```
session = requests.Session()
session.keep_alive = False  # Close the connection pool
adapter = requests.adapters.HTTPAdapter(max_retries=5)
session.mount('https://', adapter)
session.mount('http://', adapter)
```

4) Define the structure used to wrap the data sent during API calls. User access to RKLLM-Server-Flask will be encapsulated within this structure.

```
# Prepare the data to be sent
# model: the model defined by the user when setting up RKLLM-Server,
which has no effect here
# messages: the questions input by the user; supports adding multiple
questions to messages
# stream: whether to enable streaming dialogue, same as the OpenAI
interface
data = {
    "model": 'your_model_deploy_with_RKLLM_Server',
    "messages": [{"role": "user", "content": user_message}],
    "stream": is_streaming
    }
```

5) Send a request to the RKLLM-Server-Flask server and retrieve the returned data.

```
responses = session.post(server_url, json=data, headers=headers,
stream=is_streaming, verify=False)
```

6) Define the parsing method for the returned data structure. Since RKLLM-Server-Flask follows the format of the returned struct from the OpenAI-API, parsing operations are required in actual usage.

```
# Parse the response
# Non-streaming transmission
if not is_streaming:
       if responses.status_code == 200:
       print("Q:", data["messages"][-1]["content"])
       print("A:", json.loads(responses.text)["choices"][-
1]["message"]["content"])
   else:
       print("Error:", responses.text)

# Streaming transmission
else:
   if responses.status_code == 200:
       print("Q:", data["messages"][-1]["content"])
       print("A:", end="")
       for line in responses.iter_lines():
if line:
             line = json.loads(line.decode('utf-8'))
             if line["choices"][-1]["finish_reason"] != "stop":
                print(line["choices"][-1]["delta"]["content"], end="")

                sys.stdout.flush()
   else:
       print('Error:', responses.text)
```

The overall process from steps 1 to 6 represents the API access method for the RKLLM-Server-Flask server. Users can develop custom functionalities based on the example code provided above. It is