

# Cheat Sheet: Web App Deployment Using Flask

Package/Method	Description	Code Example
<b>Flask</b>	Used to instantiate an object of the Flask class named app.	<pre>1. from flask import Flask    app = Flask(name)</pre>
<b>@app.route decorator</b>	A decorator in Flask used to map URLs to specific functions in a Flask application.	<pre>1. @app.route('/')    def hello_world():        return "My first Flask application in action!"</pre>
<b>200 OK status</b>	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.	<pre>1. @app.route('/')    def hello_world():        return ("My first Flask application in action!", 200)</pre>
<b>Error 404</b>	<p><b>400</b> indicates an invalid request. This status could imply the parameters are missing or improper or the request is invalid in another way.</p> <p><b>401</b> indicates the credentials are missing or invalid.</p> <p><b>403</b> implies that the client credentials are not sufficient to fulfill the request.</p> <p><b>404</b> If the server is unable to find the resource, it returns a 404 status.</p>	<pre>1. @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</pre>

Package/Method	Description	Code Example
	<b>405</b> indicates that the requested operation is not supported.	
<b>Error 500</b>	500 is used when there is an error on the server.	<pre> 1. @app.errorhandler(500)    def server_error(error):        return {"message": "Something went wrong          on the server"}, 500 </pre>

## Author(s)

Andrew Pfeiffer

## Other Contributor(s)

Abhishek Gagneja, Sina Nazeri