

OpenAI API Configuration Notes

Overview

The ADK tutorial notebooks are configured to use **OpenAI API** for accessing LLMs. This provides students with flexibility in model choice and avoids quota limitations.

Changes Made

1. Requirements File (`requirements.txt`)

Added: - `litellm>=1.0.0` - Enables ADK to work with OpenAI and other LLM providers - `openai>=1.0.0` - OpenAI Python client library

Removed: - `google-genai>=0.3.0` - No longer needed for OpenAI

2. Both Notebooks Updated

Installation Cell

```
!pip install -q google-adk litellm openai python-dotenv nest-asyncio
```

Imports

Added:

```
from google.adk.models.lite_llm import LiteLlm
```

API Key Configuration

- Changed from `GOOGLE_API_KEY` to `OPENAI_API_KEY`
- Updated Colab secrets name to `OPENAI_API_KEY`
- Updated API key URL to: <https://platform.openai.com/api-keys>
- Added model selection configuration

Model Selection

Added configurable model selection:

```
OPENAI_MODEL = "gpt-5-nano" # Can be changed to other models
```

Available Models: - `gpt-4o` - Most capable model - `gpt-5-nano` - Default model, cost-efficient and recommended for students - `gpt-4o-mini` - Alternative cost-efficient option - `gpt-4-turbo` - Previous generation - `gpt-3.5-turbo` - Fastest, most economical

Agent Creation

Changed from:

```
LlmAgent(  
    model="gemini-2.5-flash",  
    ...  
)
```

To:

```
LlmAgent(  
    model=LiteLlm(model=f"openai/{OPENAI_MODEL}"),  
    ...  
)
```

3. Tool Compatibility

OpenAI API **supports default parameter values** in function tools. However, for clarity and consistency in the tutorial, we set default values inside functions rather than in the function signature.

You can use default parameters with OpenAI if needed:


```
def search_knowledge_base(query: str, max_results: int = 3) -> Dict[str, any]:  
    # This works with OpenAI!
```

Student Instructions

Getting an OpenAI API Key

1. Go to <https://platform.openai.com/api-keys>
2. Sign up or log in
3. Click "Create new secret key"
4. Copy the key and save it securely
5. Add it to Colab secrets or paste when prompted

Setting Up in Colab

Option 1: Colab Secrets (Recommended) 1. Click the  icon in the left sidebar 2. Click "+ Add new secret" 3. Name: `OPENAI_API_KEY` 4. Value: Your API key 5. Toggle on notebook access

Option 2: Direct Input The notebook will prompt you to enter the key securely when you run the authentication cell.

Choosing a Model

Edit the model selection cell to change models:

```
# For default (recommended)  
OPENAI_MODEL = "gpt-5-nano"
```

```
# For most capable (more expensive)
OPENAI_MODEL = "gpt-4o"

# For alternative cost-efficient option
OPENAI_MODEL = "gpt-4o-mini"

# For fastest/cheapest
OPENAI_MODEL = "gpt-3.5-turbo"
```

Cost Considerations

Approximate Costs (as of 2025): - GPT-4o: \$2.50 per 1M input tokens, \$10 per 1M output tokens - GPT-5-nano: Cost-efficient pricing (similar to gpt-4o-mini) - GPT-4o-mini: \$0.15 per 1M input tokens, \$0.60 per 1M output tokens - GPT-3.5-turbo: \$0.50 per 1M input tokens, \$1.50 per 1M output tokens

For Students: - Recommended: `gpt-5-nano` offers excellent performance at low cost - A typical notebook session uses ~50-200K tokens total - Estimated cost per session: \$0.01 - \$0.05 with gpt-5-nano

Benefits of OpenAI API

✅ **No quota limitations** - Students won't hit rate limits during class ✅ **Model choice** - Students can select models based on budget/performance ✅ **Default parameters** - OpenAI supports optional parameters in tools ✅ **Wide compatibility** - OpenAI API is widely documented and supported ✅ **Cost control** - Students can monitor usage in OpenAI dashboard ✅ **Reliability** - Stable API with high availability

Testing

Both notebooks work correctly with OpenAI API: - ✅ API key configuration - ✅ Agent creation with gpt-5-nano - ✅ Chat interactions - ✅ Session management - ✅ Tool calling (Notebook 2) - ✅ All exercises

Troubleshooting

"Invalid API key" error

- Verify the API key is correct
- Check that it starts with `sk-`
- Ensure billing is set up in OpenAI account

"Model not found" error

- Check model name spelling
- Verify your account has access to that model
- Try switching to `gpt-4o-mini` or `gpt-3.5-turbo` (widely available)

Rate limit errors

- OpenAI has rate limits based on account tier
- Wait a few seconds and retry
- Consider upgrading OpenAI account tier

Future Enhancements

With OpenAI integration, students can now: 1. **Compare models** - Run same agent with different models 2. **Add tool parameters** - Use default values in function definitions 3. **Monitor costs** - Track usage in OpenAI dashboard 4. **Scale up** - Upgrade to more powerful models as needed

Using Alternative LLM Providers

Thanks to LiteLLM, you can easily switch to other providers:

1. **Azure OpenAI:**

```
python OPENAI_MODEL = "azure/gpt-4o"  
os.environ["AZURE_API_KEY"] = your_key
```
2. **Anthropic Claude:**

```
python OPENAI_MODEL = "claude-3-5-  
sonnet-20241022" os.environ["ANTHROPIC_API_KEY"] = your_key
```

3. **Google Gemini (via Vertex AI):** `python OPENAI_MODEL = "gemini/gemini-2.0-flash-exp" # Configure Google Cloud credentials`

Questions? Check the official ADK documentation: - Models & Authentication: <https://google.github.io/adk-docs/agents/models/> - LiteLLM Integration: https://docs.litellm.ai/docs/tutorials/google_adk