LECTURE 37

FORMS

MCS 275 Spring 2021

Emily Dumas

LECTURE 37: FORMS

Course bulletins:

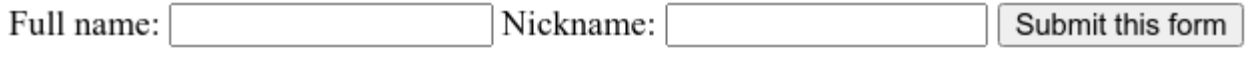
- Project 4 description posted. Make a Flask+SQLite app. Very flexible rules (e.g. can give and receive help, use online resources, base it on Yellaro/Whinge or not).
- Project 4 is due 6pm CDT Friday April 30.

CHAT APP TODO

- ✓ HTML mockup
- ✓ Stylesheet
- ✓ (Learn a bit about Flask)
- ✓ Database schema & test data
- ✓ Python code to generate message view HTML from a database query
- Add form to post a message to HTML
- Create route for new message submission

FORMS

Interactive elements in an HTML document (text entry, checkbox, dropdown list, etc.)




```
<form action="https://example.com/formsub/">
    <label for="full">Full name:</label>
    <input type="text" id="full" name="full">
    <label for="nick">Nickname:</label>
    <input type="text" id="nick" name="nick">
    <input type="submit" value="Submit this form">
</form>
```

jsfiddle is a nice way to test out form designs (for code that can be public).

By default, forms use a `GET` request to send their data as name=value pairs at the end of the URL. These are **query parameters**, e.g.

```
https://example.com/formsub/?full=David%20Dumas&nick=deedee
```

Many ascii characters appear verbatim but others* become `%` escape sequences with two hex digits. Flask decodes these and makes the parameters available as `flask.request.values.get(name)`.

* The precise encoding scheme is specified in RFC3986. Python's built-in `urllib.parse` module has functions that perform this type of encoding/decoding:

`urllib.parse.quote` and `urllib.parse.unquote`. When using Flask, you usually won't call these directly.

POST

It is better to have forms submit their data with a HTTP POST request, so form data is never part of the URL.*

Easy change: Add `method="post"` attribute to the `<form>` tag.

* URLs are sometimes not treated as sensitive user data, while request contents typically are. Thus putting form data in a URL, as GET does, carries an unintended disclosure risk.

A Flask route needs to indicate if it accepts POST requests, since the default is to only allow GET. If it does, form values are available in the same way as with GET, i.e. `flask.request.values.get(name)`.

```
from flask import Flask, request

# ... app setup ...

@app.route('/registernick',methods = ['POST', 'GET'])
def record_fullname_and_nickname():
    print("Received nickname {}".format(
        request.values.get("nick")
    ))
```

FLASK FUNCTIONS

- **`url_for(func_name,param1=val1,param2=val2,...)`** - Get URL corresponding to a function within this application, with optional query parameters, e.g. `url_for("record_score",postid=5,score=11)` might return `"/setscore?postid=5&score=11"` if your app contains:

```
@app.route("/setscore")
def record_score():
    print("recording score {} for postid {}".format(
        flask.request.values.get("score"),
        flask.request.values.get("postid"),
    )
```

FLASK FUNCTIONS

- **`redirect(url)`** - Returning this object from a route will cause the HTTP server to issue a 302 response, redirecting the client to `url`. (Basically, it means "ask them to load a different URL")
- **`abort(http_error_code)`** - Immediately stop and return a HTTP error code (usually 400 bad request, 401 not authorized, 403 forbidden, or 404 not found).

CHAT APP ROUTES

- `/` - (GET) show message feed
- `/post` - (POST) add message

VOTE APP ROUTES

- **`/top/`** - (GET) show items, ranked
- **`/new/`** - (GET) show items, chrono
- **`/post`** - (POST) submit item
- **`/plus?postid=15`** - (GET) add one to score
- **`/minus?postid=15`** - (GET) subtract one from score

REFERENCES

- jsfiddle - Write and test HTML+CSS quickly in browser
- HTML tutorial from w3schools
- CSS tutorial from w3schools
- The Flask tutorial

REVISION HISTORY

- 2021-04-15 Fixes to `url_for` and link to form input element docs
- 2021-04-13 Initial publication