Flask with External Web APIs

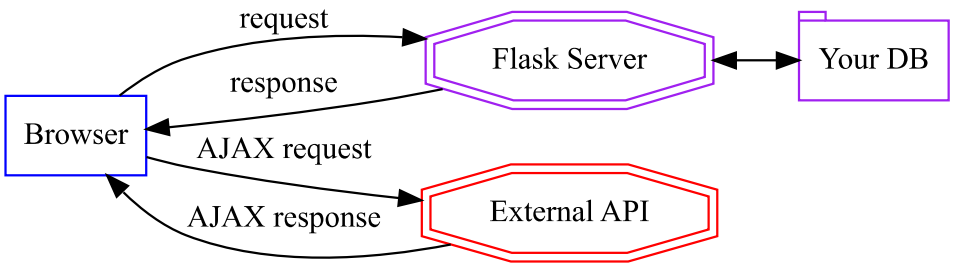
Download Demo Code <../flask-ext-apis-demo.zip>

API Requests

Two ways to talk with APIs:

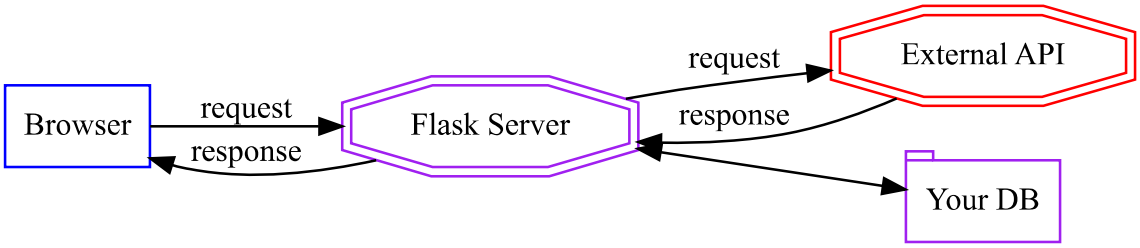
- Client-side requests (via AJAX)
- Server-side requests

Why Use Client-Side Requests?



- You can do easily using AJAX libraries
- Don't have to involve Flask in the API
- Can be faster: browser could talk directly to, say, Google Maps

Why Use Server-Side Requests?



- Same-Origin Policy may prevent browser requests
- Easier for server to store/process the data
  - e.g. have Flask requests restaurants and store in SQL database
- Need password to access API
  - If API uses password & we make request in browser JS, people could learn password from reading JS

iTunes API

```
$ curl -i
  'https://itunes.apple.com/search?term=billy+bragg&limit=3'
{
 "resultCount":5,
 "results": [
  {"wrapperType":"track", "kind":"song", "artistId":"163251",
 ...
```

iTunes API Help <https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/#searchexamples>

Returns JSON responses

Python Requests

```
(venv) $ pip install requests
```

GET Requests

`requests.get(url, params)`

```
import requests

resp = requests.get(
            "https://itunes.apple.com/search",
            params={"term": "billy bragg", "limit": 3}
       )

print(resp.json())
```

POST Requests

`requests.post(url, data, json)`

data
: Dictionary of data to send in traditional web from format

json
: Dictionary of data to send as a JSON string

Most modern APIs expect to receive JSON, not traditional web form format.

Responses

Both ***.get()*** and ***.post()*** return a Response instance

.text
Text of response

.status_code
Numeric status code (200, 404, etc)

.json()
Convert JSON response text to Python dictionary

API Keys/Secrets

Many APIs require “keys” and “secrets”

(similar to a “username” and “password”)

Why Do They Need API Keys?

- The API provides access to confidential data or sensitive methods
  - Only you should be able to send tweets from your Twitter account
- The API costs money to use
  - They need to know who to charge
- They want to limit abuse
  - Google Maps is free, but they want to keep you to from abusing it

Where Do You Get API Keys?

Typically: you register on their site.

The process is different for every site.

Example: YouTube API Key <https://console.developers.google.com/apis/credentials/>

How Do You Use API Keys?

It varies by different APIs

For example, if this API needed a secret key sent with requests, they might expect as a URL parameter:

```
requests.get("http://some-api.com/search",
    params={"key": "dhf489tuhdfhdskfsdfsd34tg",
            "isbn": "4675436632"})
```

Or, they might need complex encoding — varies by API!

Read the API docs!

Keeping Your Secrets

What's the potential problem?

app.py

```
from flask import Flask

API_SECRET_KEY = "jdfghfkgdg9345dkjfgdfg"

app = Flask(__name__)

...
```

You'll want to store this file in Git — and probably GitHub

You don't want the world to learn your API key!

Strategy: store the key info in a small, separate file

Import *that* file into your ***app.py***

Don't check that file into Git!

Example

secrets.py

```
API_SECRET_KEY = "jdfghfkgdg9345dkjfgdfg"
```

app.py

```
from flask import Flask
from secrets import API_SECRET_KEY

app = Flask(__name__)

...
```

.gitignore

```
secrets.py
```

Make sure it *never* gets into your Git!

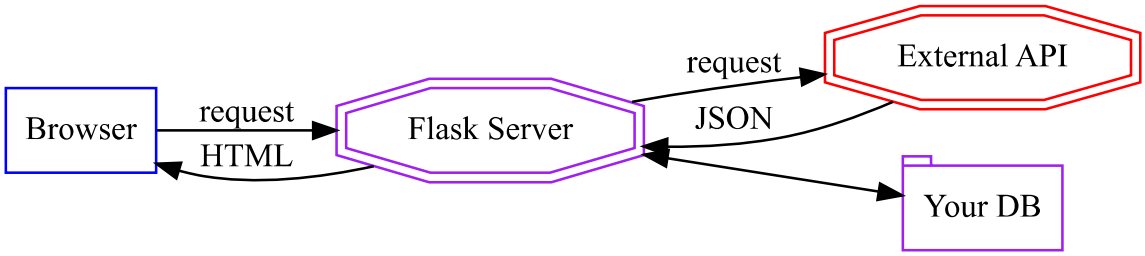
```
$ git status
# Should NOT show up here at all

$ git add .

$ git status
# Should NOT show up here at all
```

```
$ git commit ...
```

External APIs and Flask

How External APIs Get Used in Flask

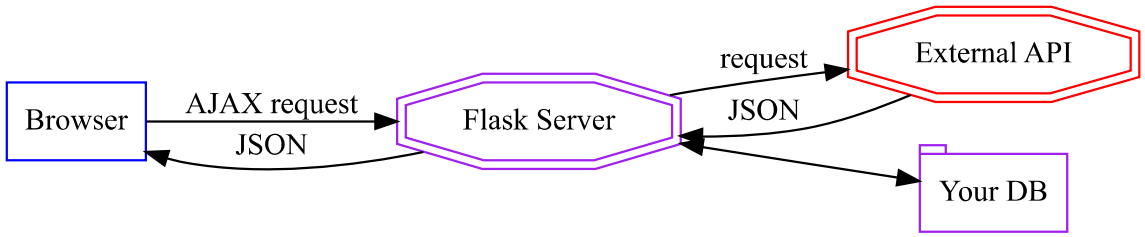
Sometimes Flask gets JSON data and it returns HTML:



app.py

```
@app.route("/book-info")
def show_book_info():
    """Return page about book."""

    isbn = request.args["isbn"]

    resp = requests.get("http://some-book-api.com/search",
        params={"isbn": isbn, "key": API_SECRET_KEY})

    book_data = resp.json()

    # using the APIs JSON data, render full HTML page
    return render_template("book_info.html", book=book_data)
```

Sometimes Flask gets JSON data and JSON data to front end:



app.py

```
@app.route("/book-data")
def show_book_info():
    """Return info about book."""

    isbn = request.args["isbn"]

    resp = requests.get("http://some-book-api.com/search",
```

```
        params={"isbn": isbn, "key": API_SECRET_KEY})

    book_data = resp.json()

    # using the APIs JSON data, return that to browser

    return jsonify(book_data)
```

This is helpful if you can't make request info directly from browser — because of Same-Origin-Policy or need to keep key/secret out of browser

API Libraries

Some popular APIs have specialized libraries *(sometimes known as SDKs)* written for a specific programming language that can help out.

For example, there is a Python library for calling the Twitter API:

Python-Twitter <https://github.com/bear/python-twitter>