Building Reliable Flask Applications

REVISITING THE FLASK BASICS

Abdul Rehman
MACHINE LEARNING ENGINEER

@aPythonist   www.pythonist.org

Overview

- Introduction to Flask

- Setup a basic Flask APP

  - Create Virtual Environment

  - Install Flask package

- Request-Response Cycle

- Big picture of View Functions

Revisiting the Flask Basics

Abdul Rehman

MACHINE LEARNING ENGINEER

@aPythonist   www.pythonist.org

Overview

Introduction to Flask

Create a Virtual Environment

Setup a Basic Flask application

Explore the BookLi Application

Flask

Micro WebFramework

Little Dependencies

Flexible

Jinja Templating

Quick & Easy Start

Scalable

Demo

Virtual Environment

- Setup Virtual Environment

- Install virtualenv package

Demo

Setup Flask App

– Install Flask Package

– Create a basic app

– Run a flask application

Introducing BookLi

REST Based APP

Provide web frontend
with a rest backend.

CRUD Operations

Perform CRUD on
Books database

Store favourite Books

Store your favourite
books at one place.

Let's explore the BookLi app!

Summary

Setup Environment & Flask

- Install and Create Virtual environment

- Install Flask

Setup Basic Flask App

– Create a simple Flask app

– Run flask app in different ways

Flask Routing and View Functions

What Routing means?

Define URLs

Define the url paths to access the data

Structure Paths

Implements expressions for URLs

Dynamic

Make urls/paths dynamic to scale

Define a Route in Flask

```python
from flask import Flask

app = Flask(__name__)

@app.route("/", methods=[])
```

view.py

Define a View
Function in Flask

```python
from flask import Flask

app = Flask(__name__)

@app.route("/", methods=[])
def index():
    return 'hello world'
```
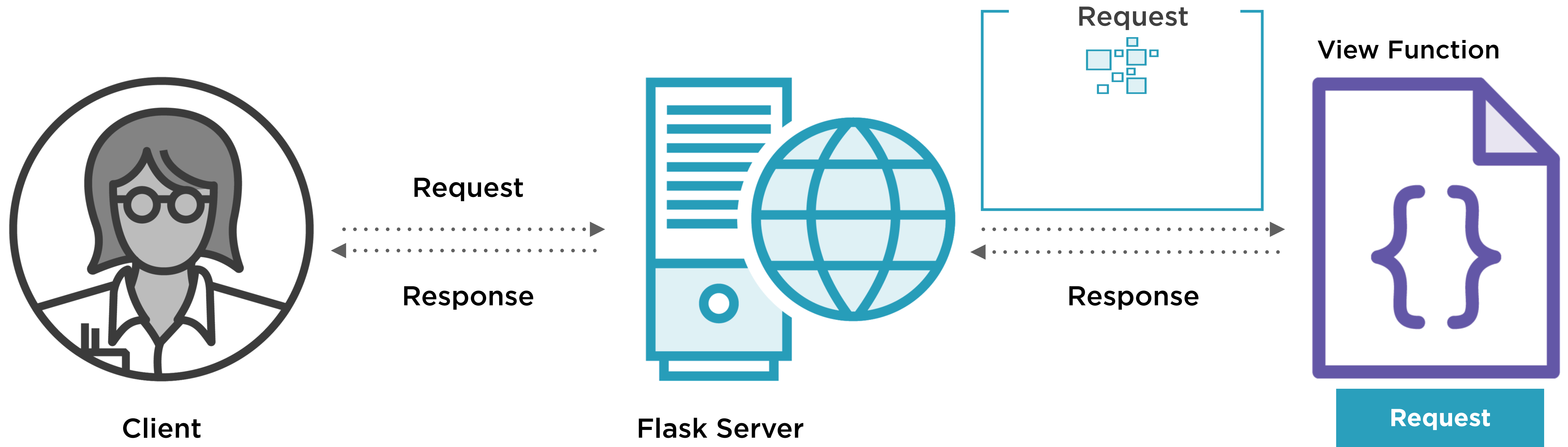
Demo

Define a Route
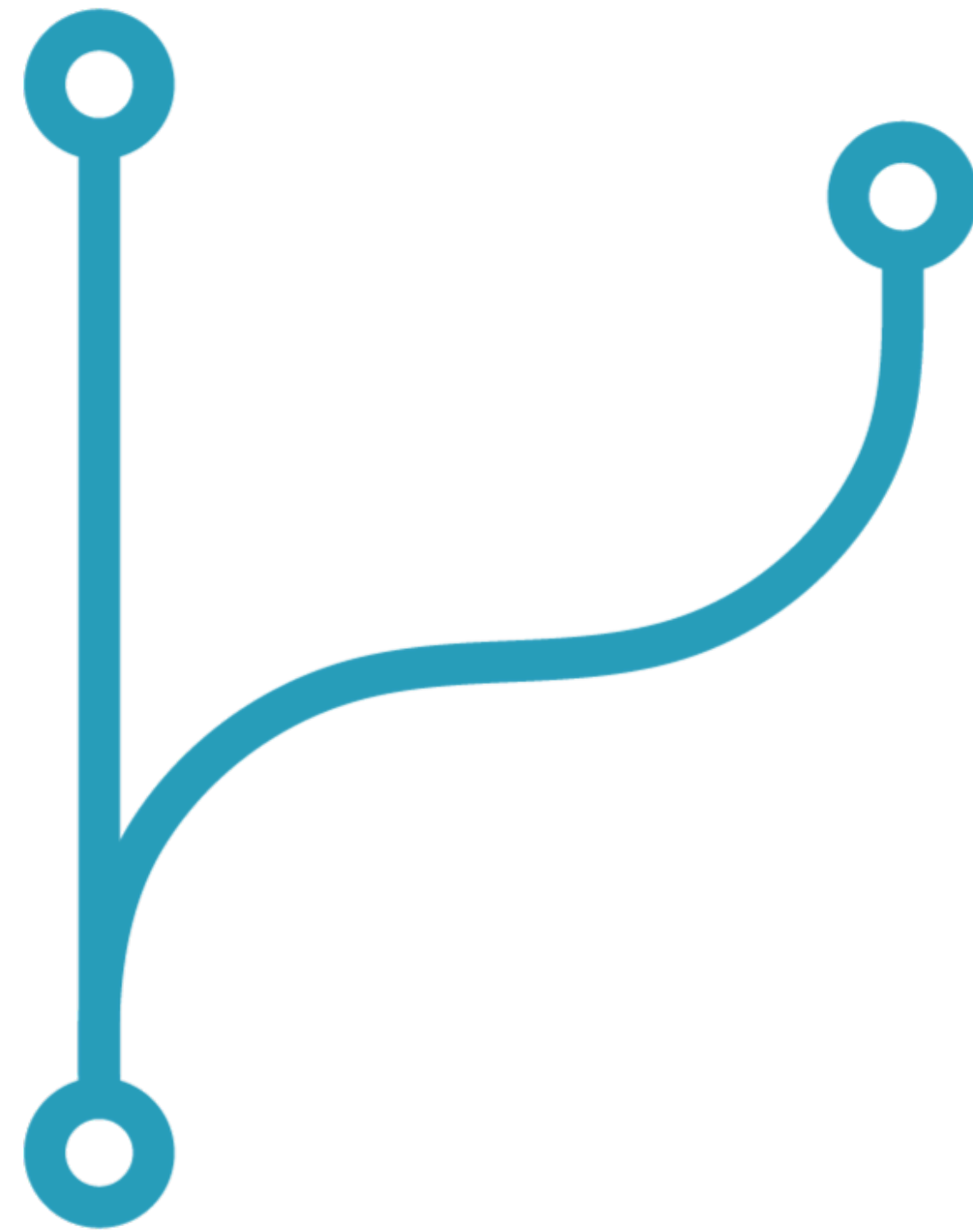
  – Define URL PATH and methods

View Function

  - Define view function

  - Explore the request

  - Return a response

Flask Request-Response Cycle

Client

Request

Response

Flask Server

Request

Response

Request

View Function

Dynamic Routes

Important for real-world applications

Carrot brackets use <>

Variable Types

- String Type

- Integer

- Float
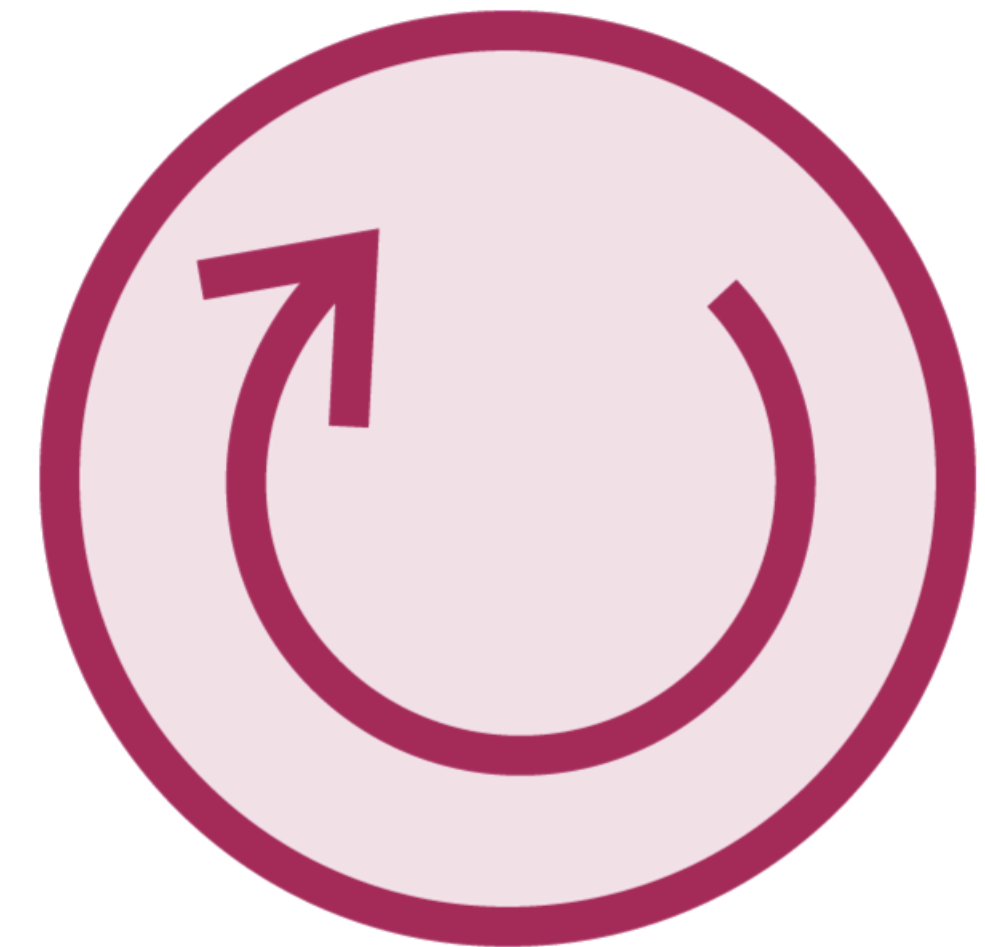
- Path

Ways to Conclude a Route

Page template

Render a template with dynamic content

Response Object

Prepare & return a Response object

Redirect

Redirect user to another page/url

Remember: views always conclude with a *return* statement.

Return Page Template & Redirect

TemplateReturn.py

```python
from flask import Flask, render_template
app = Flask(__name__)
@app.route("/")
def home():
    return render_template("index.html")
```

RedirectReturn.py

```python
@app.route("/")
def home():
    return redirect("home")
```

Making a Response Object

Utilise built-in make_response() function

Define Headers

Prepare body and add Status Code

Demo

Implement View with Response Object

- Add body

- Status Code

- Response Header

Summary

Introduction to Falsk

Setup environment & basic Flask app

Define Routes

Implement View functions

Explore BookLi app