

Use Agno with Radium

Agno's `OpenAILike` model is designed for OpenAI-compatible APIs. Configure it with Radium's endpoint and use it with normal Agno agents.

Before you start

You need Python 3.10 or newer, a Radium API key, and a terminal or PowerShell window.

1. Create a project and install Agno

macOS or Linux:

```
mkdir radium-agno
cd radium-agno
python3 -m venv .venv
source .venv/bin/activate
python -m pip install agno openai
```

Windows PowerShell:

```
mkdir radium-agno
cd radium-agno
py -m venv .venv
.venv\Scripts\Activate.ps1
python -m pip install agno openai
```

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.py`

```
import os

from agno.agent import Agent
from agno.models.openai.like import OpenAILike

api_key = os.getenv("RADIUM_API_KEY")
if not api_key:
    raise SystemExit("Set RADIUM_API_KEY before running this program.")

agent = Agent(
    model=OpenAILike(
        id=os.getenv("RADIUM_MODEL", "hal-1.0"),
        api_key=api_key,
        base_url="https://api.radium.cloud/v1",
        max_tokens=512,
        temperature=0,
        timeout=90,
    ),
    instructions="Follow the user's instruction exactly.",
)

result = agent.run("Reply with exactly: Radium connected.", stream=False)
if not result.content or not str(result.content).strip():
    raise RuntimeError("Radium returned no visible text.")
print(f"RADIUM_RESPONSE: {str(result.content).strip()}")
```

4. Run it

```
python main.py
```

Expected output:

```
RADIUM_RESPONSE: Radium connected.
```

Choose a model

```
export RADIUM_MODEL="clarke-1.0" # or hal-1.0 or tycho-1.0  
python main.py
```

Migrate an existing Agno agent

Replace its current model:

```
model = OpenAILike(  
    id="hal-1.0",  
    api_key=os.environ["RADIUM_API_KEY"],  
    base_url="https://api.radium.cloud/v1",  
    max_tokens=512,  
)  
agent = Agent(model=model)
```

Your agent instructions, knowledge sources, storage, and tools can remain unchanged. External tools keep their own dependencies and credentials.

Troubleshooting

- Authentication errors: verify the key in the active terminal.
- `404`: check the endpoint and exact model ID.
- Import errors: install both pinned packages inside the active virtual environment.
- Tool errors: validate each tool's own credentials separately from the Radium connection.

Validation

Version note: these instructions were verified with Python 3.12.7, `agno==2.9.0`, and `openai==2.54.0` on August 20, 2026. You do not need those exact package versions. Text generation and callable-tool execution passed with `hal-1.0`, `clarke-1.0`, and `tycho-1.0`.

Reference: [Agno OpenAI-compatible models](#).

Next

- [API quickstart](#), for calling Radium directly
- [Tool calling](#), for the `tool_use` and `tool_result` contract

Created 2026-08-23 21:32:16 UTC by Admin

Updated 2026-08-23 21:52:27 UTC by Admin