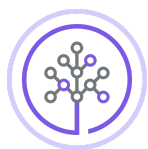


# Cheatsheet: Web App Deployment using Flask



**Skills**  
Network

**Estimated time needed:** 5 minutes

Package/Method	Description	Code Example
<b>@app decorator</b>	An app decorator is a Python tool that allows programmers to modify the behavior or function of a class. App decorators take paths as an argument.	<pre>from flask import Flask  app = Flask(__name__)  @app.route('/')</pre>
<b>@app.route decorator</b>	A decorator in Flask used to map URLs to specific functions in a Flask application.	<pre>def hello_world():      return "&lt;b&gt;My first Flask application in action! &lt;/b&gt;"</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>@app.route('/')  def hello_world():      return ("&lt;b&gt;My first Flask application in action!", 200)</pre>
<b>Error 404</b>	<b>400</b> indicates an invalid request. This status could imply the parameters are missing or improper or the	<pre>@app.route('/')  def search_response():</pre>

Package/Method	Description	Code Example
	request is invalid in another way.	<pre> 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):      return {"message": "Something went wrong on the server"}, 500 </pre>
<b>Error 500</b>	500 is used when there is an error on the server.	

## Author(s)

Andrew Pfeiffer

## Other Contributor(s)

Abhishek Gagneja, Sina Nazeri

# Changelog

Date	Version	Changed by	Change Description
2023-07-10	0.1	Andrew Pfeiffer	Initial version created

© IBM Corporation 2023. All rights reserved.