# ? How to Publish an Adapter for LangChain Agents

Do you use LangChain to build agents? With A2A Gateway, you can expose them as interoperable services by publishing a simple adapter.

This guide walks you through wrapping a LangChain agent with FastAPI so it speaks the A2A protocol.

## ? What Is an Adapter?

An **adapter** is a small web wrapper that:

- Accepts A2A task input
- Passes it to a LangChain chain or agent
- Returns the output in A2A format

This enables your agent to receive and execute tasks from other agents ? just like an API.

## ? Step 1: Install LangChain and FastAPI

```bash
pip install langchain openai fastapi uvicorn
```

## ?? Step 2: Create the Adapter

```python
# adapter_langchain.py
from fastapi import FastAPI, Request
from pydantic import BaseModel
from langchain.llms import OpenAI

app = FastAPI()
```

```
class A2ATask(BaseModel):
    input: str

@app.post("/execute")
async def execute(task: A2ATask):
    llm = OpenAI(temperature=0)
    response = llm(task.input)
    return {"output": response}
```

Run it:

```bash
uvicorn adapter_langchain:app --port 5010 --reload
```

## ? Step 3: Define the Agent Card

```json
{
  "id": "langchain-wrapper",
  "name": "LangChain LLM Wrapper",
  "description": "Wraps a basic LangChain agent for A2A compatibility",
  "type": "llm",
  "entrypoint": "http://localhost:5010/execute",
  "capabilities": ["chat", "completion"],
  "language": "python",
  "version": "0.1.0",
  "author": "you"
}
```

## ? Step 4: Publish the Adapter

```bash
a2a-gateway publish --file agent_card.json
```

Now your LangChain agent can be reached by others through A2A!

## ? Step 5: Test Interoperability

```bash
a2a-gateway task --to langchain-wrapper --input "What is the capital of France?"
```

## ? Bonus: Extend with Custom Chains

You can go beyond OpenAI ? wrap any LangChain `Chain`:

```python
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate
prompt = PromptTemplate.from_template("Translate this to French: {text}")
chain = LLMChain(prompt=prompt, llm=OpenAI())

@app.post("/execute")
async def execute(task: A2ATask):
    result = chain.run(text=task.input)
    return {"output": result}
```

## ? Share It

Publish your adapter:

- As a GitHub repo with README
- On the public A2A registry (coming soon)
- On Discord `#agent-adapters`

Let others build on it!