

Flask 101

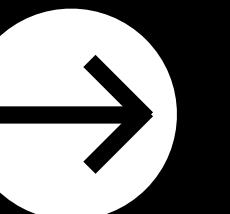
Suchandra
Thapa



Table Of Contents

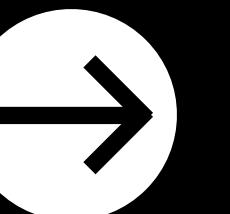
Introduction

Elaborate on what you want to discuss.



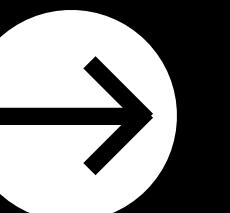
Our Projects

Elaborate on what you want to discuss.



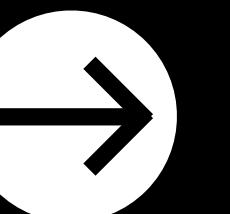
About Us

Elaborate on what you want to discuss.



Chapter Title

Elaborate on what you want to discuss.



Introduction

What is Flask?

Flask is a python module for generating web servers quickly and efficiently. Often used to generate REST api servers because it's minimalistic so that you can start create a barebones server **really** easily.

This talk will go through using Flask to create a very simple REST api server.





Your very first Flask server

Barebones Server

Import flask module and define a route for the server

Code

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route("/")  
def hello():  
    return "Hello, World!"
```

Keypoints

Very simple, basic hello world example is only 5 lines of code!



Let's complicate it

We want to differentiate from different methods

Code

```
from flask import Flask, request

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def hello():
    if request.method == "GET":
        return "Hello, World!"
    else if request.method == "POST":
        return "Posted"
```

Keypoints

Can use the methods argument in route decorate to specify which methods a function will handle

request object is available to get information about the incoming request



Return JSON response

Import flask module and define a route for the server

Code

```
from flask import Flask, request

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def hello():
    resp = {"result": "", "method": ""}
    if request.method == "GET":
        resp["result"] = 200
        resp["method"] = "GET"
        resp["mesg"] = "Hello, World!"
        return resp
    else if request.method == "POST":
        do_post_stuff()
        resp["result"] = 200
        resp["method"] = "POST"
        resp["mesg"] = "Posted!"
        return resp
```

Keypoints

If you return a dictionary from the function handling the request, Flask automatically turns it into a JSON response



Templating

Flask uses jinja for templating, this makes it easier to return html pages

Code

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/hello/")
@app.route("/hello/<name>")
def hello(name=None):
    return render_template("hello.html", name=name)
```

Keypoints

Can use <var> in route to get variables that capture part of the url

render_template function will take a template and render it with variable substitution



Templating pt.2

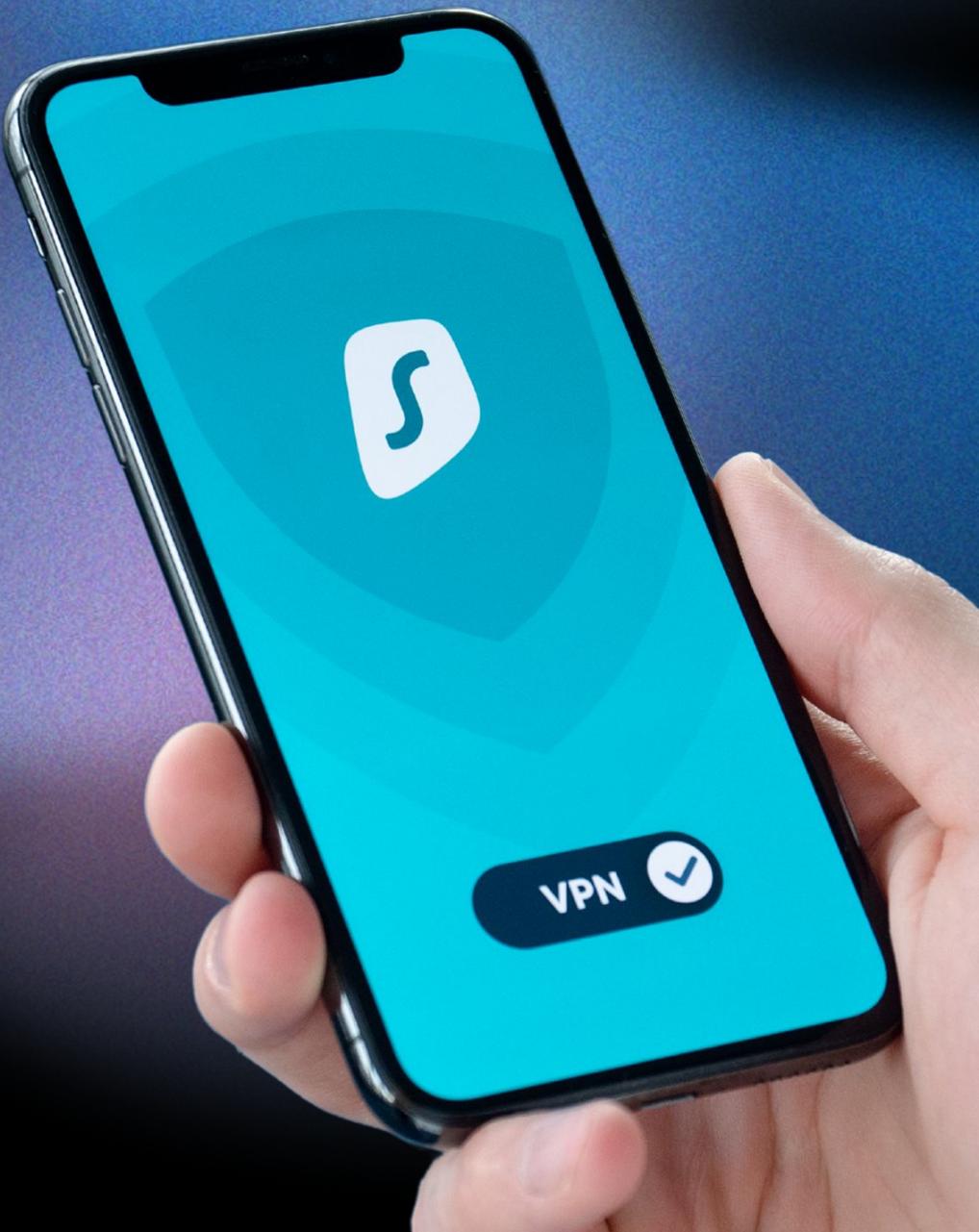
Flask uses jinja for templating, this makes it easier to return html pages

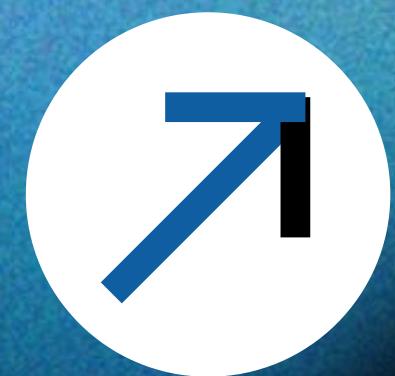
Template

```
<!doctype html>
<title>Hello from Flask</title>
{% if name %}
  <h1>Hello {{ name }}!</h1>
{% else %}
  <h1>Hello, World!</h1>
{% endif %}
```

Keypoints

Argument from render_template function is available. Can also use conditionals and a few other flow control statements





**Stop!
Demo Time**

Contact Me

suchandra@gmail.com

<https://github.com/sthana/talks/pyhawaii/flask101>

