

AgentSession

September 4, 2026

AgentSession

Kind: Class**Source:** [atloria-monorepo/libs/agent-core/src/session/agent-session.ts](#)

Interactive agent session.

Usage:

```
const session = new AgentSession(runtime, agentConfig, { maxTurns: 30 });
session.on('tool_start', (e) => console.log(`Using ${e.toolName}...`));
session.on('agent_message', (e) => console.log(e.text));

const result = await session.send('Build a login page');
const result2 = await session.send('Add password validation');
```

Extends: [EventEmitter](#)

Methods

Method	Signature	Returns
<code>on</code>	<code>on(event: K, listener: (data: SessionEventMap[K]) => void)</code>	<code>this</code>
<code>off</code>	<code>off(event: K, listener: (data: SessionEventMap[K]) => void)</code>	<code>this</code>
<code>emit</code>	<code>emit(event: K, data: SessionEventMap[K])</code>	<code>boolean</code>
<code>send</code>	<code>send(message: string)</code>	<code>Promise<RunResult></code>
<code>inject</code>	<code>inject(message: string)</code>	<code>void</code>
<code>cancel</code>	<code>cancel()</code>	<code>void</code>
<code>getState</code>	<code>getState()</code>	<code>SessionState</code>

Method	Signature	Returns
getHistory	getHistory()	AgentHistoryItem[]
getUsage	getUsage()	TokenUsage
getSendCount	getSendCount()	number
getTurnCount	getTurnCount()	number
getSnapshot	getSnapshot()	{ conversationHistory: AgentHistoryItem[]; sdkHistory: unknown[] }
loadHistory	loadHistory(conversationHistory: AgentHistoryItem[], sdkHistory: unknown[])	void
end	end()	void
setVerbose	setVerbose(enabled: boolean)	void
isVerbose	isVerbose()	boolean
isPlanMode	isPlanMode()	boolean
getActiveModel	getActiveModel()	`string
approvePlan	approvePlan(feedback: string)	Promise<RunResult>
revisePlan	revisePlan(feedback: string)	Promise<RunResult>
fork	fork(branchName: string)	AgentSession<TContext>

Where it refuses work

- `AgentSession` stops the work with `Error` when `this.state === 'running'` — “Session is already processing a message. Use `inject()` for mid-task messages.”.
- `AgentSession` stops the work with `Error` when `result.blocked`.
- `AgentSession` stops the work with `Error` when `this.state !== 'running'` — “Cannot inject: session is not running. Use `send()` instead.”.
- `AgentSession` stops the work with `Error` when `this.state !== 'idle'` — “Cannot load history: session is not idle.”.
- `AgentSession` stops the work with `Error` when `this.history.length > 0 || this.conversationHistory.length > 0` — “Cannot load history: session already has history.”.
- `AgentSession` stops the work with `Error` when `!this.planMode` — “Cannot approve plan: session is not in plan mode.”.

When something fails

- `AgentSession` handles failure in 6 places: it lets it reach the caller in 3, and discards it silently in 3. A failure discarded silently leaves no trace for whoever debugs this later.