How Do We Interface With External Resources?

```
import requests

url = "https://v2.jokeapi.dev/joke/Any?safe-mode&type=twopart"

response = requests.get(url)

print(response.json()["setup"])
print(response.json()["delivery"])
```

```
Why did the database administrator leave his wife?
She had one-to-many relationships.
```

```
response.json()
```

```
{'error': False,
 'category': 'Programming',
 'type': 'twopart',
 'setup': 'Why did the database administrator leave his wife?',
 'delivery': 'She had one-to-many relationships.',
 'flags': {'nsfw': False,
  'religious': False,
  'political': False,
  'racist': False,
  'sexist': False,
  'explicit': False},
 'safe': True,
 'id': 265,
 'lang': 'en'}
```

```
import requests
def give_joke(category : str):
    """
    Joke categories. Supports: Any, Misc, Programming, Pun, Spooky, Christmas.
    """

    url = f"https://v2.jokeapi.dev/joke/{category}?safe-mode&type=twopart"
    response = requests.get(url)
    print(response.json()["setup"])
    print(response.json()["delivery"])
```

```
USER_QUERY = "Hey! Can you get me a joke for this december?"
```

```
from utils import query_raven

raven_functions = \
f'''
def give_joke(category : str):
    """
    Joke categories. Supports: Any, Misc, Programming, Dark, Pun, Spooky, Christmas.
    """

User Query: {USER_QUERY}<human_end>
'''
call = query_raven(raven_functions)
```

```
exec(call)
```

```
What says Oh Oh Oh?
Santa walking backwards!
```

Writing A Tool That Uses OpenAPI APIs

```
#!wget https://raw.githubusercontent.com/open-meteo/open-meteo/main/openapi.yml
```

```
import yaml
import json

# Read the content of the file
```

```
with open('openapi.yml', 'r') as file:
    file_content = file.read()
file_content = file_content.replace("int\n", "number\n")
file_content = file_content.replace("float\n", "number\n")
data = yaml.safe_load(file_content)

data["servers"] = [{"url":"https://api.open-meteo.com"}]

with open('openapi.json', 'w') as file:
    json_content = json.dump(data, file)
```

```
!openapi-python-generator openapi.json ./api_specification_main/
```

```
Generating data from openapi.json
```

```
from api_specification_main.services.WeatherForecastAPIs_service\
    import get_v1forecast
```

```
user_query = "Hey how is the current weather and windspeed in New York?"
```

```
import inspect
signature = inspect.signature(get_v1forecast)
docstring = \
'''
Requires the latitude and longitude.
Set current_weather to True to get the weather.
Set hourly or daily based on preference.
'''

raven_prompt = \
f'''
Function:
{get_v1forecast.__name__}{signature}
"""{docstring}"""

User Query: {user_query}<human_end>'''

print (raven_prompt)
```

```

Function:
get_v1forecast(latitude: float, longitude: float, hourly: Optional[List[str]] = None, daily: Optional[List[str]] = None, current_we
"""
Requires the latitude and longitude.
Set current_weather to True to get the weather.
Set hourly or daily based on preference.
"""

User Query: Hey how is the current weather and windspeed in New York?<human_end>
```

```
from utils import query_raven
call = query_raven(raven_prompt)
print (call)
```

```
get_v1forecast(latitude=40.7128, longitude=-74.0060, current_weather=True)
```

```
eval(call)
```

```
{'latitude': 40.710335,
 'longitude': -73.99307,
 'generationtime_ms': 0.08893013000488281,
 'utc_offset_seconds': 0,
 'timezone': 'GMT',
 'timezone_abbreviation': 'GMT',
 'elevation': 32.0,
 'current_weather_units': {'time': 'iso8601',
  'interval': 'seconds',
  'temperature': '°C',
  'windspeed': 'km/h',
  'winddirection': '°',
  'is_day': '',
  'weathercode': 'wmo code'},
 'current_weather': {'time': '2024-06-25T16:15',
  'interval': 900,
  'temperature': 31.7,
```

```
 'windspeed': 15.7,
 'winddirection': 281,
 'is_day': 1,
 'weathercode': 0}}
```

```
Start coding or generate with AI.
```