | ); at Object.<anonymous> (src/messageBus/handlers/ia/\_\_test\_\_/iaStreamHandlers.runIdPropagation.test.ts:33:27) FAIL

src/extension/uriHandlers/\_\_test\_\_/registerDeepLinkUriHandler.scope.test.ts • registerDeepLinkUriHandler › normalizes a legacy chat deep link without storageScopeRelPath to the historical root scope TypeError: Cannot read properties of undefined (reading 'catch') 144 | value: storageScopeRelPath, 145 | }, > 146 | ]).catch((error: unknown) => { | ^ 147 | VsCode.window.showErrorMessage( 148 | `Chat deep-link could not be opened: \${ 149 | error instanceof Error at Object.handleUri (src/extension/uriHandlers/registerDeepLinkUriHandler.ts:146:23) at Object.<anonymous>

```
(src/extension/uriHandlers/__test__/registerDeepLinkUriHandler.scope.test.ts:103:20) • registerDeepLinkUriHandler ›
preserves an explicit nested storageScopeRelPath from the deep link TypeError: Cannot read properties of undefined
(reading 'catch') 144 | value: storageScopeRelPath, 145 | }, > 146 | ]).catch((error: unknown) => { | ^ 147 |
VsCode.window.showErrorMessage( 148 | `Chat deep-link could not be opened: ${ 149 | error instanceof Error at
Object.handleUri (src/extension/uriHandlers/registerDeepLinkUriHandler.ts:146:23) at Object.<anonymous>
(src/extension/uriHandlers/__test__/registerDeepLinkUriHandler.scope.test.ts:131:20) FAIL
src/messageBus/handlers/ia/__test__/iaStreamHandlers.queryDoneSound.test.ts • iaStreamHandlers source lifecycle ›
guards origin and persists meta.error before append while keeping FIFO order expect(received).toBeLessThan(expected)
Expected: < -1 Received: 1 129 | ); 130 | expect(emitted).toHaveLength(2); > 131 |
expect(order.indexOf('persist')).toBeLessThan( | ^ 132 | order.lastIndexOf('append'), 133 | ); 134 |
expect(order.at(-1)).toBe('coordinator'); at Object.<anonymous>
(src/messageBus/handlers/ia/__test__/iaStreamHandlers.queryDoneSound.test.ts:131:42) FAIL
src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.race.test.ts • Test suite failed to run SyntaxError: The
requested module './SidebarStore.queryWindow.listeners.manager.js' does not provide an export named 'panelOpened' 43 |
44 | const { SidebarQueryActivityRuntimeService } = > 45 | await import('./SidebarQueryRunningBridge.js'); | ^ 46 | 47 |
const LOCATOR = { 48 | sessionId: 'a'.repeat(32), at EsmLoader.tryLoadGraphSync
(node_modules/jest-runtime/build/index.js:725:18) at
src/messageBus/handlers/ia/__test__/SidebarQueryRunningBridge.race.test.ts:45:5 Test Suites: 12 failed, 812 passed, 824
total Tests: 16 failed, 2624 passed, 2640 total Snapshots: 1 passed, 1 total Time: 65.222 s, estimated 68 s Ran all test suites.
Command failed: C:\Program Files\nodejs\node.exe --expose-gc --max-old-space-size=8192 --experimental-vm-modules
C:\dev\AICode\src\extension\node_modules\jest\bin\jest.js --config C:\dev\AICode\src\extension\jest.config.ts --
maxWorkers 10 (exit 1)
```

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

  reading project:  [liveSession.programming-doc-readme.md](#),  [fallback-persist-and-title.programming-doc-readme.md](#),

 [jest.config.ts](#),

 [OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts](#),

 [OrchMessagesMemoryAssistantPatch.ts](#),

 [abortSource.endpoint.test.ts](#),

 [abortSource.endpoint.ts](#),

 [delete.endpoint.liveSession.test.ts](#),

 [delete.endpoint.ts](#),

 [SidebarStore.handlers.manager.test.ts](#),

 [SidebarStore.handlers.manager.ts](#),

 [abort.endpoint.test.ts](#),

 [abort.endpoint.ts](#),

 [stream.endpoint.test.ts](#),  [stream.endpoint.ts](#),  [openPanel.util.test.ts](#),  [openPanel.util.ts](#) [L1-

L220],

 [PromptQueryDoneSound.manager.test.ts](#),

 [PromptQueryDoneSound.manager.ts](#),  [iaStreamHandlers.runIdPropagation.test.ts](#),  e ["Allocation Activity export"](#),  [iaStreamHandlers.ts](#) [L1-

L260],  [registerDeepLinkUriHandler.scope.test.ts](#),  [registerDeepLinkUriHandler.ts](#) [L110-

L165],  [iaStreamHandlers.queryDoneSound.test.ts](#),  [SidebarQueryRunningBridge.race.test.ts](#) [L1-

L100],  [SidebarQueryRunningBridge.ts](#) [L1-L120],  [SidebarStore.queryWindow.listeners.manager.ts](#) [L1-

L280],  [openPanel.util.ts](#) [L510-L670],  [SidebarStore.core.manager.ts](#) [L1-L55],  e ["Imports listeners](#)

[Sidebar"](#),  [SidebarQueryRunningBridge.ts](#) [L120-L252]

  patching file:  [OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts](#)

[AICode/src/extension/src/aiTools/orch/context/memory/core/msg/ \\_\\_test\\_\\_ /](#)

[OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts](#)

Copy

Review

Accept

#### PatchFull file

```
- >(async () => {});
- memory.setPersistenceSink(sink);
-
+ >(async () => {});
+
+ // Seed the historical turn before attaching the sink so call-count assertions
isolate
+ // only the two explicit patch operations exercised by each table-driven test case.
+
- memory.addMessage(THREAD_ID, new AssistantMessage(TURN_ID, initialText));
-
+ memory.addMessage(THREAD_ID, new AssistantMessage(TURN_ID, initialText));
+ memory.setPersistenceSink(sink);
+
```

[AICode/src/extension/src/aiTools/orch/context/memory/core/msg/\\_test\\_/OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts](#)

Copy

Review

Accept



patching file: [abortSource.endpoint.test.ts](#)

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/\\_test\\_/abortSource.endpoint.test.ts](#)

Copy

Review

Accept

#### PatchFull file

```
- import { beforeEach, describe, expect, it, jest } from '@jest/globals';
-
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';
+ import type { LiveSessionService } from
+ '../../../../../aiTools/orch/liveSession/core/LiveSessionService.js';
+
- import type { LiveSessionAbortClaim } from
+ import type { LiveSessionAbortClaim } from
+ '../../../../../aiTools/orch/liveSession/registry/LiveSessionRecord.type.js';
-
+ import type { LiveSessionAbortClaim } from
+ '../../../../../aiTools/orch/liveSession/registry/LiveSessionRecord.type.js';
+ import type { LiveSessionRunRef } from
+ '../../../../../sharedlib/types/liveSessionTypes/LiveSessionRun.type.js';
+
- const order: string[] = [];
- const abortMock = jest.fn(() => order.push('abort'));
- const claimSourceAbortMock = jest.fn(() => LiveSessionAbortClaim | undefined)(
-   () => ({
-     runRef: {
-       locator: {
-         sessionId: 'a'.repeat(32),
-         storageScopeRelPath: '',
-       },
-       turnId: 'turn',
-       sourceId: 'source',
-     },
-     abort: abortMock,
-   }),
- );
- const stopRequestedMock = jest.fn(() => {
-   order.push('stopRequested');
-   return true;
- });
```

```

- });
-
+ const order: string[] = [];
+ const abortMock = jest.fn<() => void>();
+ const claimSourceAbortMock = jest.fn<LiveSessionService['claimSourceAbort']>();
+ const stopRequestedMock = jest.fn<(runRef: LiveSessionRunRef) => boolean>();
+
-     order.length = 0;
-     claimSourceAbortMock.mockClear();
-     abortMock.mockClear();
-     stopRequestedMock.mockClear();
-
+     order.length = 0;
+
+     // Jest resetMocks removes module-level implementations, so reconstruct the
exact
+     // synchronous claim → Activity latch → abort contract before every scenario.
+     abortMock.mockImplementation(() => {
+         order.push('abort');
+     });
+     claimSourceAbortMock.mockImplementation(
+         (): LiveSessionAbortClaim => ({
+             runRef: {
+                 locator: {
+                     sessionId: 'a'.repeat(32),
+                     storageScopeRelPath: '',
+                 },
+                 turnId: 'turn',
+                 sourceId: 'source',
+             },
+             abort: abortMock,
+         }),
+     );
+     stopRequestedMock.mockImplementation(() => {
+         order.push('stopRequested');
+         return true;
+     });
+

```

[AI/Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /abortSource.endpoint.test.ts](#)

Copy

Review

Accept



patching file: [abort.endpoint.test.ts](#)

[AI/Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /abort.endpoint.test.ts](#)

Copy

Review

Accept

PatchFull file

```

- const order: string[] = [];
- const abortMock = jest.fn(() => order.push('abort'));
- const claimViewerAbortMock = jest.fn<LiveSessionService['claimViewerAbort']>(
-     () => ({
-         runRef: {
-             locator: {
-                 sessionId: 'a'.repeat(32),
-                 storageScopeRelPath: '',
-             },
-             turnId: 'turn',
-             sourceId: 'source',
-         },
-     });

```

```

-         },
-         abort: abortMock,
-     })),
- );
-
+ const order: string[] = [];
+ const abortMock = jest.fn<() => void>();
+ const claimViewerAbortMock = jest.fn<LiveSessionService['claimViewerAbort']>();
+
+
+     order.length = 0;
+     claimViewerAbortMock.mockClear();
+     abortMock.mockClear();
+
+     order.length = 0;
+
+     // Restore the exact viewer claim after Jest's global resetMocks cleanup.
+     abortMock.mockImplementation(() => {
+         order.push('abort');
+     });
+     claimViewerAbortMock.mockImplementation(() => ({
+         runRef: {
+             locator: {
+                 sessionId: 'a'.repeat(32),
+                 storageScopeRelPath: '',
+             },
+             turnId: 'turn',
+             sourceId: 'source',
+         },
+         abort: abortMock,
+     })));
+

```

[AI Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /abort.endpoint.test.ts](#)

Copy

Review

Accept

  **patching file:**  [stream.endpoint.test.ts](#)

[AI Code/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /stream.endpoint.test.ts](#)

Copy

Review

Accept

PatchFull file

```

- import { describe, expect, it, jest } from '@jest/globals';
-
+ import { beforeEach, describe, expect, it, jest } from '@jest/globals';
+
+
+
+ const replayMock = jest.fn<LiveSessionService['replay']>(async function* () {
+     yield { turnId: 'turn', type: 'token' as const, value: 'backlog' };
+     yield { turnId: 'turn', type: 'token' as const, value: 'tail' };
+ });
+
+
+ const replayMock = jest.fn<LiveSessionService['replay']>();
+
+
+ describe('liveSession stream endpoint', () => {
+
+
+ describe('liveSession stream endpoint', () => {
+     beforeEach(() => {

```

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/liveSession/ test /stream.endpoint.test.ts](#)

## Review

  *patching file:*  [delete.endpoint.liveSession.test.ts](#)

Error

  *patching file:*  [SidebarStore.handlers.manager.test.ts](#)

Copy

## Review

Accept

PatchFull file

[AICode/src/extension/src/extension/panels/sidebar/ test /SidebarStore.handlers.manager.test.ts](#)

## Review

Accept

  *patching file:*  [openPanel.util.test.ts](#)

Copy

## Review

Accept

PatchFull file

```
-         loadPanelMock.mockImplementation(() => createPanel());
-         coordinatorOpenMock.mockClear();
```



```

-     }) => { runId: number; publication: Promise<void> }
-   >(() => ({
-     runId: 7,
-     publication: Promise.resolve(),
-   }));
-
+     panelState: 'open' | 'closed';
+   }) => { runId: number; publication: Promise<void> }
+ >();
+
- describe('iaStreamHandlers run identity propagation', () => {
-
+ describe('iaStreamHandlers run identity propagation', () => {
+   beforeEach(() => {
+     // Restore the allocation result cleared by the extension Jest resetMocks
+     policy.
+       allocateStartMock.mockReturnValue({
+         runId: 7,
+         publication: Promise.resolve(),
+       });
+   });
+
+
+

```

[AICode/src/extension/src/messageBus/handlers/ia/ test /iaStreamHandlers.runIdPropagation.test.ts](#)

Copy

Review

Accept



patching file: [registerDeepLinkUriHandler.scope.test.ts](#)

[AICode/src/extension/src/extension/uriHandlers/ test /registerDeepLinkUriHandler.scope.test.ts](#)

Copy

Review

Accept

PatchFull file

```

-   registeredUriHandler = undefined;
-   registerUriHandlerMock.mockClear();
-   openCentralPanelMock.mockClear();
-   showErrorMessageMock.mockClear();
-
+   registeredUriHandler = undefined;
+
+   // Restore asynchronous collaborators after Jest resetMocks so fire-and-forget
+   // opening always receives a real Promise with the production `.catch(...)`
+   // contract.
+   registerUriHandlerMock.mockImplementation((handler) => {
+     registeredUriHandler = handler;
+     return { dispose: (): void => {} };
+   });
+   openCentralPanelMock.mockResolvedValue(undefined);
+   showErrorMessageMock.mockResolvedValue(undefined);
+

```

[AICode/src/extension/src/extension/uriHandlers/ test /registerDeepLinkUriHandler.scope.test.ts](#)

Copy

Review

Accept



patching file: [iaStreamHandlers.queryDoneSound.test.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/ test /iaStreamHandlers.queryDoneSound.test.ts](#)

PatchFull file

[AICode/src/extension/src/messageBus/handlers/ia/ test /iaStreamHandlers.queryDoneSound.test.ts](#)

[AICode/src/extension/src/messageBus/handlers/ia/](#) test [/SidebarQueryRunningBridge.race.test.ts](#)

PatchFull file

```

- ../../../../extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.js',
-     () => ({
-
+
+ ../../../../extension/panels/sidebar/SidebarStore.queryWindow.listeners.manager.js',
+     () => ({
+         // Rehydration and mutation modules share this mocked ESM namespace, so every
+         // statically
+         // imported listener helper must remain available while the runtime bridge is
+         // isolated.
+         isSameQueryActivityLocator: (
+             left: ChatSessionLocator | undefined,
+             right: ChatSessionLocator,
+         ) =>
+             left?.sessionId === right.sessionId &&
+             left.storageScopeRelPath === right.storageScopeRelPath,
+         panelOpened: async () => {},
+         setQueryWindowTitle: async () => {},
+
-         jest.clearAllMocks();
-
+         jest.clearAllMocks();
+

```

```
+ // Restore exact async and synchronous collaborators after Jest resetMocks.
+ sourceStartedMock.mockResolvedValue(undefined);
+ sourceSettledMock.mockResolvedValue(undefined);
+ sourceStopRequestedMock.mockReturnValue(true);
+ panelClosedMock.mockResolvedValue(undefined);
+ setElapsedMock.mockResolvedValue(undefined);
+
```

[AICode/src/extension/src/messageBus/handlers/ia/\\_test\\_/SidebarQueryRunningBridge.race.test.ts](#)

Copy

Review

Accept

  reading project:  [delete.endpoint.liveSession.test.ts](#)

  patching file:  [delete.endpoint.liveSession.test.ts](#)

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\\_test\\_/delete.endpoint.liveSession.test.ts](#)

Copy

Review

Accept

PatchFull file

```
- LiveSessionService['acquireLocatorDestructionClaim']
- >(() => claim);
- const evictMock = jest.fn<LiveSessionService['evictLocator']>(() => {
-   order.push('evict');
- });
- const releaseMock = jest.fn<
-   LiveSessionService['releaseLocatorDestructionClaim']
- >(() => {
-   order.push('release');
-   return true;
- });
- const deleteMock = jest.fn<(locator: ChatSessionLocator) => Promise<void>>(
-   async () => {
-     order.push('delete');
-   },
- );
-
+ LiveSessionService['acquireLocatorDestructionClaim']
+ >();
+ const evictMock = jest.fn<LiveSessionService['evictLocator']>();
+ const releaseMock = jest.fn<
+   LiveSessionService['releaseLocatorDestructionClaim']
+ >();
+ const deleteMock = jest.fn<(locator: ChatSessionLocator) => Promise<void>>();
+ const showErrorMessageMock = jest.fn<
+   (message: string) => Promise<undefined>
+ >();
+
-
-   updateSidebarStore: async () => {},
- },
- );
-
+   updateSidebarStore: async () => {},
+ },
+ );
+ jest.unstable_mockModule('../../../../../extension/core/VsCode.js', () => ({
+   VsCode: {
+     window: {
+       showErrorMessage: showErrorMessageMock,
+     },
+   },
+ }));
```

```

+ }));
+
-     jest.clearAllMocks();
-
+     jest.clearAllMocks();
+
+     // Recreate every stateful service mock after Jest resetMocks so the
destructive claim
+     // remains held through durable deletion, exact eviction, and finally release.
+
-     acquireMock.mockReturnValue(claim);
-
+     acquireMock.mockReturnValue(claim);
+     evictMock.mockImplementation(() => {
+         order.push('evict');
+     });
+     releaseMock.mockImplementation(() => {
+         order.push('release');
+         return true;
+     });
+
-     order.push('delete');
-     });
-
+     order.push('delete');
+     });
+     showErrorErrorMessageMock.mockResolvedValue(undefined);
+

```

[AICode/src/extension/src/slimbk-local/endpoints/ai/orch/chat/\\_test\\_/delete.endpoint.liveSession.test.ts](#)

Copy

Review

Accept



reading project: [OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts](#) [L25-

L90], [abortSource.endpoint.test.ts](#) [L1-L95], [abort.endpoint.test.ts](#) [L1-L75], [stream.endpoint.test.ts](#) [L1-

L65], [delete.endpoint.liveSession.test.ts](#) [L1-L135], [SidebarStore.handlers.manager.test.ts](#) [L1-

L90], [openPanel.util.test.ts](#) [L110-L190], [PromptQueryDoneSound.manager.test.ts](#) [L1-

L85], [iaStreamHandlers.runIdPropagation.test.ts](#) [L1-L60], [registerDeepLinkUriHandler.scope.test.ts](#) [L65-L100],

[iaStreamHandlers.queryDoneSound.test.ts](#) [L1-L135], [SidebarQueryRunningBridge.race.test.ts](#) [L1-

L105], [SidebarStore.rehydrate.manager.ts](#) [L1-L35]

**Les 12 suites défailantes ont été corrigées**

J'ai appliqué **LLM-DEBUG-3** et relu les documents d'architecture `liveSession` et de persistance avant toute modification.

### Cause racine principale

La configuration `jest.config.ts` active simultanément :

- `clearMocks: true`
- `resetMocks: true`
- `restoreMocks: true`

`resetMocks: true` conserve l'objet Mock injecté dans les modules ESM, mais **efface son implémentation avant chaque test**. Plusieurs tests définissaient leurs comportements uniquement lors de la déclaration globale du mock.

Le résultat était cohérent sur l'ensemble des pannes :

### Symptôme

Claims d'abort refusés

Replay non itérable

`.catch` sur `undefined`

### Cause réelle

Les mocks de claim retournaient `undefined` après le reset

Le mock du générateur asynchrone retournait `undefined`

Les mocks asynchrones du son et de l'ouverture de panneau ne retournaient plus de promesse

| Symptôme                                                                                                                                                                                | Cause réelle                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Coordinateur Prompt inactif                                                                                                                                                             | Son mock ne délégait plus au low-level opener                            |
| Éviction/release absentes                                                                                                                                                               | Leurs implémentations d'enregistrement de l'ordre avaient été supprimées |
| Append absent de l'ordre FIFO                                                                                                                                                           | Le mock <code>appendChunk</code> avait perdu son implémentation          |
| Allocation Activity indéfinie                                                                                                                                                           | Le mock d'allocation avait perdu son résultat                            |
| RPC Stop retournant <code>undefined</code>                                                                                                                                              | Le mock du service détaché n'avait plus de résultat                      |
| Export                                                                                                                                                                                  | Le namespace mocké des listeners était incomplet                         |
| ESM <code>panelOpened</code> manquant                                                                                                                                                   |                                                                          |
| Les implémentations nécessaires sont désormais reconstruites dans les <code>beforeEach</code> , après le reset Jest, tout en conservant exactement les signatures réelles des services. |                                                                          |

#### Autres causes corrigées

##### Compteur du sink de persistance

Dans `OrchMessagesMemoryAssistantPatch.errorSuffix.test.ts`, le sink était installé **avant** la création du message utilisateur et de l'assistant.

Les quatre appels observés étaient donc légitimes :

1. persistance du message utilisateur ;
2. persistance de l'assistant initial ;
3. première application du suffixe ;
4. seconde application idempotente.

Le test voulait seulement mesurer les deux opérations de patch. Le sink est maintenant attaché après la création de la fixture historique. L'idempotence du texte reste vérifiée, sans modifier le comportement applicatif.

##### Namespace ESM incomplet

Dans `SidebarQueryRunningBridge.race.test.ts`, le module des transitions Sidebar était remplacé entièrement, mais le faux namespace n'exportait pas les fonctions importées par les dépendances transitives.

Le mock expose désormais les helpers statiquement requis, dont :

- `panelOpened`;
- `setQueryWindowTitle`;
- `isSameQueryActivityLocator`;
- les transitions source testées.

#### Fichiers corrigés

| Domaine             | Test corrigé                                | Changement                                                                                                          | État |
|---------------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------|------|
| Persistance Orch    | <a href="#">test des suffixes d'erreur</a>  | Le sink mesure uniquement les opérations de patch                                                                   | ✓    |
| Stop source         | <a href="#">endpoint abortSource</a>        | Restauration du claim, du latch Activity et de l'abort                                                              | ✓    |
| Stop viewer         | <a href="#">endpoint abort</a>              | Restauration du claim viewer exact                                                                                  | ✓    |
| Replay              | <a href="#">endpoint stream</a>             | Reconstruction du générateur asynchrone après reset                                                                 | ✓    |
| Delete atomique     | <a href="#">test du Delete sous claim</a>   | Restauration <code>acquire</code> , <code>delete</code> , <code>evict</code> , <code>release</code> et mock VS Code | ✓    |
| RPC Sidebar         | <a href="#">handlers SidebarStore</a>       | Restauration des retours Promise et <code>{ accepted }</code>                                                       | ✓    |
| Coordination Prompt | <a href="#">façade openPanel1</a>           | Le coordinateur redélègue au low-level opener                                                                       | ✓    |
| Son de fin          | <a href="#">manager du son Prompt</a>       | Le mock audio retourne de nouveau une promesse                                                                      | ✓    |
| Identité Activity   | <a href="#">propagation du runId</a>        | Résultat d'allocation exact restauré                                                                                | ✓    |
| Deep links          | <a href="#">handler des URI</a>             | Ouverture Prompt promise-safe après reset                                                                           | ✓    |
| FIFO source         | <a href="#">ordre persistance et append</a> | <code>appendChunk</code> restauré, ordre durable vérifiable                                                         | ✓    |
| Courses Activity    | <a href="#">courses du bridge Activity</a>  | Namespace ESM complet et mocks restaurés                                                                            | ✓    |

#### Garanties de qualité

- **Aucun fichier applicatif n'a été modifié dans ce lot.**

- Aucun test n'a été désactivé, supprimé ou rendu moins strict.
  - Aucun `any`, cast artificiel, `eslint-disable`, fallback ou délai arbitraire.
  - Les assertions importantes restent inchangées :
  - `claim exact` ;
  - ordre `stopRequested` → `abort` ;
  - ordre `close` → `removeActivity` → `delete` → `evict` → `release` ;
  - persistance avant `append` ;
  - déduplication du son par `run` exact ;
  - suppression du mapping avant `dispose` coordonné ;
  - conservation du `runId` jusqu'à la finalisation.
  - Les mocks utilisent toujours les chemins relatifs depuis leurs propres fichiers de test.
- Relance maintenant la suite complète de tests `extension` et colle le prochain résumé des échecs — ou la belle ligne verte si Jest décide enfin de coopérer. ●

### Nom du commit

text

Copy

test(live-session): restore reset-safe mocks across extension suites

🎵 Les mocks survivent au grand reset 🗑️ Les claims et les replays reprennent leur ballet 🎶

</ASSISTANT>

<USER>

Summary of all failing tests FAIL src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx ● Prompt auto-run prompt › passes one-shot EasyAgents auto-run options only to the direct prompt auto-run TypeError: Cannot read properties of undefined (reading 'accepted') 1189 | nextTurnId, 1190 | ); > 1191 | if (!result.accepted) { | ^ 1192 | setSubmitTxnPending(false); 1193 | return; 1194 | } at src/app/prompt/Prompt/Prompt.tsx:1191:41 at gateAction (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:249:18) at onSend (src/app/prompt/Prompt/Prompt.tsx:1163:21) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:148:18 at executeDispatch (node\_modules/react-dom/cjs/react-dom-client.development.js:19116:9) at runWithFiberInDEV (node\_modules/react-dom/cjs/react-dom-client.development.js:874:13) at processDispatchQueue (node\_modules/react-dom/cjs/react-dom-client.development.js:19166:19) at node\_modules/react-dom/cjs/react-dom-client.development.js:19767:9 at batchedUpdates\$1 (node\_modules/react-dom/cjs/react-dom-client.development.js:3255:40) at dispatchEventForPluginEventSystem (node\_modules/react-dom/cjs/react-dom-client.development.js:19320:7) at dispatchEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23585:11) at dispatchDiscreteEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23553:11) at HTMLDivElement.callTheUserObjectsOperation (node\_modules/jsdom/lib/jsdom/living/generated/EventListener.js:26:30) at innerInvokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:350:25) at invokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:286:3) at HTMLButtonElementImpl.\_dispatch (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:233:9) at fireAnEvent (node\_modules/jsdom/lib/jsdom/living/helpers/events.js:18:36) at HTMLButtonElementImpl.click (node\_modules/jsdom/lib/jsdom/living/nodes/HTMLElement-impl.js:85:5) at HTMLButtonElement.click (node\_modules/jsdom/lib/jsdom/living/generated/HTMLElement.js:108:34) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:578:47 at node\_modules/@testing-library/react/dist/act-compat.js:47:24 at process.env.NODE\_ENV.exports.act (node\_modules/react/cjs/react.development.js:814:22) at node\_modules/@testing-library/react/dist/act-compat.js:46:25 at Object.<anonymous> (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:577:15) ● Prompt auto-run prompt › passes one-shot EasyAgents auto-run options only to the direct prompt auto-run TypeError: Cannot read properties of undefined (reading 'accepted') 1189 | nextTurnId, 1190 | ); > 1191 | if (!result.accepted) { | ^ 1192 | setSubmitTxnPending(false); 1193 | return; 1194 | } at src/app/prompt/Prompt/Prompt.tsx:1191:41 at gateAction (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:249:18) at onSend (src/app/prompt/Prompt/Prompt.tsx:1163:21) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:148:18 at executeDispatch (node\_modules/react-dom/cjs/react-dom-client.development.js:19116:9) at runWithFiberInDEV (node\_modules/react-dom/cjs/react-dom-client.development.js:874:13) at processDispatchQueue (node\_modules/react-dom/cjs/react-dom-client.development.js:19166:19) at node\_modules/react-dom/cjs/react-dom-

client.development.js:19767:9 at batchedUpdates\$1 (node\_modules/react-dom/cjs/react-dom-client.development.js:3255:40) at dispatchEventForPluginEventSystem (node\_modules/react-dom/cjs/react-dom-client.development.js:19320:7) at dispatchEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23585:11) at dispatchDiscreteEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23553:11) at HTMLDivElement.callTheUserObjectsOperation (node\_modules/jsdom/lib/jsdom/living/generated/EventListener.js:26:30) at innerInvokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:350:25) at invokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:286:3) at HTMLButtonElementImpl.\_dispatch (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:233:9) at fireAnEvent (node\_modules/jsdom/lib/jsdom/living/helpers/events.js:18:36) at HTMLButtonElementImpl.click (node\_modules/jsdom/lib/jsdom/living/nodes/HTMLElement-impl.js:85:5) at HTMLButtonElement.click (node\_modules/jsdom/lib/jsdom/living/generated/HTMLElement.js:108:34) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:578:47 at node\_modules/@testing-library/react/dist/act-compat.js:47:24 at process.env.NODE\_ENV.exports.act (node\_modules/react/cjs/react.development.js:814:22) at node\_modules/@testing-library/react/dist/act-compat.js:46:25 at Object.<anonymous> (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:577:15) • Prompt auto-run prompt › passes one-shot EasyAgents auto-run options only to the direct prompt auto-run TypeError: Cannot read properties of undefined (reading 'accepted') 1189 | nextTurnId, 1190 | ); > 1191 | if (!result.accepted) { | ^ 1192 | setSubmitTxnPending(false); 1193 | return; 1194 | } at src/app/prompt/Prompt/Prompt.tsx:1191:41 at gateAction (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:249:18) at onSend (src/app/prompt/Prompt/Prompt.tsx:1163:21) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:148:18 at executeDispatch (node\_modules/react-dom/cjs/react-dom-client.development.js:19116:9) at runWithFiberInDEV (node\_modules/react-dom/cjs/react-dom-client.development.js:874:13) at processDispatchQueue (node\_modules/react-dom/cjs/react-dom-client.development.js:19166:19) at node\_modules/react-dom/cjs/react-dom-client.development.js:19767:9 at batchedUpdates\$1 (node\_modules/react-dom/cjs/react-dom-client.development.js:3255:40) at dispatchEventForPluginEventSystem (node\_modules/react-dom/cjs/react-dom-client.development.js:19320:7) at dispatchEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23585:11) at dispatchDiscreteEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23553:11) at HTMLDivElement.callTheUserObjectsOperation (node\_modules/jsdom/lib/jsdom/living/generated/EventListener.js:26:30) at innerInvokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:350:25) at invokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:286:3) at HTMLButtonElementImpl.\_dispatch (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:233:9) at fireAnEvent (node\_modules/jsdom/lib/jsdom/living/helpers/events.js:18:36) at HTMLButtonElementImpl.click (node\_modules/jsdom/lib/jsdom/living/nodes/HTMLElement-impl.js:85:5) at HTMLButtonElement.click (node\_modules/jsdom/lib/jsdom/living/generated/HTMLElement.js:108:34) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:578:47 at node\_modules/@testing-library/react/dist/act-compat.js:47:24 at process.env.NODE\_ENV.exports.act (node\_modules/react/cjs/react.development.js:814:22) at node\_modules/@testing-library/react/dist/act-compat.js:46:25 at Object.<anonymous> (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:577:15) • Prompt auto-run prompt › passes one-shot EasyAgents auto-run options only to the direct prompt auto-run TypeError: Cannot read properties of undefined (reading 'accepted') 1189 | nextTurnId, 1190 | ); > 1191 | if (!result.accepted) { | ^ 1192 | setSubmitTxnPending(false); 1193 | return; 1194 | } at src/app/prompt/Prompt/Prompt.tsx:1191:41 at gateAction (src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:249:18) at onSend (src/app/prompt/Prompt/Prompt.tsx:1163:21) at src/app/prompt/Prompt/\_\_test\_\_/Prompt.autoRunPrompt.test.tsx:148:18 at executeDispatch (node\_modules/react-dom/cjs/react-dom-client.development.js:19116:9) at runWithFiberInDEV (node\_modules/react-dom/cjs/react-dom-client.development.js:874:13) at processDispatchQueue (node\_modules/react-dom/cjs/react-dom-client.development.js:19166:19) at node\_modules/react-dom/cjs/react-dom-client.development.js:19767:9 at batchedUpdates\$1 (node\_modules/react-dom/cjs/react-dom-client.development.js:3255:40) at dispatchEventForPluginEventSystem (node\_modules/react-dom/cjs/react-dom-client.development.js:19320:7) at dispatchEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23585:11) at dispatchDiscreteEvent (node\_modules/react-dom/cjs/react-dom-client.development.js:23553:11) at HTMLDivElement.callTheUserObjectsOperation (node\_modules/jsdom/lib/jsdom/living/generated/EventListener.js:26:30) at innerInvokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:350:25) at invokeEventListeners (node\_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:286:3) at HTMLButtonElementImpl.\_dispatch

```
(node_modules/jsdom/lib/jsdom/living/events/EventTarget-impl.js:233:9) at fireAnEvent
(node_modules/jsdom/lib/jsdom/living/helpers/events.js:18:36) at HTMLButtonElementImpl.click
(node_modules/jsdom/lib/jsdom/living/nodes/HTMLElement-impl.js:85:5) at HTMLButtonElement.click
(node_modules/jsdom/lib/jsdom/living/generated/HTMLElement.js:108:34) at
src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx:578:47 at node_modules/@testing-library/react/dist/act-
compat.js:47:24 at process.env.NODE_ENV.exports.act (node_modules/react/cjs/react.development.js:814:22) at
node_modules/@testing-library/react/dist/act-compat.js:46:25 at Object.<anonymous>
(src/app/prompt/Prompt/__test__/Prompt.autoRunPrompt.test.tsx:577:15) Test Suites: 1 failed, 376 passed, 377 total Tests:
1 failed, 1046 passed, 1047 total Snapshots: 0 total Time: 21.337 s Ran all test suites. Command failed: C:\Program
Files\nodejs\node.exe --expose-gc --max-old-space-size=8192 --experimental-vm-modules C:
\dev\AICode\src\frontend\node_modules\jest\bin\jest.js --config C:\dev\AICode\src\frontend\jest.config.ts --maxWorkers
10 (exit 1)
</USER>
<ASSISTANT>
```

  patching file:  [Prompt.autoRunPrompt.test.tsx](#)

Copy

Accept

```

- busCallMock.mockImplementation(async () => undefined);
- handleSendMock.mockClear();
- handleSendSpecifActionMock.mockClear();
-
+ busCallMock.mockImplementation(async () => undefined);
+
+ // The frontend Jest profile resets mock implementations before every test.
Restore the
+ // exact synchronous acceptance contract used by both auto-run and manual Send
paths.
+ handleSendMock.mockImplementation(() => ({
+   accepted: true,
+   runRef: {
+     locator: sessionLocatorMock,
+     turnId: 'turn-auto',
+     sourceId: 'source-auto',
+   },
+   streamInstanceId: 'stream-auto',
+ }));
+ handleSendSpecifActionMock.mockImplementation(() => ({
+   accepted: true,
+   runRef: {
+     locator: sessionLocatorMock,
+     turnId: 'turn-specif',
+     sourceId: 'source-specif',
+   },
+   streamInstanceId: 'stream-specif',
+ }));
+

```

Copy

Accept

**Dernier test frontend corrigé** 

J’ai appliqué **LLM-DEBUG-3**. La cause est la même politique `resetMocks: true` identifiée dans les tests Extension, cette fois dans [la configuration Jest frontend](#).

Les mocks `handleSendMock` et `handleSendSpecifActionMock` étaient correctement typés et possédaient une implémentation lors de leur déclaration. Cependant, Jest effaçait cette implémentation avant chaque test. Le composant recevait alors :

- `handleSend(...)` `=== undefined` ;
  - puis tentait légitimement de lire `result.accepted` ;
  - d’où l’exception répétée pendant le clic manuel.
- La correction est exclusivement dans [Prompt.autoRunPrompt.test.tsx](#) :
- le `beforeEach` restaure le résultat d’acceptation synchrone de `handleSend` ;
  - il restaure également celui de `handleSendSpecifAction` ;
  - chaque résultat contient une identité complète :
  - `locator` ;
  - `turnId` ;
  - `sourceId` ;
  - `streamInstanceId` ;
  - l’auto-run et l’envoi manuel passent donc par le même contrat exact que le code réel.

**Résumé**

| Élément                  | Correction                                                          | État                                                                                  |
|--------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Mock Send                | Résultat <code>accepted: true</code> réinstallé après le reset Jest |    |
| Mock SPECIF              | Même contrat synchrone réinstallé                                   |    |
| Identité source          | <code>locator + turnId + sourceId</code> conservés                  |    |
| Identité frontend        | <code>streamInstanceId</code> conservé                              |    |
| Test one-shot EasyAgents | Vérifie toujours que les options ne contaminent pas le Send manuel  |  |
| Code applicatif          | Aucun changement                                                    |  |
| Contournements           | Aucun cast, <code>any</code> , délai ou test assoupli               |  |

Le bilan précédent indiquait déjà **376 suites vertes sur 377** et **1046 tests verts sur 1047**. Relance la suite frontend : ce dernier test devrait maintenant rejoindre les autres. 

**Nom du commit**

text

Copy

test(prompt): restore synchronous send mocks after Jest reset

 Le dernier mock retrouve son résultat  Mille quarante-sept tests avancent d’un même pas 

</ASSISTANT>