N03a_URL_Buildings_and_Redirects

In this chapter, students will learn and understand:

- Use the Jinja2 template language and flask.url_for() to dynamically generate links and URLs in HTML responses
- Use flask.redirect() to send a HTTP status code and have the browser request a different address

Section 1: Using url_for() in Python

`flask.url_for(endpoint, values)`
Generates a URL to the given endpoint with the method provided. The `endpoint` here refers to a function. `values` are the optional arguments passed to the function.

1. Create two new HTML files and save them as **student.html** and **teacher.html** in the templates subdirectory.

In each of the webpage, add some information about the students and teachers of the class.

2. In **app.py**, import `url_for`.

3. Add the following three decorators and functions into **app.py**.

```
@app.route('/student/')
def student():
    return render_template('student.html')


@app.route('/teacher/')
def teacher():
    return render_template('teacher.html')


@app.route('/display/<role>')
def display(role):
    if role == 'student':
        return "URL for student is: " + url_for('student')
    elif role == 'teacher':
        return "URL for teacher is: " + url_for('teacher')
    else:
        return 'Student and teacher page not chosen.'
```

4. Run **app.py**.

5. Browse on the browser to test out our code.

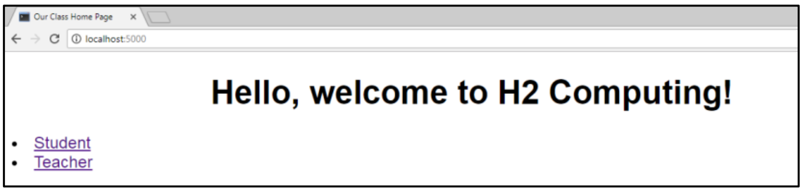
Section 2: Using `url_for()` In Templates

Now we will be creating hyperlinks from **index.html** to **student.html** and **teacher.html**.

1. Open **index.html**.

2. Add 2 hyperlinks using `url_for` such that they will appear as shown below. Take special note of the double quotes, single quotes and brackets.

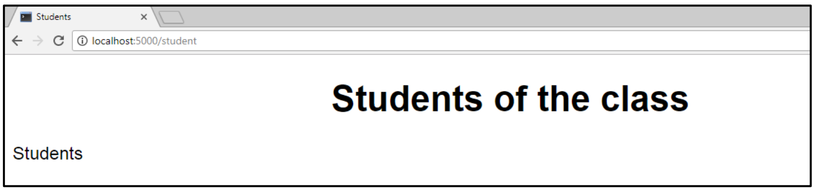
```
<p>
    <a href="{{ url_for('student') }}">Student</a>
</p>
<p>
    <a href="{{ url_for('teacher') }}">Teacher</a>
</p>
```



Hello, welcome to H2 Computing!

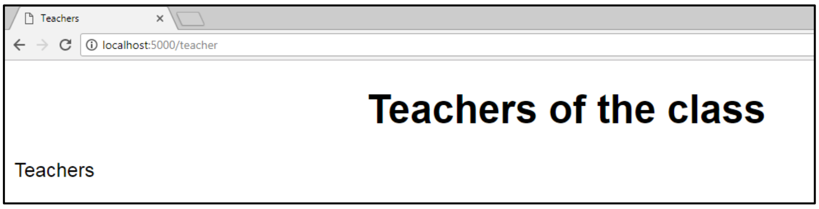
- Student
- Teacher

3. Click on each of the link. The respective webpages should appear.



Students of the class

Students



Teachers of the class

Teachers

Basically, `url_for` will contain the python function name we are looking for, and it will generate and return the routing URL for this function.

In addition, we can also use `url_for` in generating file path in static folder by using the following code in html:

```
<head>
    <link rel="stylesheet" type="text/css"
        href="{{ url_for('static', filename='style.css') }}">
    <title>Home</title>
</head>
```

Section 3: Using `redirect` in Templates

Flask class has a `redirect()` function. When called, it returns a response object and redirects the user to another target location with specified status code.

`flask.redirect(location, statuscode, response)`
`location` parameter is the URL where response should be redirected. `statuscode` sent to browser's header, defaults to `302`. `response` parameter is used to instantiate response

1. In **app.py**, import `redirect`.

2. Now modify the display function such that a text response is returned using redirect instead of the URL.

3.

```
@app.route('/display/<role>')
def display(role):
    if role == 'student':
        return redirect(url_for('student'))
    elif role == 'teacher':
        return redirect(url_for('teacher'))
    else:
        return 'Student and teacher page not chosen.'
```

4. Test both URLs:

```
http://localhost:5000/display/student
http://localhost:5000/display/teacher
```

Notice that when the user enters `http://localhost:5000/display/teacher` into the address bar of the browser, the user is being redirected to `localhost:5000/teacher` and the relevant text is displayed on the browser.

Terminologies:

- **url_for():** Generates a URL to the given `endpoint` with the method provided, where `endpoint` is a function.
- **redirect():** Returns a response object and redirects the user to another target location with specified status code.

References:

- MOE Python Flask Web Application Starter Kit
- MOE Teacher Training 2018 - HTML and CSS
- Udemy: Python and Flask Bootcamp: Create Websites using Flask!
- Udemy: Build Responsive Real World Websites with HTML5 and CSS3