

Permalink URI:

vscode://joelabenhaim.aicode-ja/chat?sessionId=e3e1ba00cfa44201b46e5e854b6c8fc1-VS-V%C3%A9rifier-finalisa

<USER>

[Verify]SPECIF_FINALISER_LIVESESSION_POUR_LES_SESSIONS_PROMPT_NORMALES_20260712_204731

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧 ☒ scanning recent files

🔧 ☒ scanning git: 🗑️ "État Git courant", 🗑️ "Étendue des changements"

🔧 ☒ reading project: 📁 g "Docs liveSession proches", 📁 g "Docs invariants ciblés", 📁 g "Docs flux ciblés"

🔧 ☒ scanning git: 🗑️ "Historique récent", 🗑️ "Contenu du dernier commit"

🔧 ☒ reading project: 📄 [liveSession.programming-doc-readme.md](#), 📄 [easyAgents.programming-doc-readme.md](#), 📄 [isomorphism-invariant-rule-between-streaming-and-rehydration-of-chat-sessions.programming-doc-readme.md](#), 📄 [thinking-bubble-and-hourglass-streaming-rules-invariants.programming-doc-readme.md](#), 📄 [prompt-stream-dispatch-lock.programming-doc-readme.md](#), 📄 [fallback-persist-and-title.programming-doc-readme.md](#), 📄 h "Contrats liveSession", 📄 h "Registre et claims", 📄 h "Source Prompt normale", 📄 h "Origine et transport", 📄 h "Coordination panneaux", 📄 h "Frontend transactionnel", 📄 h "Activité Sidebar"

🔧 ☒ scanning git: 🗑️ "Interdits dans le commit"

🔧 ☒ reading project: 📄 "Preuves architecture clés", 📄 [iaStreamHandlers.ts](#) [L200-L320], 📄 [useAiStream.hook.ts](#) [L400-L620], 📄 [useAiStream.hook.ts](#) [L620-L920], 📄 [usePromptLiveSession.hook.ts](#) [L1-L380], 📄 [PromptLiveSessionPanelCoordinator.service.ts](#) [L100-L240], 📄 [parseSlimbkStreamBodyError.util.ts](#) [L1-L220], 📄 [LiveSessionRegistry.service.ts](#) [L1-L220], 📄 [LiveSessionOpen.service.ts](#) [L1-L260], 📄 [useAiStream.hook.ts](#) [L850-L1035], 📄 [PromptLiveSessionPanelQuery.util.ts](#) [L1-L220], 📄 [PromptLiveSessionBootstrap.adapter.ts](#) [L1-L240], 📄 [Prompt.tsx](#) [L1-L230], 📄 [applyHydratedPromptSessionState.util.ts](#) [L1-L260], 📄 [SidebarStore.rehydrate.manager.ts](#) [L1-L300], 📄 [delete.endpoint.ts](#) [L1-L260], 📄 [EasyAgentsBatchRunRetry.service.ts](#) [L1-L300], 📄 [openPanel.util.ts](#) [L540-L700], 📄 [EasyAgentsBatchRunRetry.service.ts](#) [L300-L587]

🔧 ☒ scanning git: 🗑️ "Diff Queue A", 🗑️ "Diff Queue B", 🗑️ "Diff lock de stream", 🗑️ "Diff sanitizers Prompt", 🗑️ "Diff sanitizers courts", 🗑️ "Qualité du patch", 🗑️ "Résumé quantitatif", 🗑️ "Fichiers générés du patch", 🗑️ "Régression historique"

🔧 ☒ reading project: 📄 [usePromptLiveSession.hook.ts](#) [L330-L459]

🔧 ☒ scanning git: 🗑️ "Tests désactivés", 🗑️ "Tests focalisés accidenta"

🔧 ☒ reading project: 📄 e "Finalisation transport", 📄 e "Mutation generated", 📄 [SidebarQueryRunningBridge.ts](#) [L1-L300], 📄 [SidebarStore.queryWindow.listeners.manager.ts](#) [L1-L260], 📄 [SidebarStore.queryWindow.mutations.manager.ts](#) [L1-L180], 📄 [SidebarActivityQuery.tsx](#) [L1-L125], 📄 [SidebarStore.ts](#) [L120-L260], 📄 [SidebarStore.queryWindow.listeners.manager.ts](#) [L240-L340], 📄 [Prompt.tsx](#) [L640-L850]

Verdict global 📊

J'ai réinspecté le commit 5340864b, les principaux chemins d'exécution, les contrats partagés, les tests associés et les six programming docs concernées. Le dépôt est propre, et le commit représente une implémentation très importante : **176 fichiers modifiés, 18 146 insertions et 11 291 suppressions.**

Verdict : l'implémentation fonctionnelle est pratiquement complète, mais l'adhérence totale à la spécification n'est pas parfaite.

1) L'intégralité de la spécification est-elle implémentée ?

Réponse

⚠️ Environ 98 % de la spécification est implémentée.

La totalité des grands sous-systèmes demandés est présente :

- identité exacte `locator` + `turnId` + `sourceId` ;
- réservation synchrone avant `await` ;
- époques, retraites, tombstones et génération destructive ;
- replay `replaceTurnAssistants` et `continueAssistant` ;

- persistance des `meta.error` avant exposition ;
- détection des erreurs Slimbk post-header ;
- contrôle de la webview source exacte ;
- démarrage frontend transactionnel ;
- protection par `streamInstanceId` ;
- réouverture des Prompt normaux ;
- activité persistante panneau fermé ;
- Stop détaché ;
- handoff direct ;
- mode `preserveLocal` ;
- continuation et thinking ;
- Delete et Retry sous claim destructif ;
- conservation du comportement EasyAgents.

Les fichiers créés prévus par la spécification existent, notamment :

- [LiveSessionReplay.type.ts](#)
- [NormalPromptLiveSessionSource.service.ts](#)
- [PromptLiveSessionPanelCoordinator.service.ts](#)
- [NormalPromptLiveSession.adapter.ts](#)
- [PromptLiveSessionBootstrap.adapter.ts](#)
- [parseSlimbkStreamBodyError.util.ts](#)

Écarts vérifiés

⚠ **Programming doc EasyAgents partiellement obsolète**

La programming doc [easyAgents.programming-doc-readme.md](#) affirme encore que des paramètres `liveSession` invalides ne doivent pas empêcher la création du panneau lorsque le locator de base est valide. Cela contredit désormais :

- la spécification ;
- le parseur extension ;
- la frontière React frontend ;
- le comportement réel de [PromptLiveSessionPanelQuery.util.ts](#) ;
- le comportement réel de [PromptLiveSessionBootstrap.adapter.ts](#).

Une query live explicite invalide est bien rejetée avant l'opener, ce qui est le comportement voulu. **Le code est correct, la documentation ne l'est plus.**

✗ **Deux fichiers personnels hors périmètre ont été modifiés**

Les métadonnées Git du commit indiquent aussi des modifications dans :

- [sol-exploit.txt](#)
- [todo.txt](#)

Cela viole directement l'invariant selon lequel la documentation personnelle interdite ne devait pas être modifiée dans cette tranche.

Je n'ai pas chargé leur contenu. Je ne proposerai pas non plus de les éditer, conformément à l'interdiction explicite portant sur cette arborescence.

Conclusion de la question 1

⚠ **Le produit fonctionnel demandé est implémenté, mais la spécification n'est pas respectée à 100 % en raison d'une programming doc obsolète et de modifications Git hors périmètre.**

2) L'implémentation actuelle fait-elle ce qu'elle doit faire ?

Réponse détaillée par fonctionnalité

Fonctionnalité technique demandée	État	Preuves et conclusion
Identité exacte <code>locator</code> + <code>turnId</code> + <code>sourceId</code>	✓ 100 %	Le run key inclut les trois dimensions dans LiveSessionRun.type.ts . Les opérations exactes ne retombent pas sur le run actif du locator.
Contrat strict des queries live	✓ 100 %	Union EasyAgents/normalPrompt dans LiveSessionPanel.type.ts ; détection des doublons, paramètres inconnus et descripteurs partiels côté extension et frontend.

Fonctionnalité technique demandée	État	Preuves et conclusion
Query live invalide rejetée avant runtime	✓ 100 %	L'extension rejette avant le low-level opener et Prompt.tsx utilise une frontière React externe qui ne monte ni contrôleur, ni hydratation, ni auto-run.
Vérification de la webview d'origine exacte	✓ 100 %	Le contexte extension-only est transmis par le bus ; iaStreamHandlers.ts compare la webview émettrice au panneau actuellement enregistré avant la réservation.
Réservation synchrone avant tout <code>await</code>	✓ 100 %	La réservation revendique immédiatement le pointeur actif, incrémente l'époque et refuse doublons, tombstones, claims et sources concurrentes dans LiveSessionRegistry.service.ts .
Promotion réservation → record	✓ 100 %	La promotion exige le handle exact, conserve l'époque et applique l'abort précoce après priming du stream.
Buffer complet, ordonné et sans limite ajoutée	✓ 100 %	Le registre conserve tous les <code>AiStreamedChunk</code> dans l'ordre ; aucune compression ou limite supplémentaire n'a été ajoutée.
Baselines et replay modes	✓ 100 %	<code>replaceTurnAssistants</code> retire uniquement les assistants du tour exact ; <code>continueAssistant</code> réutilise la baseline immuable capturée immédiatement avant l'ouverture Orch. NormalPromptLiveSessionSource.service.ts appelle la
Persistance de tout <code>meta.error</code> avant <code>append</code>	✓ 100 %	persistance avant <code>append</code> et forwarding. Un terminal observé dont la persistance échoue n'est pas remplacé par un terminal synthétique.
Terminal synthétique <code>throw/EOF</code>	✓ 100 %	Créé uniquement lorsqu'aucun vrai terminal n'a été observé, puis persisté avant <code>append</code> .
Protocole V1 premier-header	✓ 100 %	Le premier <code>meta.error</code> peut être converti en erreur de header ; les erreurs ultérieures restent des chunks du body.
Erreur Slimbk post-header	✓ 100 %	parseSlimbkStreamBodyError.util.ts intercepte les enveloppes non-200 avant leur interprétation comme <code>AiStreamedChunk</code> .
FIFO extension et erreurs auxiliaires	✓ 100 %	Le handler utilise une FIFO unique et continue séparément Activity, son et émission frontend, tout en agrégeant les erreurs dans l'ordre.
Finalisation post-forwarding	✓ 100 %	La finalisation n'est appelée qu'après retour Slimbk et drainage FIFO. Les résultats sont bien les cinq états spécifiés.
Record sans baseline ou terminal	✓ 100 %	Retrait exact avec priorité <code>retiredMissingBaseline</code> , puis <code>retiredMissingTerminal</code> . Aucun record source non handoffable ne reste bloqué.
Époques, tombstones et retraites	✓ 100 %	Les trois notions sont distinctes et utilisées conformément au contrat anti-ABA et anti-résurrection.
Claims destructifs	✓ 100 %	Acquisition et release incrémentent la génération destructive ; l'éviction exige le handle exact et ne libère pas le claim.
Delete durable sous claim	✓ 100 %	delete.endpoint.ts acquiert le claim avant fermeture, n'évince qu'après succès disque et réconcilie Activity après échec.
Retry EasyAgents sous claim	✓ 100 %	EasyAgentsBatchRunRetry.service.ts conserve le claim pendant fermeture, éviction, suppression, reconstruction et redémarrage, puis le libère dans <code>finally</code> .
Coordinateur de panneaux	✓ 100 %	FIFO par locator, injection normalPrompt, releases par adaptateur, cleanups différés et wake-ups sont centralisés dans PromptLiveSessionPanelCoordinator.service.ts .
Continuation après fermeture du panneau	✓ 100 %	Le dispose détache le panneau sans abort de la source active ; Activity passe à <code>Running/source/closed</code> .

Fonctionnalité technique demandée	État	Preuves et conclusion
Projection Activity	✓ 100 %	Union stricte Idle/open, Running/observer/open et Running/source/open ou closed dans SidebarStore.ts .
Stop détaché	✓ 100 %	Affiché uniquement pour source/closed, icône <code>SquareStop</code> , taille <code>small</code> , variant <code>danger</code> , label <code>Stop</code> et désactivation immédiate dans SidebarActivityQuery.tsx .
Propagation synchrone de <code>stopRequested</code>	✓ 100 %	Le runtime et le store sont mutés avant l'appel d'abort, ce qui empêche la réactivation du Stop après fermeture immédiate.
Son exact et absence de doublon	✓ 100 %	Corrélation par <code>runRef</code> et tentative sonore ; les reconnections n'arment pas un second terminal sonore pour le même run. useAiStream.hook.ts refuse synchroniquement un second start
Start frontend transactionnel	✓ 100 %	et ne mute aucun état dans cette branche. Les identifiants ne sont générés qu'après acceptation.
Guards Queue A et Queue B	✓ 100 %	Les items transportent désormais <code>sourceId</code> + <code>streamInstanceId</code> ; la coalescence ne traverse pas une tentative et les continuations après reset sont protégées par génération.
Algorithmes historiques Queue/locks/sanitizers	✓ 98 %	Les algorithmes fonctionnels sont conservés. Queue A et Queue B ont nécessairement reçu des gardes d'identité et de génération, sans changement du contrat de dispatch métier.
Terminal suivi d'une erreur transport	✓ 100 %	Le frontend attend le drainage avant de décider si un terminal synthétique est nécessaire ; aucun second terminal n'est ajouté lorsque le vrai terminal a déjà été consommé.
Handoff direct normalPrompt	✓ 100 %	Exige terminal data, outcome <code>completed</code> , <code>streamInstanceId</code> exact et <code>finalizedStreamInstanceId</code> correspondant. Aucun second Open n'est exécuté.
Retry <code>preserveLocal</code>	✓ 100 %	L'hydratation sélective ne lit pas les champs de brouillon, attachments ou <code>nextTurnId</code> ; les tests emploient des getters piégés.
Composer interactif pendant replay	✓ 100 %	L'adaptateur normalPrompt autorise le brouillon et les attachments après armement, tout en gardant les chemins Send verrouillés.
Stop unique dans Prompt	✓ 100 %	Le Stop normal du compositeur est réutilisé ; la bannière normalPrompt n'en ajoute pas un second.
Barrière <code>fallback_persisted</code>	✓ 100 %	Corrélation par <code>{turnId, streamInstanceId}</code> ; clear sur ack exact, simple démasquage si la tentative finalise sans ack.
Continuation et thinking	✓ 100 %	Nouvelle préparation pure, tentative frontend distincte et bases thinking gardées par <code>streamInstanceId</code> .
Préservation EasyAgents	✓ 100 %	<code>sourceId === turnId</code> , mode replace, stop last-viewer, lifecycle partagé et UX read-only sont préservés.
Tests demandés	✓ 96 %	Les tests couvrent les scénarios critiques annoncés, sans tests focalisés ou désactivés détectés dans les zones auditées. La très grande surface asynchrone laisse néanmoins un risque résiduel d'intégration.
Programming docs	⚠ 94 %	Les documents principaux ont été enrichis, mais la programming doc EasyAgents contient au moins un ancien invariant contradictoire.
Respect du périmètre des fichiers	✗ 90 %	Deux fichiers personnels interdits ont été inclus dans le commit. Cela n'altère pas le runtime, mais viole explicitement le périmètre.
Conclusion de la question 2		
✓		Le comportement fonctionnel central demandé est présent et cohérent avec la spécification ; les écarts vérifiés concernent la documentation et l'hygiène du commit, pas le mécanisme runtime principal.

3) Y a-t-il des bugs ?

Réponse

Bug de documentation confirmé

La règle de gestion des queries live invalides documentée dans [easyAgents.programming-doc-readme.md](#) est obsolète.

Impact :

- pas de panne runtime immédiate ;
- risque réel qu'une future maintenance réintroduise une dégradation classique pour une query live invalide ;
- contradiction entre documentation, tests, frontend et extension.

Sévérité : moyenne pour la maintenance, faible pour le runtime actuel.

Défaut de périmètre confirmé

Deux fichiers personnels interdits ont été modifiés dans le commit fonctionnel.

Impact :

- aucun défaut runtime démontré ;
- commit pollué par des changements non liés ;
- violation explicite d'un invariant de livraison et du périmètre autorisé.

Sévérité : élevée pour la discipline de dépôt, nulle pour le comportement liveSession.

Aucun bug fonctionnel certain identifié dans les chemins audités

Je n'ai trouvé aucune preuve statique d'un défaut fonctionnel dans :

- la réservation/promotion ;
- la finalisation ;
- le replay ;
- les identités exactes ;
- le filtrage Slimbk ;
- le Stop ;
- le handoff direct ;
- les claims destructifs ;
- les transitions Activity ;
- les barrières frontend.

Cela ne signifie pas que le risque est nul : un changement atomique de **176 fichiers** avec plusieurs pipelines asynchrones conserve nécessairement un risque résiduel, particulièrement sur les courses Electron, le timing React et les erreurs réseau combinées.

Conclusion de la question 3

 **Deux défauts certains existent — une documentation contradictoire et une pollution du commit — mais aucun bug runtime certain n'a été trouvé dans l'implémentation liveSession examinée.**

4) Quelle est mon opinion sur le code source et la spécification ?

Réponse

Sur la spécification

Excellente spécification d'architecture, mais exceptionnellement large.

Ses principaux points forts sont :

- invariants explicites ;
- distinction précise des identités ;
- modélisation approfondie des courses ;
- prise en compte des erreurs post-header ;
- séparation retraites/tombstones/claims ;
- détail des comportements de retry et handoff ;
- exigence d'isomorphisme ;
- inventaire exact des fichiers et tests.

Le facteur de risque annoncé de **92 %** était crédible compte tenu du blast radius. La spécification protège bien les zones où une implémentation naïve aurait produit des résurrections, doublons, pertes de brouillon ou cleanups concurrents.

Son principal défaut est son atomicité extrême : elle rend la revue humaine et l'identification d'une régression beaucoup plus difficiles.

Sur le code

Architecture globalement très solide et professionnelle.

Points particulièrement réussis :

- identité structurelle sans concaténation ambiguë ;
- réservation synchrone ;
- handles opaques ;
- états discriminés strictement ;
- séparation service source / handler de forwarding / coordinateur panneau ;
- guards frontend par tentative ;
- hydratation sélective réellement opaque ;
- commentaires techniques abondants ;
- tests spécifiques aux courses ;
- absence de fallback implicite vers un run actif.

Points de vigilance :

- le système possède désormais beaucoup d'identités simultanées : `turnId`, `sourceId`, `streamInstanceId`, `tentative` `Open`, `runId` `Activity`, époque et génération destructive ;
- la maintenance dépend fortement de la fidélité des programming docs ;
- quelques fichiers centraux restent très volumineux, notamment [useAiStream.hook.ts](#) et [Prompt.tsx](#) ;
- la moindre divergence entre stream, replay et hydratation peut provoquer un défaut difficile à reproduire.

Évaluation

Critère	Note
Architecture	9,3/10 ✓
Typage et contrats	9,5/10 ✓
Gestion des courses	9,2/10 ✓
Tests présents	9,0/10 ✓
Lisibilité locale	8,5/10 ✓
Maintenabilité globale	8,2/10 ⚠
Discipline de périmètre du commit	7,0/10 ✗
Adhérence fonctionnelle	9,8/10 ✓

Conclusion de la question 4

✓ Le code est remarquablement sérieux pour une fonctionnalité aussi risquée ; sa principale faiblesse n'est pas l'architecture, mais la taille de la tranche et la nécessité absolue de garder sa documentation exacte.

5) Résumé du rapport

Domaine	Complétude estimée	Verdict	Correction nécessaire
Contrats shared	100 %	✓	Non
Registre, réservations et finalisation	100 %	✓	Non
Source Prompt normale	100 %	✓	Non
Transport Slimbk et origine webview	100 %	✓	Non
Coordination panneaux et cleanup	100 %	✓	Non
Sidebar Activity et Stop détaché	100 %	✓	Non
Son et timers	100 %	✓	Non
Frontend transactionnel	100 %	✓	Non
Replay, handoff et preserveLocal	100 %	✓	Non
Continuation et thinking	100 %	✓	Non
Delete et Retry destructifs	100 %	✓	Non
Préservation EasyAgents	100 %	✓	Non
Couverture de tests statiquement visible	96 %	✓	Surveillance recommandée
Programming docs	94 %	⚠	Oui
Périmètre du commit	90 %	✗	Oui, mais fichiers personnels interdits

Domaine	Complétude estimée	Verdict	Correction nécessaire
Ensemble fonctionnel	≈ 99 %	✓	hors de mon périmètre
Adhérence totale à la spécification	≈ 98 %	⚠	Une correction documentaire autorisée Documentation + hygiène historique du commit

6) Conclusion en une phrase
La tranche liveSession pour Prompt normal est fonctionnellement complète et remarquablement robuste, mais elle n'est pas parfaitement conforme à la spécification à cause d'une programming doc obsolète et de deux modifications personnelles hors périmètre dans le commit.

7) Ensemble de corrections proposé
Ensemble complet des corrections directement autorisées

- Corriger dans [easyAgents.programming-doc-readme.md](#) l'ancien invariant concernant les queries live invalides.
- Documenter le contrat actuel :
 - toute query live explicite valide est prioritaire ;
 - toute query live explicite invalide est rejetée avant le low-level opener ;
 - aucune dégradation vers une ouverture classique n'est autorisée.
- Vérifier les autres paragraphes voisins de la même programming doc pour éviter une seconde contradiction avec le nouveau coordinateur.
- Ne toucher à aucun fichier personnel interdit ; leur retrait éventuel de l'historique du commit reste à gérer séparément par le propriétaire du dépôt.

🔧 Coder directement toutes les corrections autorisées

</ASSISTANT>

<USER>

[Coder directement toutes les corrections autorisées]

</USER>

<ASSISTANT>

I'm thinking...

</ASSISTANT>

<ASSISTANT>

🔧 ✓ reading project: [easyAgents.programming-doc-readme.md](#), [PromptLiveSessionPanelQuery.util.ts](#) [L1-L170], [PromptLiveSessionBootstrap.adapter.ts](#) [L1-L137], [PromptLiveSessionPanelCoordinator.service.ts](#) [L1-L155], [PromptLiveSession.types.ts](#) [L1-L240], [EasyAgentsBatchLiveSession.adapter.ts](#) [L1-L260], [NormalPromptLiveSession.adapter.ts](#) [L1-L260], ["Ancien invariant invalide"](#), ["Modes génériques documentés"](#), [easyAgents.programming-doc-readme.md](#) [L990-L1160]

🔧 ✓ patching file: [easyAgents.programming-doc-readme.md](#)
[AICode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md](#)

Copy

Review

Accept

PatchFull file

Mode	Meaning

`liveSessionOpening`	Exact liveSession Open bootstrap is in progress.
`liveSessionStreaming`	Read-only replay/tail through the shared Prompt stream pipeline.
`liveSessionHandoffPending`	Exact terminal snapshot revalidation is in progress.
`liveSessionHandoffFailed`	Exact terminal snapshot is unavailable; Retry remains read-only and explicit.

| `normal` | The exact returned terminal snapshot was hydrated and Prompt is interactive. |

| Mode | Meaning

| ----- |

----- |
| `liveSessionOpening` | Exact liveSession Open bootstrap is in progress.

| `liveSessionStreaming` | Replay/tail runs through the shared Prompt stream pipeline; the bound adapter decides whether the composer remains read-only or locally editable. |

| `liveSessionHandoffPending` | The adapter-specific handoff prerequisites are being revalidated after the replayed stream settled.

| `liveSessionHandoffFailed` | The adapter-specific terminal handoff could not complete; Retry remains explicit. |

| `liveSessionReplayFailed` | Open, replay, or transport failed before a handoffable terminal state was reached; Retry remains explicit.

| `normal` | The adapter-specific handoff completed and the regular interactive Prompt runtime owns the session again.

- `AiStreamStartArgs.kind='liveSessionObserver'` never starts a new provider query, Queue A, Queue B, stream locks, sanitizers, and raw-live processing remain unchanged in their Prompt subsystem,
- `AiStreamStartArgs.kind='liveSessionObserver'` never starts a new provider query, Queue A, Queue B, stream locks, sanitizers, and raw-live processing retain their existing algorithms; source and viewer attempts only add exact `sourceId + streamInstanceId` guards around inputs, outputs, and delayed continuations,

- the frontend-injected `meta.error` may arm a handoff attempt but cannot turn a non-terminal persistent snapshot into a terminal one,
- terminal bootstrap, automatic handoff, and manual Retry hydrate the exact snapshot returned by liveSession Open and never call `/api/ai/orch/chat/open` a second time.
- the frontend-injected `meta.error` may arm a handoff attempt but cannot turn a non-terminal persistent snapshot into a terminal one,
- for EasyAgents, terminal bootstrap, automatic handoff, and manual Retry hydrate the exact snapshot returned by liveSession Open and never call `/api/ai/orch/chat/open` a second time,
- normal Prompt sources use the generic subsystem's direct-after-replay handoff and preserve-local retry policies; those policies do not alter the EasyAgents read-only adapter contract.

- disposing a Batch history panel detaches the history binding,
- disposing a valid stop-on-close liveSession Prompt releases the exact locator + turn + lease before normal Prompt cleanup,
- invalid liveSession parameters never weaken the strict base Prompt locator and never block panel creation when that base locator is valid,
- disposing a Batch history panel detaches the history binding,
- disposing a valid stop-on-close EasyAgents liveSession Prompt releases the exact locator + turn + source + lease through the EasyAgents adapter before normal Prompt cleanup,
- any explicit liveSession query is parsed as one complete shared descriptor before the low-level Prompt opener is called,
- a valid explicit liveSession descriptor always has priority over implicit normal-Prompt source injection,
- an invalid explicit liveSession descriptor is rejected before panel creation and never degrades to a classic Prompt opening, even when the base locator alone would be valid,
- the frontend repeats the strict raw-query validation before mounting the Prompt runtime as a defense-in-depth boundary,

[AICode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md](https://github.com/aiCode/src/extension/src/aiTools/easyAgents/easyAgents.programming-doc-readme.md)

Copy
Review
Accept

  reading project:  [easyAgents.programming-doc-readme.md](#) [L1008-L1162]
  scanning git:  ["Contrôle du correctif doc"](#),  ["Diff final du document"](#)

Correction appliquée

J'ai relu la programming doc ainsi que les parseurs extension/frontend et le coordinateur de panneaux concernés. La documentation est désormais alignée avec le comportement réellement implémenté.

Correction	Résultat	Complétude
Ajout du mode <code>liveSessionReplayFailed</code> dans la table des états Prompt	La documentation couvre maintenant tous les modes définis par le contrat frontend	 100 %
Généralisation des descriptions <code>liveSessionStreaming</code> , <code>handoff</code> et <code>normal</code>	Les différences entre l'adaptateur EasyAgents read-only et le Prompt normal interactif sont explicites	 100 %
Clarification de Queue A, Queue B, locks et sanitizers	Les algorithmes historiques restent inchangés ; seuls les flux et continuations sont gardés par <code>sourceId</code> + <code>streamInstanceId</code>	 100 %
Clarification du handoff	EasyAgents hydrate le snapshot terminal ; Prompt normal utilise le handoff direct et les retries <code>preserveLocal</code>	 100 %
Correction de l'ancien invariant sur les queries invalides	Une query live explicite invalide est désormais documentée comme rejetée avant la création du panneau, sans fallback classique	 100 %
Priorité des queries explicites valides	La priorité sur l'injection implicite <code>normalPrompt</code> est documentée	 100 %
Validation frontend de défense en profondeur	La frontière frontend avant montage du runtime Prompt est documentée	 100 %
Identité lors du dispose EasyAgents	La documentation mentionne maintenant <code>locator</code> + <code>turn</code> + <code>source</code> + <code>lease</code>	 100 %
Fichiers personnels interdits	Aucun n'a été chargé ou modifié pendant cette correction	 Respecté
Fichier corrigé	easyAgents.programming-doc-readme.md	 Terminé

Résultat final

La contradiction identifiée pendant la vérification est supprimée, et les paragraphes voisins ont également été corrigés afin de ne plus présenter le comportement historique EasyAgents comme le comportement générique de toutes les sessions Prompt.

Nom de commit proposé

text

Copy

`docs(easy-agents): align live-session invariants with implementation`

 La doc suit désormais le vrai flux  Les prochains refactors avanceront sans brouillard 

</ASSISTANT>