Exercises to Get You Started with GPT-3.5 API for Applications

These exercises will guide you through exploring the GPT-3.5 API's capabilities and using it to build applications. Refer to the official guide (https://platform.openai.com/docs/models) for detailed explanations.

Getting Started (Exercises 1-5):

1. **Account Setup (Visit https://platform.openai.com/ and sign up for an OpenAI account.)**
    - Create an OpenAI account to access the GPT-3.5 API.
2. **API Key Retrieval (Go to your account settings and navigate to the "API Keys" section.)**
    - Locate your unique API key, which is essential for making API requests.
3. **Library Installation (Use `pip install openai` in your terminal.)**
    - Install the official `openai` library to interact with the GPT-3.5 API from Python code.
4. **Library Import (In your Python script, use `import openai`.)**
    - Import the `openai` library in your code to access its functionalities.
5. **API Key Configuration (Set the `openai.api_key` variable to your unique API key.)**
    - Authenticate your code by assigning your API key to the `openai.api_key` variable.

Text Generation Exercises (Exercises 6-10):

6. **Simple Text Generation (Use `response = openai.Completion.create(engine="text-davinci-003", prompt="Write a poem about nature.", max_tokens=100)`.)**
    - Generate a poem about nature with maximum 100 tokens (words) using the "text-davinci-003" engine.
7. **Conditional Text Generation (Use `prompt` and `temperature` parameters to influence the output.)**
    - Generate a news article with a specific style or tone by adjusting the `prompt` and `temperature` parameters.

8. **Code Generation (Utilize the "text-davinci-003" engine for code-related prompts.)**
    - Generate a Python function that calculates the factorial of a number using the "text-davinci-003" engine.
9. **Translation (Specify the `source` and `target` languages in the request.)**
    - Translate a sentence from English to Spanish using the GPT-3.5 API.
10. **Multiple Choice Question Generation (Craft a prompt that asks a question and provides answer options.)**
    - Generate a multiple-choice question about a specific topic, including answer choices.

Fine-Tuning Exercises (Exercises 11-15):

11. **Fine-Tuning Dataset Preparation (Create a JSONL file with examples and their desired completions.)**
    - Prepare a JSONL file containing data points where each line has an "prompt" and "completion" pair for fine-tuning.
12. **Fine-Tuning File Upload (Use `openai.File.create` to upload your JSONL file.)**
    - Upload your fine-tuning dataset to OpenAI using the `openai.File.create` function.
13. **Fine-Tuning Model Creation (Specify the fine-tuned model name and purpose.)**
    - Create a new fine-tuned model by specifying its name and purpose.
14. **Fine-Tuning Model Training (Initiate training with `openai.FineTune.create`.)**
    - Trigger the training process for your fine-tuned model using the `openai.FineTune.create` function.
15. **Fine-Tuning Evaluation (Use the `openai.Completion.create` endpoint with the fine-tuned model engine.)**
    - Evaluate your fine-tuned model's performance by making API requests with its specific engine name.

Application Building Exercises (Exercises 16-20):

16. **Building a Chatbot (Use `Completion` endpoint for text generation and conditional responses.)**

- Develop a simple chatbot that interacts with users by generating responses based on their input.

17. **Content Summarization Tool (Utilize GPT-3.5 to condense lengthy text into a concise summary.)**
    - Create an application that summarizes lengthy articles or documents by processing them through the API.
18. **Creative Writing Assistant (Leverage GPT-3.5 to generate ideas, story outlines, or character descriptions.)**
    - Build a tool that assists writers by generating creative content prompts or story elements.