

Web Application using Flask

1 Introduction

what is HTTP and what does it have to do with Flask?

HTTP is a protocol for websites. The internet uses it to interact and communicate with computers and servers.

For example when a user tries to access a specific website HTTP request is sent to a server. The server receives the request and figures out how to respond to request. The server sends back an HTTP response that contains the information that browser needs. Then the browser displays the response from the server.

How is Flask involved?

Flask will take care of the server side processing.

What is Flask?

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Flask depends on the Jinja template engine and the Werkzeug WSGI toolkit. It makes the process of designing a web application simpler. Flask lets us focus on what the users are requesting and what sort of response to give back.

1.1 How does a Flask app work?

The code lets us run a basic web application that we can serve, as if it were a website.

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route("/")
6 def home():
7     return "Hello, World!!!"
8
9 if __name__ == "__main__":
10     app.run(debug=True)
```

- L1: import Flask module for creating web server based on the module
- L3: specify current file. current file represent web application. app is an instance of Flask class which is a web application.
- L5: represent a default page
- L6-7: when a user visits the page, they will see the default page and the function below will be activated
- L10: run the application. debug=True allows possible python errors to appear on the web page which allows us to trace errors.
-

1.2 HTML, CSS, and virtual environments

The Flask Framework looks for HTML files in a folder called templates. We need to create a templates folder and put all our HTML files in there. The `render_template()` method from flask framework looks for a template in the templates folder.

To connect multiple pages we can use a navigation menu on the top. Flask can be used to make the process of creating a navigation menu easier.

Let's create a `template.html` which serves as a parent template. Our child templates will inherit code from it. This allows us to not have to copy the code for the navigation menu in the `about.html` and `home.html`

As we have created a folder called templates to store all HTML templates, we need a folder called static to store our CSS, JavaScript, images, and other necessary files.

2 Deploy your Web Application to the Cloud

To deploy our web application to the cloud, we will use Google App Engine (Standard Environment). This is an example of a Platform as a Service (PaaS).

PaaS refers to the delivery of operating systems and associated services over the internet without downloads or installation. The approach lets customers create and deploy applications without having to invest in the underlying infrastructure.

<https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/>