Lecture 19: Using AI APIs – Focus on OpenAI

---

1 What is an API?

- **API (Application Programming Interface)**: A bridge between client & server
- **AI APIs**: Connect apps to AI models (like ChatGPT)
- Sends prompts → gets AI-generated responses

 Example: You (Client) → Waiter (API) → Kitchen (Server)

---

2 What is an API Key?

- **API Key** = Secret token
- Used for:
  - Authentication
  - Usage tracking
  - Preventing abuse

How to Get OpenAI API Key

1. Log into https://platform.openai.com
2. Go to **API Keys**
3. Click **+ New Key**
4. Name it (e.g., "Fera Practice Key")
5. Copy and store it safely (can't view again!)

---

3 Secure Storage with `.env`

✅ NEVER hardcode API key like this:

```
python
CopyEdit
api_key = "sk-abc123" ✗
```

✅ INSTEAD: Use a `.env` file

```
ini
CopyEdit
OPENAI_API_KEY=sk-abc123
```

And load it using:

```
python
CopyEdit
from dotenv import load_dotenv
import os

load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")
```

---

4 Install Required Packages

```
bash
CopyEdit
pip install openai
pip install python-dotenv
```

---

5 Code Example: Use OpenAI API

```
python
CopyEdit
import openai
import os
from dotenv import load_dotenv

load_dotenv()
openai.api_key = os.getenv("OPENAI_API_KEY")

def getAIResponse(prompt):
    try:
        response = openai.ChatCompletion.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "You are a helpful
assistant."},
                {"role": "user", "content": prompt}
            ],
            temperature=0.7,
            max_tokens=150
        )
        return response['choices'][0]['message']['content']
    except Exception as e:
        print("Error:", e)
```

---

6 Key Request Components

Component	Description
Client	Sends request using API key
Prompt	User's question or instruction
Model	E.g., gpt-3.5-turbo, gpt-4
Messages	Chat history: system, user, assistant
Parameters	e.g., temperature, max_tokens
Response	AI-generated output

---

7 API Rate Limits & Quotas

- You may get errors like:
  - **401 Unauthorized** – wrong/missing key
  - **429 Too Many Requests** – rate limit hit
  - **Quota Exceeded** – usage limit reached

🛠 Use try-except to handle these errors safely.

---

8 Best Practices ✅

- ✔ Store keys in `.env`
- ✔ Use `try-except` for errors
- ✔ Read API docs carefully
- ✔ Monitor usage and credits
- ✔ Never expose secrets in code
- ✔ Test API key before calling it in loops

---

9 Common Question

❓ *How do we use an API key securely?*

✅ **A:** Store it in `.env`, and use `os.getenv()` to access it. Never paste it directly in `.py` files.

🔟 Final Thoughts

- You now know how to:
  - Generate & secure API keys
  - Install and use `openai` Python library
  - Send chat prompts and receive responses
  - Handle errors and follow best practices

➡️ **Next Lecture:** Using **Google AI APIs** & integration with Python