Use the Vercel AI SDK with Radium

Use the Vercel AI SDK with Radium

Use the AI SDK's OpenAI-compatible provider to call Radium from Node.js or server-side code in a Next.js application.

Before you start

You need:

- Node.js 22 or newer
- npm
- A Radium API key

Radium calls must run on the server. Never expose the key through browser code or a `NEXT_PUBLIC_...` environment variable.

1. Create a project and install the packages

```
mkdir radium-vercel-ai-sdk
cd radium-vercel-ai-sdk
npm init -y
npm pkg set type=module
npm install ai @ai-sdk/openai-compatible
```

2. Set your Radium credentials

macOS or Linux:

```
export RADIUM_API_KEY="your-radium-api-key"
export RADIUM_MODEL="hal-1.0"
```

Windows PowerShell:

```
$env:RADIUM_API_KEY = "your-radium-api-key"
$env:RADIUM_MODEL = "hal-1.0"
```

3. Create `main.mjs`

```
import { generateText } from "ai";
import { createOpenAICompatible } from "@ai-sdk/openai-compatible";

if (!process.env.RADIUM_API_KEY) {
  throw new Error("Set RADIUM_API_KEY before running this program.");
}

const radium = createOpenAICompatible({
  name: "radium",
  apiKey: process.env.RADIUM_API_KEY,
  baseURL: "https://api.radium.cloud/v1",
});

const { text } = await generateText({
  model: radium.chatModel(process.env.RADIUM_MODEL ?? "hal-1.0"),
  system: "Follow the user's instruction exactly.",
  prompt: "Reply with exactly: Radium connected.",
  maxOutputTokens: 512,
  temperature: 0,
});

if (!text?.trim()) {
  throw new Error("Radium returned no visible text.");
}
console.log(`RADIUM_RESPONSE: ${text.trim()}`);
```

The base URL must include `/v1`. The provider adds `/chat/completions` itself.

4. Run it

```
node main.mjs
```

Expected output:

```
RADIUM_RESPONSE: Radium connected.
```

Choose a model

```
export RADIUM_MODEL="clarke-1.0"  # or hal-1.0 or tycho-1.0
node main.mjs
```

Migrate an existing AI SDK application

Create the provider once:

```
import { createOpenAICompatible } from "@ai-sdk/openai-compatible";

const radium = createOpenAICompatible({
  name: "radium",
  apiKey: process.env.RADIUM_API_KEY,
  baseURL: "https://api.radium.cloud/v1",
});
```

Then replace the model supplied to `generateText`, `streamText`, or your server route:

```
const model = radium.chatModel("hal-1.0");
```

Your prompts and server-side AI SDK handlers can remain unchanged.

Troubleshooting

- Node version errors: use Node.js 22 or newer for the pinned packages.
- Authentication errors: verify `RADIUM_API_KEY` in the server process.
- `404`: use the exact base URL and one of the listed model IDs.
- Browser key exposure: move the call into a server route or server action.

- Provider compatibility warning: update both `ai` and `@ai-sdk/openai-compatible` together so they use compatible current releases.

Validation

Version note: these instructions were verified with Node.js 24.1.0, `ai@7.0.71`, and `@ai-sdk/openai-compatible@3.0.33` on August 20, 2026. You do not need those exact package versions. Text generation and multi-step tool calling passed cleanly with `hal-1.0`, `clarke-1.0`, and `tycho-1.0`.

Reference: Vercel AI SDK OpenAI-compatible providers.

Next

- API quickstart, for calling Radium directly
- Tool calling, for the `tool_use` and `tool_result` contract

Created 2026-08-23 21:38:00 UTC by Admin
Updated 2026-08-23 21:54:28 UTC by Admin