Cheat Sheet: Web App Deployment Using Flask

Estimated time needed: 5 minutes

Package/Method	Description	Code Example
Flask	Used to instantiate an object of the Flask class named app.	`from flask import Flask` `app = Flask(name)`

@app.route decorator	A decorator in Flask used to map URLs to specific functions in a Flask application.	`@app.route('/')`
		`def hello_world():`
		`  return "My first Flask application in action!"`

200 OK status	Flask servers automatically return a 200 OK status when you return from the @app.route method. 200 is also returned by default when you use the jsonify() method to respond to a request. A successful response with a status code of 200 will be sent back when the given code executes.	`@app.route('/')`
		`def hello_world():`
		`  return ("My first Flask application in action!", 200)`

Error 404	**400** indicates an invalid request. This status could imply the parameters are missing or improper or the request is invalid in another way.
	401 indicates the credentials are missing or invalid.
	403 implies that the client credentials are not sufficient to fulfill the request.
	404 If the server is unable to find the resource, it returns a 404 status.
	405 indicates that the requested operation is not supported.
		`@app.route('/')`
		`def search_response():`
		`  query = request.args.get("q")`
		`  if query is None:`
		`  return {"error_message": "Input parameter missing"}, 422`
		`  # fetch the resource from the database`
		`  resource = fetch_from_database(query)`
		`  if resource:`
		`  return {"message": resource}`
		`  else:`
		`  return {"error_message": "Resource not found"}, 404`

Error 500	500 is used when there is an error on the server.	`@app.errorhandler(500)`
		`def server_error(error):`
		`  return {"message": "Something went wrong on the server"}, 500`

Author(s)

Andrew Pfeiffer

Other Contributor(s)

Abhishek Gagneja, Sina Nazeri