Creating Forms using Flask-WTF and WTForms

In this lesson, we will learn how to create forms using the Flask-WTF and WTForms modules.

In the last lesson, we created a simple application containing a login form and validation.

> **Note:** Our application did not contain any checks on the `email` and `password` fields because it was a simple example. If we wanted to add these checks, we would have to write the logic for that at the front-end or backend (inside the `login` view).

In larger applications, these kinds of extra components can easily become boiler-plate and hard to read. For this purpose, some libraries this process easier.

Introduction to `WTForms`

`WTForms` is a library that makes form handling easy. It handles not only form validation but also form rendering at the front-end. Additionally,

> WTForms is not just limited to Flask.

> ### Introduction to `Flask-WTF`
>
> `Flask-WTF` is a `Flask` specific library that integrates the `WTForm` library with `Flask`. It acts as an add-on to `WTForms` and adds some extra components, such as security.

In this lesson, we will be using `Flask-WTF` in conjunction with `WTForms` to handle forms. Let's get started.

How do we create the forms module?

To get started with `Flask-WTF`, we will first separate our application module from the forms module. Let's add a new file called `forms.py`, which will act as the forms module.

Import `FlaskForm` from `flask_wtf`

First, we will import the `FlaskForm` class from the `flask_wtf` module. This class is a subclass of `Form` from the `wtforms` library.

```
from flask_wtf import FlaskForm
```

Create `LoginForm` class

For each form on our website, we will create a class. As we are making a login form. Therefore, let's name this class `LoginForm`. This class will inherit from the `FlaskForm` class that we imported previously.

```
class LoginForm(FlaskForm):
    ...
```

Add form fields from `wtforms`

The login form that we created in the last lesson had three components:

1. An input field for the email.
2. An input field for the password.
3. The submit button field.

For each possible field, `wtforms` has associated classes. For this particular example we will only import the fields we need:

1. `StringField` for an email
2. `PasswordField` for a password
3. `SubmitField` for the submit button

Let’s import these classes.

```
from wtforms import StringField, PasswordField, SubmitField
```

Now we will make instances of these classes as member variables of our class, and we will pass the labels of these fields as input to the constructors.

```
class LoginForm(FlaskForm):
    email = StringField('Email')
    password = PasswordField('Password')
    submit = SubmitField('Login')
```

Add field validators from `wtforms`

Validators are the rules and checks that we want to apply to a field inside the form. For example, with an email field, we want to make sure that the input is a valid email address. For this purpose, we will use the `Email` validator. A complete list of built in validators can be found at WTForms documentation. In this example, we are going to use only two of them: `Email` and `RequiredField`. So, let’s import them.

```
from wtforms.validators import InputRequired, Email
```

Now, to apply these validators to the fields, we provide a list of validators as parameters to them like so:

```
class LoginForm(FlaskForm):
    email = StringField('Email', validators=[InputRequired(), Email()])
    password = PasswordField('Password', validators=[InputRequired()])
    submit = SubmitField('Login')
```

- **`InputRequired()`** : Sets the `required` attribute in the `HTML`.

- `Email()` : Checks if the given input is a valid email.

> **Note:** If the input is not valid, according to the set validators, then a `ValidationError` is raised. We can find the errors raised by a field by accessing the `field_name.errors` dictionary. More on `errors` later in this chapter.

Complete implementation of `forms.py`

The complete implementation of the forms module, `forms.py`, can be found below:

forms.py

```python
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import InputRequired, Email

class LoginForm(FlaskForm):
    email = StringField('Email', validators=[InputRequired(), Email()])
    password = PasswordField('Password', validators=[InputRequired()])
    submit = SubmitField('Login')
```

In the next lesson, we will learn how to render this login form in a template.