Flask, REST, JSON

Recommendation Engine Framework Fundamentals

Gene Olafsen

Flask

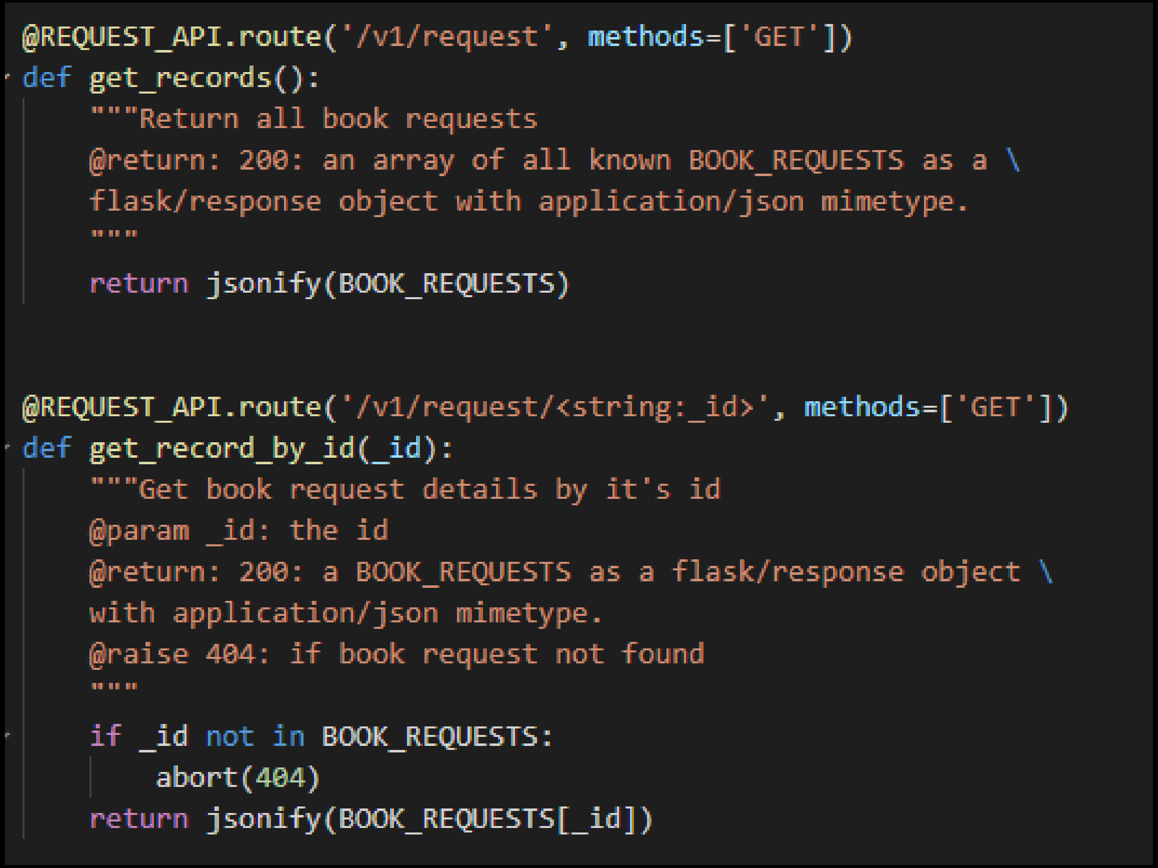
- A web framework.
- A Python module that lets you develop web applications easily.
- Has a small and easy-to-extend core.
- A microframework that doesn’t include an ORM (Object Relational Manager) or requires other API's or frameworks to work out of the box.

REST

- REST is acronym for **RE**presentational **S**tate **T**ransfer.
- It is an architectural style that defines a set of rules in order to create Web Services.
- REST service API's may return XML or JSON responses. Though JSON is increasingly the preferred format because the JavaScript compatible format seamlessly integrates with web client frameworks. (data formats listed)
  - application/json
  - application/xml
  - application/x-wbe+xml
  - application/x-www-form-urlencoded
  - multipart/form-data

Flask URL Routes via Decorated Methods

- todo



```
@REQUEST_API.route('/v1/request', methods=['GET'])
def get_records():
    """Return all book requests
    @return: 200: an array of all known BOOK_REQUESTS as a \
    flask/response object with application/json mimetype.
    """
    return jsonify(BOOK_REQUESTS)


@REQUEST_API.route('/v1/request/<string:_id>', methods=['GET'])
def get_record_by_id(_id):
    """Get book request details by it's id
    @param _id: the id
    @return: 200: a BOOK_REQUESTS as a flask/response object \
    with application/json mimetype.
    @raise 404: if book request not found
    """
    if _id not in BOOK_REQUESTS:
        abort(404)
    return jsonify(BOOK_REQUESTS[_id])
```

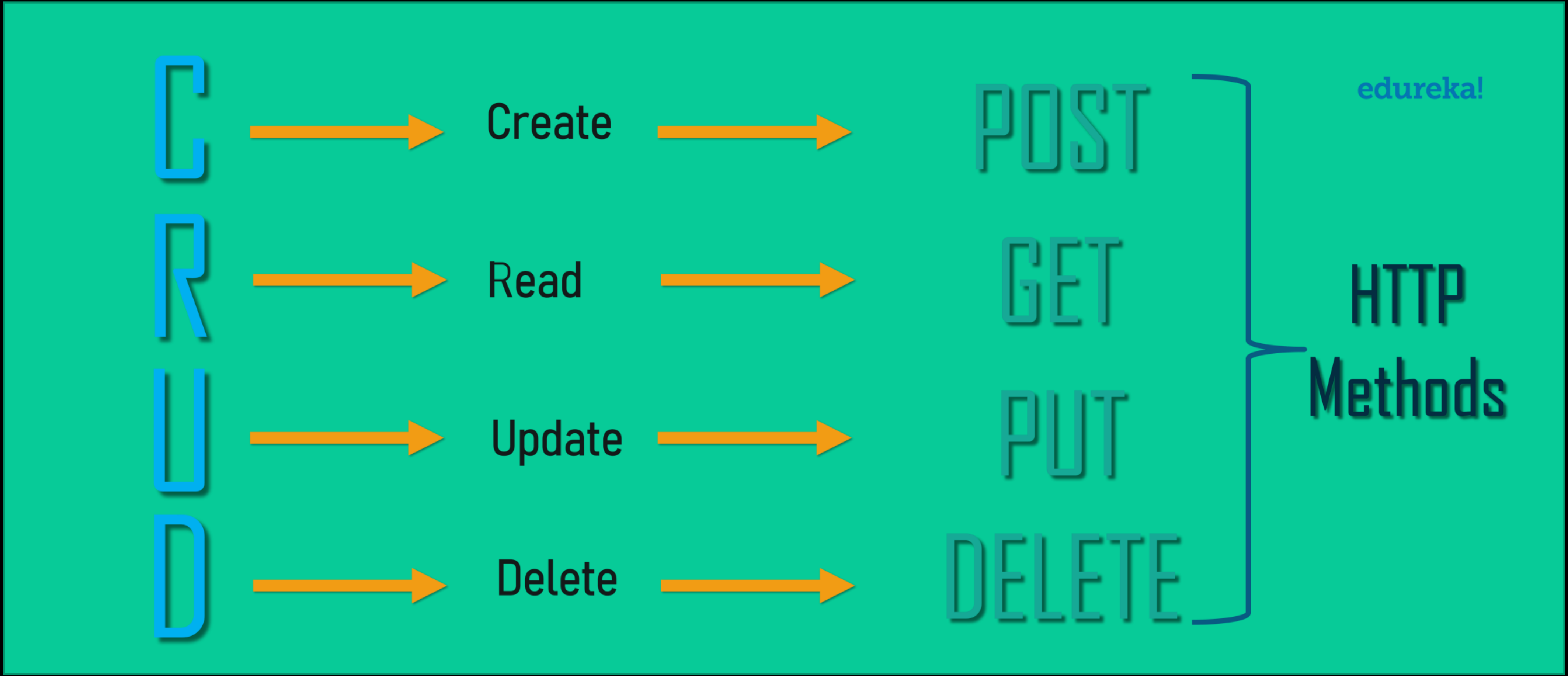
REST Principles

- Stateless
  - State management that is required should take place on the client, not the server.
- Client-Server
  - The loose coupling of the client and server enables each to be developed and enhanced independent of the other.
- Uniform Interface
  - Resources should be uniquely identifiable through a single URL, and only by using the underlying methods of the network protocol.
- Cacheable
  - All resources should allow caching unless explicitly indicated that caching is not possible.
- Layerable System
  - Allows for an architecture composed of multiple layers of servers.

REST Scales

- REST is useful in cloud applications and preferred for most inter-machine communication.
- Stateless components can be freely redeployed if there is a failure.
- REST systems can scale to accommodate load changes. This is because any request can be directed to any instance of a component; there can be nothing saved that has to be remembered by the next transaction.

REST Methods



REST Practices

- Endpoint consistency
  - Paths of endpoints are consistent by following common web standards.
- Versioning
  - Typically a /V1 is part of the initial release of the endpoint.
- Security
  - Secure via secure HTTP protocol.
- Authentication
  - Web Tokens, OAuth 2.0 is common

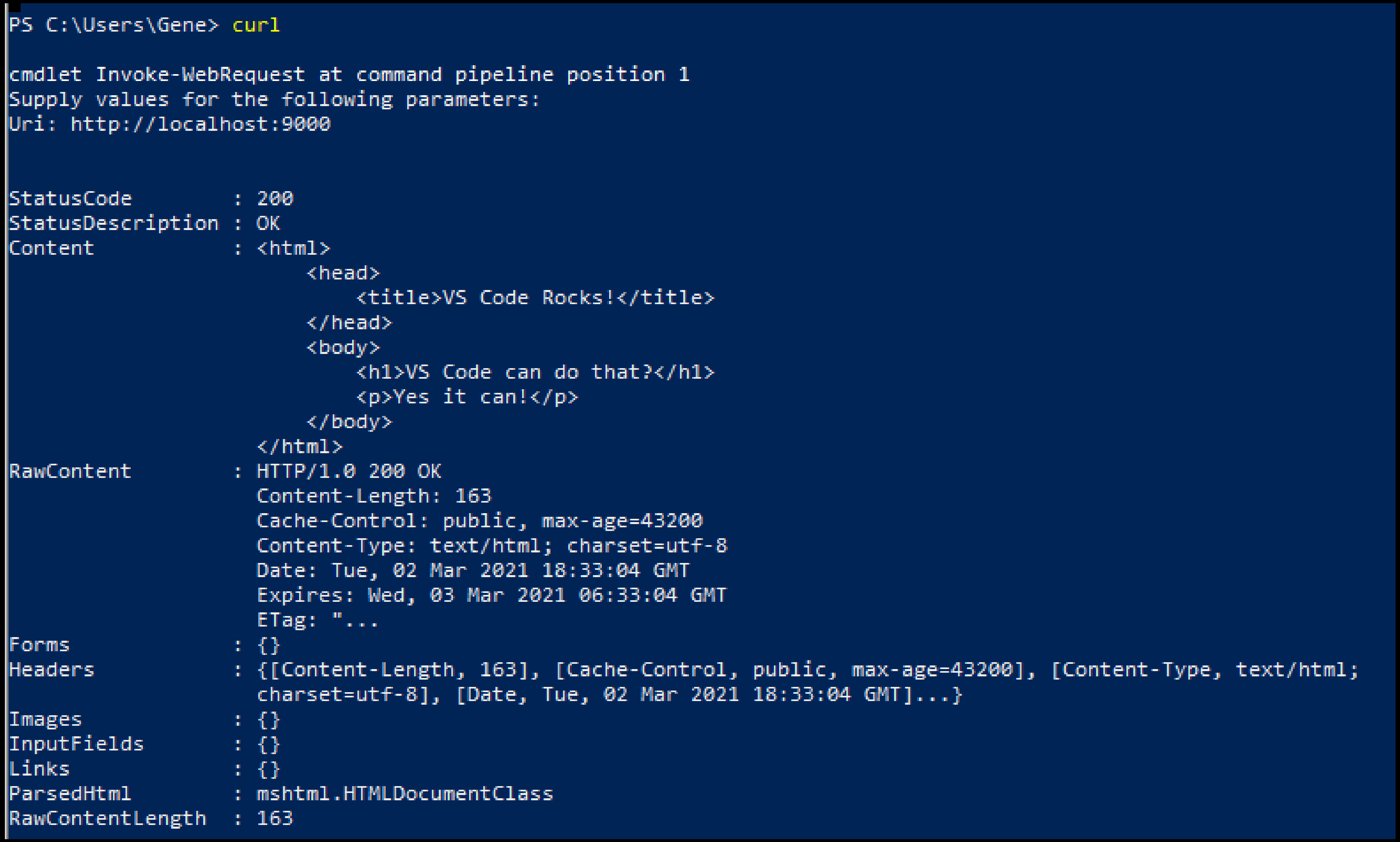
Exercise REST Services

- Curl
- Postman
- Swagger

Curl

- Curl is a command-line tool for transferring data and supports about 22 protocols including HTTP.
- This combination makes it a very good ad-hoc tool for testing our REST services.
- Support:
  - DICT, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT, POP3, POP3S, RTMP, RTMPS, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET and TFTP. curl supports SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, HTTP/2, HTTP/3, cookies, user+password authentication (Basic, Plain, Digest, CRAM-MD5, SCRAM-SHA, NTLM, Negotiate and Kerberos), file transfer resume, proxy tunneling and more

Curl Example



```
PS C:\Users\Gene> curl

cmdlet Invoke-WebRequest at command pipeline position 1
Supply values for the following parameters:
Uri: http://localhost:9000


StatusCode        : 200
StatusDescription : OK
Content           : <html>
                        <head>
                            <title>VS Code Rocks!</title>
                        </head>
                        <body>
                            <h1>VS Code can do that?</h1>
                            <p>Yes it can!</p>
                        </body>
                    </html>
RawContent        : HTTP/1.0 200 OK
                    Content-Length: 163
                    Cache-Control: public, max-age=43200
                    Content-Type: text/html; charset=utf-8
                    Date: Tue, 02 Mar 2021 18:33:04 GMT
                    Expires: Wed, 03 Mar 2021 06:33:04 GMT
                    ETag: "...
Forms             : {}
Headers           : {[Content-Length, 163], [Cache-Control, public, max-age=43200], [Content-Type, text/html;
                    charset=utf-8], [Date, Tue, 02 Mar 2021 18:33:04 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 163
```

Postman

- Send Requests and View Responses
- Create and execute any REST, SOAP, and GraphQL queries from within Postman.
- Application Choices
  - Download the App for Windows, MacOS or linux
  - Install the browser extension

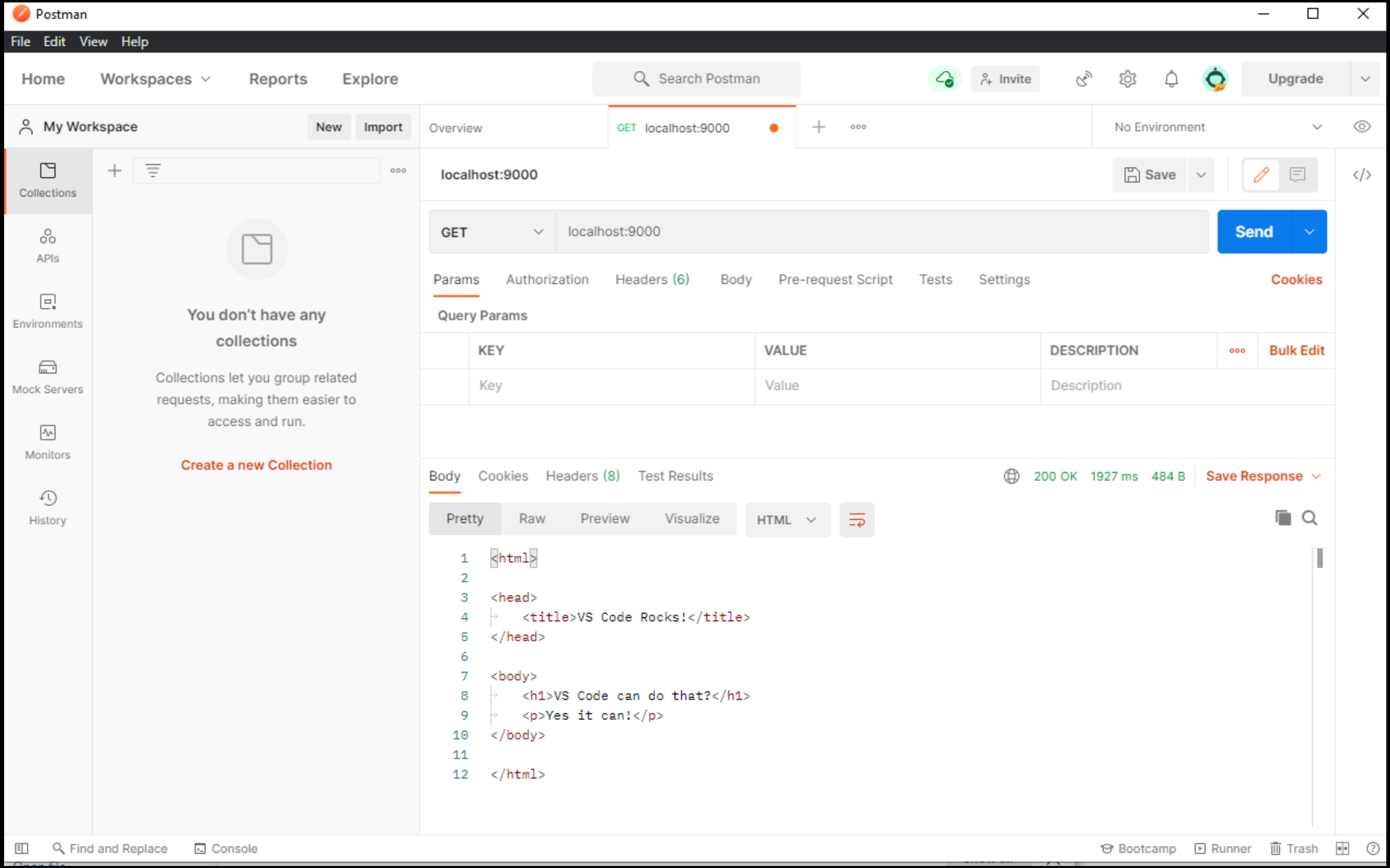
Define Complex Requests

- Send any type of request in Postman. Create and save custom methods and send requests with the following body types:
    - URL-encoded—The default content type for sending simple text data
    - Multipart/form-data—For sending large quantities of binary data or text containing non-ASCII characters
    - Raw body editing—For sending data without any encoding
    - Binary data—For sending image, audio, video, or text files

Postman Collections

- Instead of creating calls manually to send over the command line, all you need is a Postman Collection.
- Import a collection directly or generate one with one click from:
  - An API schema in the RAML, WADL, OpenAPI, or GraphQL format
  - A data file containing the cURL commands

Postman Example


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
My Workspace New Import
Collections APIs Environments Mock Servers Monitors History
Overview
GET localhost:9000
No Environment
Save
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (8) Test Results
Save Response
Pretty Raw Preview Visualize HTML
Find and Replace Console
Bootcamp Runner Trash

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create a new Collection

localhost:9000

GET localhost:9000

200 OK 1927 ms 484 B

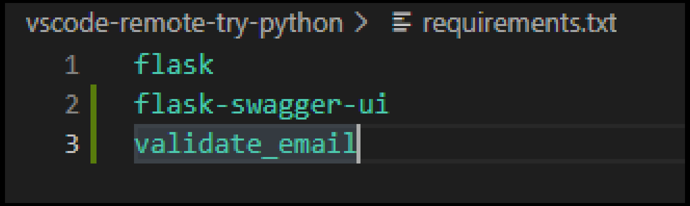
```
<html>

<head>
    <title>VS Code Rocks!</title>
</head>

<body>
    <h1>VS Code can do that?</h1>
    <p>Yes it can!</p>
</body>

</html>
```

Swagger

- Swagger is an Interface Description Language for describing RESTful APIs expressed using JSON.
- Swagger is used together with a set of open-source software tools to design, build, document, and use RESTful web services.
- Swagger includes automated documentation, code generation (into many programming languages), and test-case generation.

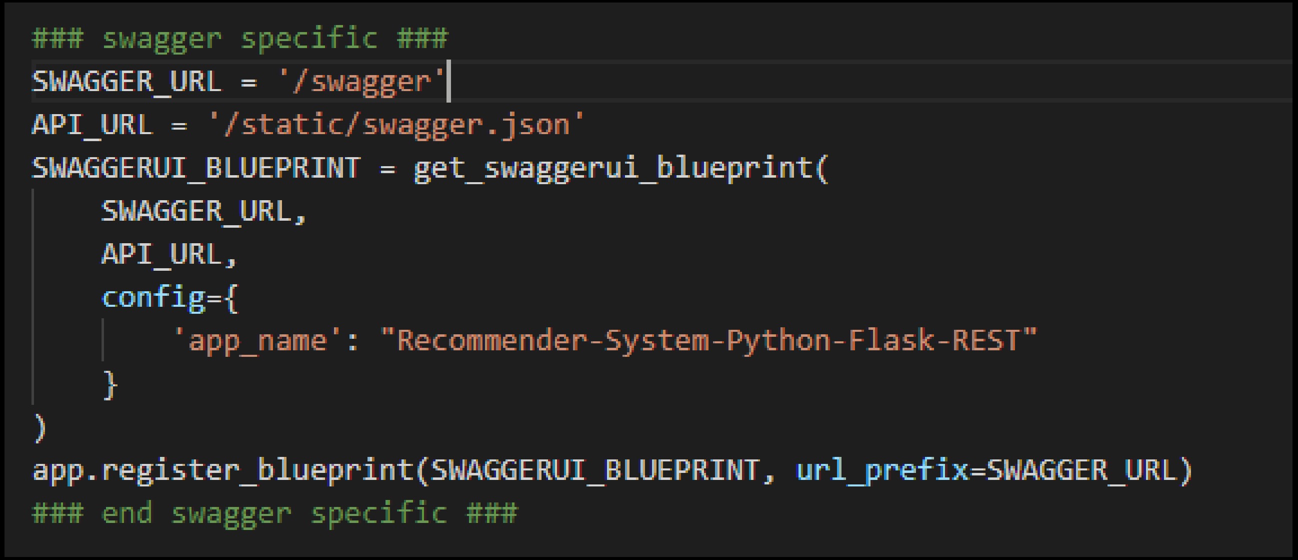
Enable Swagger

- 1. Add flask-swagger-ui library to requirements.txt
- 2. Boilerplate code to render the Swagger page



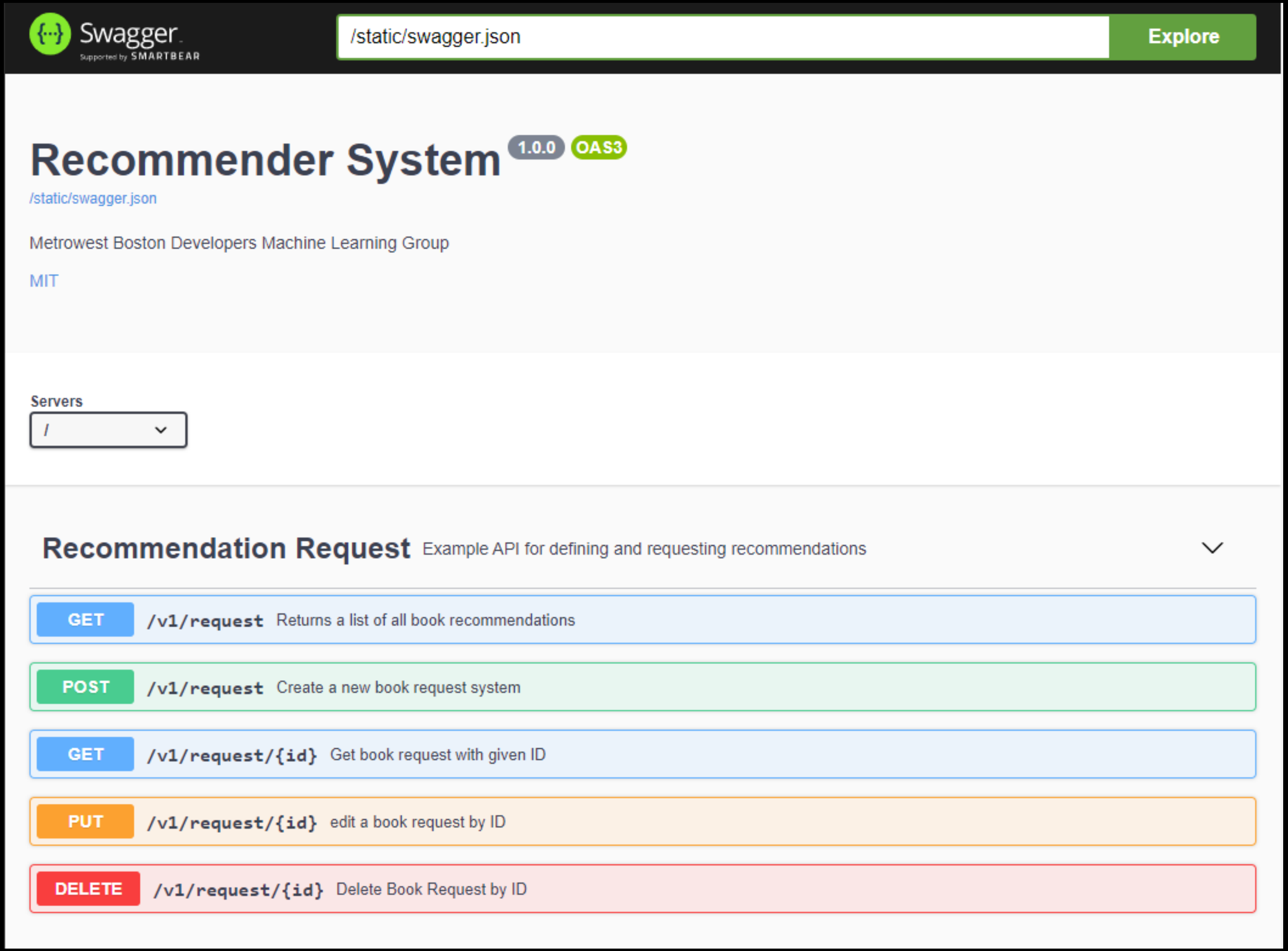
vscode-remote-try-python > requirements.txt

```
flask
flask-swagger-ui
validate_email
```



```python
### swagger specific ###
SWAGGER_URL = '/swagger'
API_URL = '/static/swagger.json'
SWAGGERUI_BLUEPRINT = get_swaggerui_blueprint(
    SWAGGER_URL,
    API_URL,
    config={
        'app_name': "Recommender-System-Python-Flask-REST"
    }
)
app.register_blueprint(SWAGGERUI_BLUEPRINT, url_prefix=SWAGGER_URL)
### end swagger specific ###
```

Swagger Interaction



/static/swagger.json

Recommender System

/static/swagger.json

Metrowest Boston Developers Machine Learning Group

MIT

Servers

/

Recommendation Request Example API for defining and requesting recommendations

GET `/v1/request` Returns a list of all book recommendations

POST `/v1/request` Create a new book request system

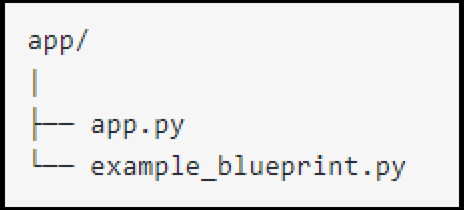
GET `/v1/request/{id}` Get book request with given ID

PUT `/v1/request/{id}` edit a book request by ID

DELETE `/v1/request/{id}` Delete Book Request by ID

Flask Blueprints and Code Structure

- As your code grows, it is not appropriate to contain everything in a single file.
- Flask offers a way to structure your code to keep it maintainable and clear to understand… enter Blueprints!
- Flask Blueprints encapsulate **functionality**, such as views, templates, and other resources.



```
app/
|
├── app.py
└── example_blueprint.py
```

- Originally, all of the code resided in app.py
- In a simple refactor, example_blueprint.py will contain the Flask Blueprint implementation. Then you modify app.py to recognize the additional code.

How Blueprints Work

- A Flask Blueprint is not actually an application. It needs to be registered in an application before you can run it. When you register a Flask Blueprint in an application, you’re actually extending the application with the contents of the Blueprint.

High Level Recommendation

- Select the recommendation engine type:
  - Content-based Filtering
  - Collaborative Filtering

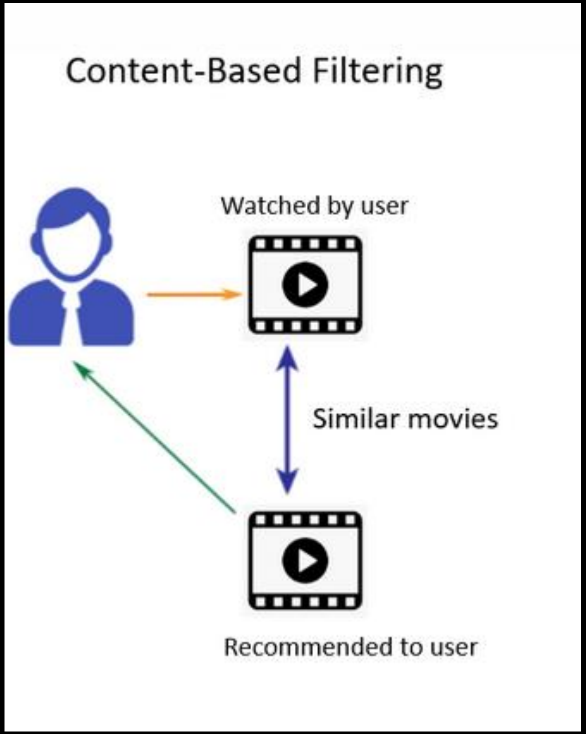
Recommendation Engine Goal

Person Profile Data + Configuration

Product Data + Configuration

Movie Recommendation

- Starting point: Content-based
- Content-Based recommendation works on the principle that if a user likes a certain item then we recommend the user a similar item based on the item's features or attributes.
- If a user likes a movie of a particular genre or an actor then we recommend a movie along similar lines to our user.

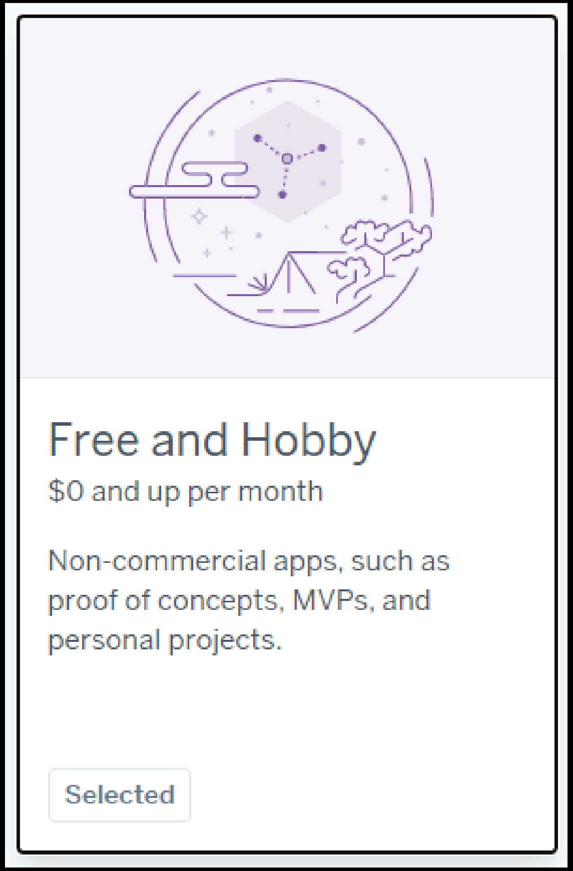


Data Sources

- A generic solution is possible when more than one data source is considered.
  - TMDB 5000 Movie Dataset | Kaggle
  - The Indian Movie Database | Kaggle

Deployment

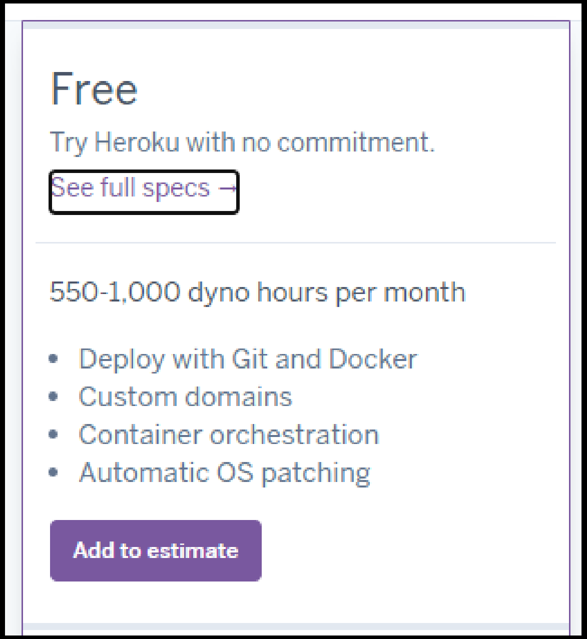
- Deploy your container to run in the cloud.



Free and Hobby

$0 and up per month

Non-commercial apps, such as proof of concepts, MVPs, and personal projects.



Free

Try Heroku with no commitment.

550-1,000 dyno hours per month

- Deploy with Git and Docker
- Custom domains
- Container orchestration
- Automatic OS patching