

Cheatsheet: 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.	<pre>from flask import Flask app = Flask(__name__) @app.route('/') def hello_world(): return "My first Flask application in action!" @app.route('/') def hello_world(): return ("My first Flask application in action!", 200) @app.route('/') def search_response(): query = request.args.get("q") if not query: 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 @app.errorhandler(500) def server_error(error):</pre>
@app.route decorator	A decorator in Flask used to map URLs to specific functions in a Flask application.	
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.	
Error 404	<p>400 indicates an invalid request. This status could imply the parameters are missing or improper or the request is invalid in another way.</p> <p>401 indicates the credentials are missing or invalid.</p> <p>403 implies that the client credentials are not sufficient to fulfill the request..</p> <p>404 If the server is unable to find the resource, it returns a 404 status.</p> <p>405 indicates that the requested operation is not supported.</p>	
Error 500	500 is used when there is an error on the server.	

Package/ Method	Description	Code Example
		<pre>return {"message": "Something went wrong on the server"}, 500</pre>

Author(s)

Andrew Pfeiffer

Other Contributor(s)

Abhishek Gagneja, Sina Nazeri



skills Ne