Flask Tools

Download Demo Code <../flask-tools-demo.zip>

Goals

- Explore other common features of web frameworks like Flask, including:
  - Redirecting
  - Flash messaging
  - Returning JSON data
- Debug Flask errors from the error page
- Set break points in Python code with ***pdb***

Redirecting

What is an HTTP redirect?

- An HTTP response
- The status code is a “redirect code” (often, 302)
- It contains a URL for browser to re-request
- Typically, for ancient browsers, contains HTML with a link

```
$ curl -v http://localhost:5000/redirect-me

< HTTP/1.0 302 FOUND
< Content-Type: text/html; charset=utf-8
< Location: http://localhost:5002/somewhere-else <http://localhost:5002/somewhere-else>

<h1>Redirecting...</h1>
<p>You should be redirected automatically to target URL:
 <a href="/somewhere-else">/somewhere-else</a>.
 If not click the link.</p>
```

Your browser won’t typically show you this page — it makes the re-request so fast you don’t even notice it happened!

Flask Debug Toolbar & Redirects

The Debug Toolbar makes redirects explicit

This is very useful for debugging!

If you don’t want this, you can turn it off:

```
app.config['DEBUG_TB_INTERCEPT_REDIRECTS'] = False
```

Common Pattern: Redirect POST to GET

- POST requests are often from a form
  - And change data on the server
- If you return HTML from a POST request, the browser shows it fine
  - But if the user hits “Refresh”, they get weird “ok to resubmit” dialog
- Better strategy:
  - Do the work you want inside your POST route
  - Then *redirect to* a page that shows the confirmation

demo/app.py

```
@app.route("/post-example", methods=["POST"])
def post_example():
    """An example of good POST handling."""

    isbn = request.form["isbn"]

    print(f"\n\nBuying Book: {isbn}\n\n")

    # flash message: we'll talk about this soon
    # flash(f"Book {isbn} bought!")

    return redirect("/thanks")


@app.route("/thanks")
def say_thanks():
    """Thank user for buying a book."""

    return render_template("thanks.html")
```

Message Flashing

Often you want to provide feedback at “the next page user sees”

This is most common when you will redirect

```
from flask import flash

@app.route("/your/route")
def your_route():
  """Some route that redirects."""
```

```
  flash("Message for user!")
  return redirect("/somewhere/else")
```

template used by /somewhere/else

```
{% for msg in get_flashed_messages() %}
  <p>{{ msg }}</p>
{% endfor %}
```

Returning JSON

JSON is just a string — so you don't *need* to do anything special

```
@app.route("/some/route")
def some_route():
    """Route that returns JSON."""

    return '{"name": "Whiskey", "cute": "hella"}'
```

Two considerations:

- It's finicky to hand-write JSON and get it right
- It's sometimes helpful to send header to browser that "this is JSON"
  - Some AJAX plugins are better than others in guessing in absence of this

demo/app.py

```
@app.route("/example-json")
def example_json_route():
    """Return some JSON."""

    info = {"name": "Whiskey", "cute": "Hella"}
    return jsonify(info)

    # Alternate syntax
    # return jsonify(name="Whiskey", cute="Hella")
```

Flask Debugging

Strategies:

- as always ***print()*** *(appears in terminal)*
- Flask Debug Toolbar
- Get an error? You can debug on the error page!

Debugging Errors

Click the black “Terminal” symbol in a traceback

You’ll need to enter “PIN code” (printed out to terminal at start)

That will give you a Python interpreter right there!

You can examine variables, try out code, etc.

Python Debugger

Python includes a built-in debugger, ***pdb***

To add a breakpoint to your code:

```
def my_function():
    ...

    import pdb
    pdb.set_trace()

    ...
```

When you hit that ***set_trace()***, Python will stop so you can debug this

Debugger Basics

Key	Command
?	Get help
l	List code where I am
p	Print this expression
pp	Pretty print this expression
n	Go to next line (step over)
s	Step into function call
r	Return from function call
c	Continue to next breakpoint
w	Print “frame” (where am I?)
q	Quit debugger