Flask URL Building

The url_for() function is used to build a URL to the specific function dynamically. The first argument is the name of the specified function, and then we can pass any number of keyword argument corresponding to the variable part of the URL.

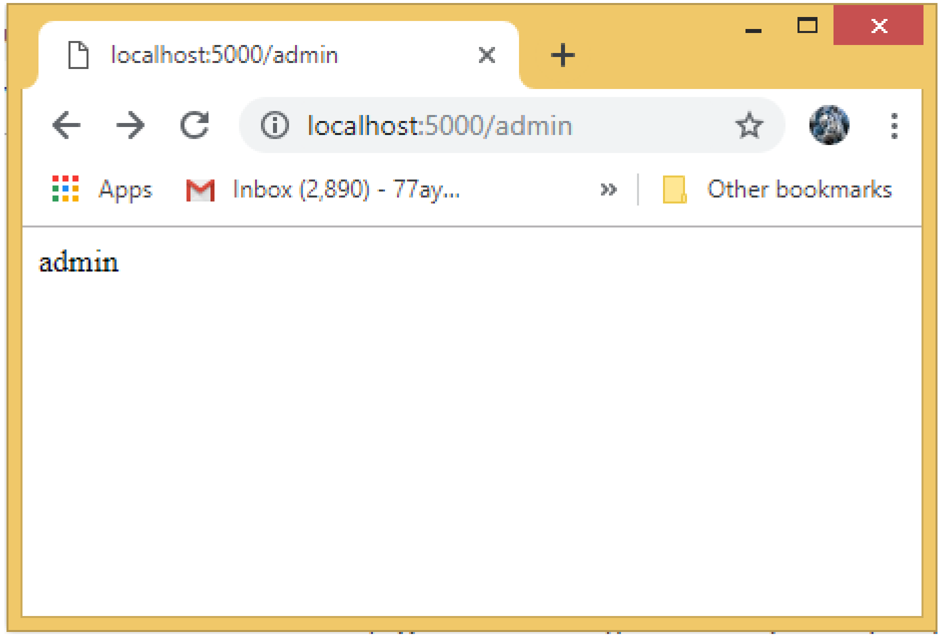
This function is useful in the sense that we can avoid hard-coding the URLs into the templates by dynamically building them using this function.

Consider the following python flask script.

Example

```
from flask import *

app = Flask(__name__)

@app.route('/admin')
def admin():
    return 'admin'

@app.route('/librarion')
def librarion():
    return 'librarion'

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

@app.route('/user/<name>')
def user(name):
    if name == 'admin':
        return redirect(url_for('admin'))
    if name == 'librarion':
        return redirect(url_for('librarion'))
    if name == 'student':
        return redirect(url_for('student'))
if __name__ =='__main__':
    app.run(debug = True)
```

The above script simulates the library management system which can be used by the three types of users, i.e., admin, librarion, and student. There is a specific function named user() which recognizes the user the redirect the user to the exact function which contains the implementation for this particular function.



admin

For example, the URL http://localhost:5000/user/admin is redirected to the URL http://localhost:5000/admin, the URL localhost:5000/user/librarion, is redirected to the URL http://localhost:5000/librarion, the URL http://localhost:5000/user/student is redirected to the URL http://localhost/student.

Benefits of the Dynamic URL Building

1. It avoids hard coding of the URLs.
2. We can change the URLs dynamically instead of remembering the manually changed hard-coded URLs.

3. URL building handles the escaping of special characters and Unicode data transparently.
4. The generated paths are always absolute, avoiding unexpected behavior of relative paths in browsers.
5. If your application is placed outside the URL root, for example, in /myapplication instead of /, url_for() properly handles that for you.

← Prev

Next →

Feedback

- Send your Feedback to [email protected]

Help Others, Please Share