

API quickstart

Call the Radium API. Anthropic-compatible Messages endpoint, authentication, model names, production and staging base URLs, and OpenAI-compatible Chat Completions.

Quickstart

Radium serves an Anthropic-compatible Messages API. If your application already speaks that shape, or you use Anthropic-style tool calling, or you run Claude Code, this is the endpoint you want and the change is a base URL and a model name.

An OpenAI-compatible Chat Completions endpoint is available as well, and it is covered further down for applications that expect it.

Your first request

Set your key once:

```
export RADIUM_API_KEY="YOUR_RADIUM_API_KEY"
```

Then send this:

```
curl -s https://api.radium.cloud/v1/messages \
-H "content-type: application/json" \
-H "anthropic-version: 2023-06-01" \
-H "x-api-key: $RADIUM_API_KEY" \
-d '{
  "model": "hal-1.0",
  "max_tokens": 2048,
  "messages": [
    { "role": "user", "content": "Say only: radium hal connected." }
  ]
}'
```

That is the whole integration. Everything below is detail.

Authentication

Every request needs a key. The default header is `x-api-key`:

```
x-api-key: $RADIUM_API_KEY
```

Some deployments use a bearer token instead:

```
Authorization: Bearer $RADIUM_API_KEY
```

Use the mode Radium supplied with your key. If nobody specified one, start with `x-api-key` and you will almost certainly be right.

Models

Model	Built for
<code>hal-1.0</code>	Higher-capability coding and agentic work. Used in the examples on this page.
<code>clarke-1.0</code>	General-purpose and structured work
<code>tycho-1.0</code>	Faster, lighter-weight tasks

Swap the identifier in the request body, or in your Claude Code configuration, and nothing else changes.

The Messages API

Base URL:

```
https://api.radium.cloud
```

Full endpoint:

```
POST https://api.radium.cloud/v1/messages
```

The request body is the standard Anthropic Messages shape.

curl

```
curl -s https://api.radium.cloud/v1/messages \  
-H "content-type: application/json" \  

```

```
-H "anthropic-version: 2023-06-01" \  
-H "x-api-key: $RADIUM_API_KEY" \  
-d '{  
  "model": "hal-1.0",  
  "max_tokens": 2048,  
  "messages": [  
    { "role": "user", "content": "Hello, Radium!" }  
  ]  
'
```

Python

```
import anthropic  
  
client = anthropic.Anthropic(  
    api_key="YOUR_RADIUM_API_KEY",  
    base_url="https://api.radium.cloud",  
)  
  
message = client.messages.create(  
    model="hal-1.0",  
    max_tokens=1024,  
    messages=[  
        {"role": "user", "content": "Hello, Radium!"}  
    ],  
)  
  
print(message.content[0].text)
```

TypeScript

```
import Anthropic from "@anthropic-ai/sdk";  
  
const client = new Anthropic({  
    apiKey: "YOUR_RADIUM_API_KEY",  
    baseUrl: "https://api.radium.cloud",  
});  
  
const message: Anthropic.Message = await client.messages.create({  
    model: "hal-1.0",
```

```
max_tokens: 1024,  
messages: [{ role: "user", content: "Hello, Radium!" }],  
});  
  
console.log(message.content[0].text);
```

Tool calling works through the same endpoint, and it has its own page. See [tool calling](#).

Production and staging

Messages API base URLs

Environment	Base URL	Endpoint
Production	<code>https://api.radium.cloud</code>	<code>POST /v1/messages</code>
Staging	<code>https://stage.api.radium.cloud</code>	<code>POST /v1/messages</code>

“ **Credentials do not cross over** Staging needs staging credentials. A production key will not authenticate against the staging endpoint, and that is deliberate.

Dedicated deployments

Organisations running on a dedicated Radium deployment use the host supplied to them, and every endpoint pattern on this page applies unchanged. If you need one, [talk to us](#).

OpenAI-compatible Chat Completions

Use this only when your application expects an OpenAI-compatible Chat Completions API. Anthropic-compatible clients should use `POST /v1/messages` above.

The base URL for this path includes `/v1`:

```
https://api.radium.cloud/v1
```

```
curl -s https://api.radium.cloud/v1/chat/completions \  
-H "content-type: application/json" \  
-H "x-api-key: $RADIUM_API_KEY" \  

```

```
-d '{
  "model": "hal-1.0",
  "max_tokens": 2048,
  "temperature": 0,
  "messages": [
    { "role": "user", "content": "Hello, Radium!" }
  ]
}'
```

Bearer authentication works here too, and the model identifiers are the same three.

Every endpoint

Endpoint summary

Use case	Method	Endpoint
Messages, production	POST	<code>https://api.radium.cloud/v1/messages</code>
Messages, staging	POST	<code>https://stage.api.radium.cloud/v1/messages</code>
Chat Completions, production	POST	<code>https://api.radium.cloud/v1/chat/completions</code>
Chat Completions, staging	POST	<code>https://stage.api.radium.cloud/v1/chat/completions</code>

Claude Code

Claude Code points at the Anthropic-compatible base URL and adds the request path itself, so configure it with the base URL alone:

```
export ANTHROPIC_BASE_URL="https://api.radium.cloud"
export ANTHROPIC_MODEL="hal-1.0"
export ANTHROPIC_API_KEY="$RADIUM_API_KEY"
```

Check it:

```
claude --print --model hal-1.0 "Say only: radium hal connected."
```

“ **The one that catches everybody** Do not put `/v1/messages` into `ANTHROPIC_BASE_URL`. Claude Code appends it, and a base URL with the path

already on it returns a 404 on every request.

The full setup, including the scripted install, model selection, and troubleshooting, is on [Run Claude Code on Radium](#).

Next

- [Run Claude Code on Radium](#)
- [Tool calling](#), for the `tool_use` and `tool_result` contract
- [MCP and Radium](#)
- [Errors and troubleshooting](#)

Created 2026-08-20 17:11:14 UTC by Admin
Updated 2026-08-20 17:13:23 UTC by Admin