

# Advanced Database Design

## Web App Basics

Gregory S. DeLozier, Ph.D.

[gdelozie@kent.edu](mailto:gdelozie@kent.edu)

# Web Concepts

## —Web server

- Conversation in HTTP
- Responds to requests for content
- GET request
  - Additional info (queries) in URL
- POST request
  - Additional info in body content
- Web server returns whatever it wants to

# URLs

Parts of URL

<protocol><address><route><query>

# URLs

## Parts of URL

<protocol><address><route><query>

http://

ftp://

https://

# URLs

## Parts of URL

<protocol><address><route><query>

[www.google.com](http://www.google.com)

121.32.45.111

# URLs

## Parts of URL

<protocol><address><route><query>

[www.google.com](http://www.google.com)

121.32.45.111

127.0.0.1

localhost

# URLs

## Parts of URL

<protocol><address:port><route><query>

127.0.0.1:80

Localhost:3000

Normal HTTP: 80

Normal HTTPS: 443

# URLs

## Parts of URL

<protocol><address><route><query>

/

/index

/hello

/hello/whatever

/hello/name      <- more on this later



# URLs

## Parts of URL

<protocol><address><route><query>

?name=greg&county=medina  
?id=1232451

# URLs

## Parts of URL

<protocol><address><route><query>

?name=greg&county=medina  
?id=1232451

--or, if you have a route manager—

/person/greg/medina  
/case/1232451

# URLs

## Parts of URL

<protocol><address><route><query>

<http://www.mysite.com/person?name=greg>

https://learn.kent.edu/webapps/blackboard/execute/  
content/file?

cmd=view&mode=designer&content\_id=\_3682168\_1&  
course\_id=\_106379\_1

# Static Web Server

```
python -m SimpleHTTPServer 8000
```

Index.txt

Hello, world.

# Common Gateway Interface (CGI)

```
python -m CGIHTTPServer 8000  
cgi-bin/hello.py
```

```
#!/usr/bin/env python  
print """Content-Type: text/text
```

```
hello  
"""
```

# Web Frameworks

Programs to serve web pages

- Route manager

- Allows assigning functions to URLs

- Template framework

- Allows mixing code and content to create pages

- Requests, sessions, cookies

- Allows continuity between pages

# Bottle.py

My favorite framework...

<http://bottlepy.org>

Easy to create simple things

Hard things are definitely possible

# Simple Bottle Example

```
from bottle import route, run
```

```
@route('/hello')                # define the route
```

```
def hello():                    # create response
```

```
    return "Hello, world!"
```

```
run(host='localhost', port=8080) # run server
```



# Another Bottle Example

```
# in this case a “template” allows mixing of  
#  html presentation information  
#  python variable contents
```

```
from bottle import route, run, template
```

```
@route('/hello/<name>')
```

```
def hello():
```

```
    return template('<b>Hello {{name}}</b>!', name=name)
```

```
run(host='localhost', port=8080)
```

# Database Bottle Example

```
from bottle import route, run, template
import sqlite3
```

```
connection = sqlite3.connect("pets.db")
```

```
@route('/animals')
def animals():
    c = connection.cursor()
    c.execute("select * from animals")
    s = ""
    for x in c.fetchall():
        s = s + str(x)
    return s
run(host='localhost', port=8080)
```

# A Complete Web App (\*)

A “todo” list manager

- Add items to the list

- View current items

- Check off items when done

- Delete items

(\*) Sort of. There’s no UI here to speak of.

# Tutorial Time

[http://bottlepy.org/docs/dev/tutorial\\_app.html](http://bottlepy.org/docs/dev/tutorial_app.html)

# Things to note

- Use GET for requests
  - Create a URL in your code
  - Request the URL with arguments in URL parameters
  - Get the values from request or from URL route
- Use POST for submissions
  - Wrap a post form
  - Create a submit button
  - Get data from the request body
- (These rules change later for RESTful APIs)

# Git & GitHub

## Basic Source Control

% git init	Create repository
% git clone	Copy repository
% git add --all	“Stage” files and directories
% git commit -m “My message”	Commit staged content to local
% git push origin master	Send committed content to origin

# Git & GitHub

Demo Time...

[www.github.com](http://www.github.com)

[try.github.com](http://try.github.com)

[https://github.com/gregdelozier/kent\\_adb\\_class.git](https://github.com/gregdelozier/kent_adb_class.git)