Advanced Programming with Python. Session 5

Pepe García

2020-04-20

Advanced Programming with Python. Session 5

Plan for today

- HTML forms
- handling HTML forms in flask

HTML forms

Whenever we want to gather data from the user in HTML, we'll use forms.

All fields in forms must be inside a <**form**> tag

```
<form
  action="http://localhost:5000/form"
  method="POST">
 ...
</form>
```

We'll put the URL for handling the form in **action**

And the HTTP method in the **method** attribute

HTML forms. Fields

Well use the **input** tag for handling different kinds of inputs from the user.

We'll always need to give a unique **name** to it and a **type**

```
<input name="user" type="text"/>
```

HTML forms. Fields

There are a lot of types of inputs we can use.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

```
<input name="pass" type="password"/>
<input name="date" type="datetime-local"/>
...
```

HTML forms. Submit

In order to create a button that submits the **form**, we'll use

<input type="submit"/>

```
<input type="submit" value="submit!"/>
```

Practice

Create a simple login form in HTML. It should contain a user field, a password field, and a submit button.

session-5/exercise-1

Handling HTML forms in flask

When receiving **form** data in flask, we can access to it using the **request** object.

```
from flask import request


@app.route("/handle", methods = ["POST"])
def handle_form_submission():
    user = request.form["user"]
    password = request.form["pass"]

    if user in users and users[user] == password:
        return "logged in!"
    else:
        return "not logged in..."
```

the keys in the **form** dictionary are the values we put in the **name** attribute of the **input**

Practice

Let's see an example of HTML form being handled in flask.

session-5/example-1

Homework

Create a new server that will behave like twitter. We should be able to send new tweets, and they should be displayed in the website