

HTTP:

metode: **GET**, **PUT**, **POST**, **DELETE**, **HEAD**

coduri de raspuns: 200, 404

2xx - succes

3xx - redirectare

4xx - erori de continut

5xx - erori de server

Flask:

```
virtualenv env
source env/bin/activate
pip install flask
pip install flask-sqlalchemy
```

What do we have so far:

```
todo.py
templates/home.html
templates/add.html
```

NOTA BENE: to use a common frame, define:

```
templates/layout.html
```

with the content:

```
<html>
  <head><title>Twitter Clone</title></head>
  <body>
    {% block content %}
      Here comes content, overridden by other templates.
    {% endblock %}
  </body>
</html>
```

then modify each template accordingly:

```
{% extends 'layout.html' %}
{% block content %}
... current content ...
{% endblock %}
```

Hands on: Twitter Clone

- log in
- log out
- home page: all messages with users, dates, texts
 - ordered by timestamp
- user's page: messages

- ordered by timestamp
- click on user on home page -> go to user's page

Today's TODO List App:

```
from flask import Flask
from flask import render_template, request, url_for, redirect,
session, g
from flask.ext.sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['DEBUG'] = True
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///tmp/data.db'
app.config['SECRET_KEY'] = 'something secret'

db = SQLAlchemy(app)

@app.route('/')
def homepage():
    if not session.get('username', None):
        return redirect(url_for('loginview'))
    return render_template('home.html',
                           aplicatie='OSSS', tasks=Task.query.all())

@app.before_request
def setup():
    g.username = session.get('username', None)

@app.route('/login', methods=('GET', 'POST'))
def loginview():
    if request.method == 'POST':
        session['username'] = request.form['username']
        return redirect(url_for('homepage'))
    return render_template('login.html')

@app.route('/logout')
def logoutview():
    session['username'] = None
    return redirect(url_for('loginview'))

@app.route('/add', methods=('GET', 'POST'))
def add():
```

```
    if request.method == 'POST':
        new_task = Task(text=request.form['taskname'])
        db.session.add(new_task)
        db.session.commit()
    return render_template('add.html')

@app.route('/setdone/<id>')
def setdone(id):
    task = Task.query.filter_by(id=id).first()
    task.done = True
    db.session.commit()
    return redirect(url_for('homepage'))

@app.route('/delete/<id>')
def delete(id):
    task = Task.query.filter_by(id=id).first()
    db.session.delete(task)
    db.session.commit()
    return redirect(url_for('homepage'))

class Task(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    text = db.Column(db.String)
    done = db.Column(db.Boolean, default=False)

db.create_all()
app.run()
```