

Introduction to Software Systems (CS6.201)

Summer 2021, IIIT Hyderabad

13 July, Tuesday (Lecture 14) – Python 5

Taught by Abhinav Gupta

Flask

Introduction

Flask is a library that allows one to create REST APIs.

When we try to open a site using the browser, the browser sends a **GET** request to the server and the server responds with the information requested. These communications are made using a protocol like HTTP or HTTPS.

Verbs

GET is an example of a verb; others are POST and DELETE.

- GET retrieves the data for the requester to use.
- POST allows the requester to modify the retrieved data.
- DELETE allows the user to remove content.

Writing an Application

First, we need to create and name an app in Flask:

```
app = Flask(__name__)
```

This command gives a unique name to the application `app`.

Now, in order to tell the application what request it needs to make, we need to use a decorator with a method. A decorator is written in the following way:

```
@app.route("/")
def home():
    return "HelloWorld"
```

In order to run this application, we use the `run()` method:

```
app.run(port=5000)
```

Here, we have not specified the type of request – by default, it is **GET**.

We can add other pages (`stores` is a predefined list of dictionaries):

```
@app.route("/json")
def json():
    return jsonify({'key': 'value', 'key2': [1,2,3]})
```

```
@app.route("/store"):
def get_store():
    json_file = jsonify({'stores': stores})
    return json_file
```

Now, we can use the `request` module to send requests using POST.

```
@app.route("/store", methods=['POST'])
def create_store():
    request_data = request.get_json()
    new_store = {'name': request_data['name'], 'items': request_data['items']}
    stores.append(new_store)
    return jsonify(new_store)
```