

# Scripting & Computer Environments

## *Web2py: an Introduction*

IIT-H

Oct 30, 2013

# ...Previously & Today...

## Previously: Core Python objects

- Sequences (Strings, Lists, Tuples)
- Modules
- Mappings (Dictionaries)
- Classes
- Functions
- Files

## Today:

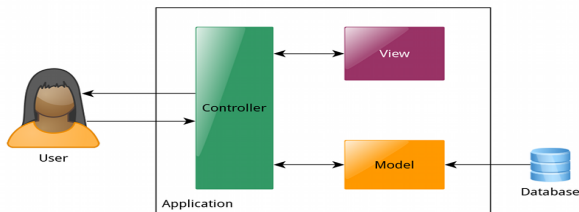
- Web Application Frameworks:

**WEB2PY**

- Software packages that facilitate the creation of dynamic web sites, applications and services.
- For quick and easy development of new web apps. How?
- Provide tools and APIs that simplify the task.
- Flavors:
  - ① **Glued** Framework - by *assembling* together 3rd party components.
  - ② **Full-Stack** Framework - by *creating* components that work together.

e.g. **web2py** (Python) ✓ , Django (Python), Drupal (PHP),  
Ruby on Rails (Ruby), etc

- Many web frameworks are based on the *model-view-controller* (MVC) architecture.
- The user application is divided into 3 components.

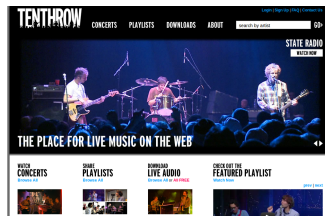
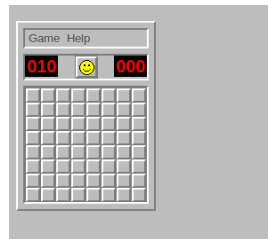
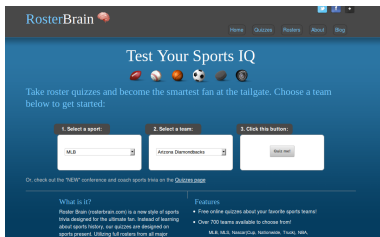


Typically, a client sends a request, which is received by the web server and passed on to a *controller*. The controller decides what to do (action). Usually it talks to the *model*, reads and writes to the database. Then, the controller takes the information from the model and generates a *view* and sends it back to the browser.

- Written by Massimo Di Pierrro.
- Open-source, full-stack web application framework.
- Follows the MVC pattern.
- Built-in components for:
  - Easy access to HTTP requests and responses.
  - Managing authentication, login
  - Managing cookies and sessions.
  - Database handling with its Database Abstraction Layer (DAL).
  - Security-related tasks (input validation, sanitization ...)

And many more...

- Lower learning Curve
- A “zero installation/configuration” framework
- Runs everywhere  
Windows, Linux/unix, Mac, Google App Engine, CPython, Jython ...
- Security
- Preloaded with many extensions and apps
- Easy packaging and deployment.



Visit [here](#) for more.

- Web2py comes in two flavors:
  - ① *Binary version* - (Windows + Mac) - Python interpreter + SQLite
  - ② *Source code version* - assumes Python is preinstalled
- Get it from the official site [here](#).
- No installation/configuration/dependencies
- Just unzip and click!



# The Startup Interface

- The splash screen
- The GUI-widget → One-time admin password
  - No password  $\Rightarrow$  No admin interface.
  - Server IP and port numbers
  - Default 127.0.0.1:8000
- Default web browser fired off.

`http://127.0.0.1:8000/welcome/default/index`

# The Web-Based Admin Interface

Web-based Integrated Development Environment (IDE) for building web2py applications.

- Login → Takes you to the “site” page.
- Create and Design Apps
- Manage Apps
- Packaging and Deployment
- Test and Debug Apps
- Live notification (updates) and Announcements (Twitter)

- The `site` page displays 3 default applications:

- 1 The `admin` App

- 2 The `examples` App

- Replica of the web2py official website.

- 3 The `welcome` App (a.k.a. the scaffolding App)

- This is the app that welcomes a user at startup.
- Also, the basic template for other apps

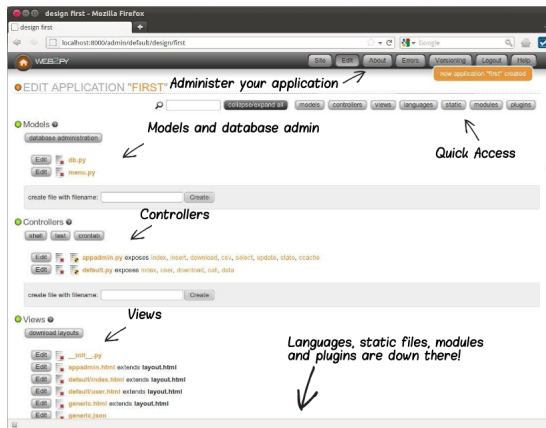
# Operations on an App

- Install
- Uninstall
- Create → By default, a clone of the `welcome` app.
- Package → for distribution
- Cleanup → temp files (e.g. sessions, errors, cache files ...)
- Edit

# Web2py App Components

Each web2py app has the following files:

{Models, Controllers, Views, Languages, Modules, Static Files, Plugins}



- **Models** - describe the *data representation* of your app

`db.py` - connects to DB, defines tables, Auth, etc

`menu.py` - defines menus for the scaffolding app

- **Models** - describe the *data representation* of your app

`db.py` - connects to DB, defines tables, Auth, etc

`menu.py` - defines menus for the scaffolding app

- **Controllers** - describe the logic & workflow of your app

`default.py` - is the default controller of your app

`appadmin.py` - defines a database administrative interface for your app (i.e. `appadmin`).

- **Models** - describe the *data representation* of your app

`db.py` - connects to DB, defines tables, Auth, etc

`menu.py` - defines menus for the scaffolding app

- **Controllers** - describe the logic & workflow of your app

`default.py` - is the default controller of your app

`appadmin.py` - defines a database administrative interface for your app (i.e. `appadmin`).

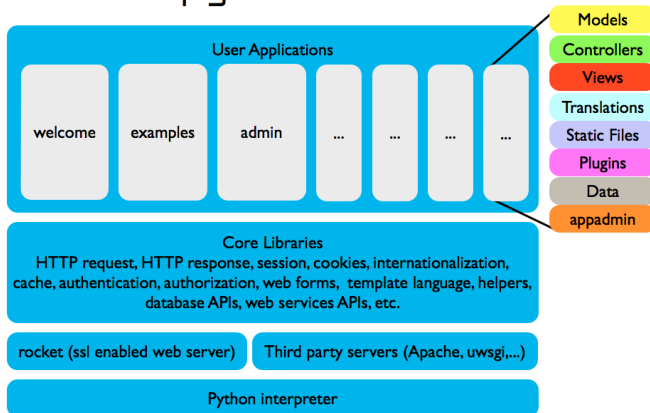
- **Views** - describe how data should be presented to the user.

- Default view file extension is `html` (others: `json`, `xml`, `rss` ...)
- A function in controller returns a dictionary that is interpreted by a view file.
- A function can have multiple views with different extensions.



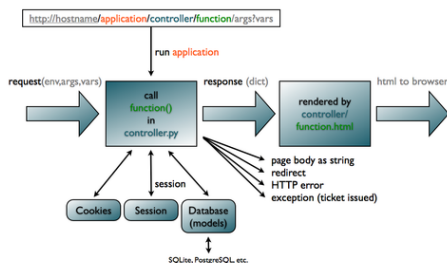
- **Language** - describe how to translate strings in the app to other languages.
- **Modules** - optional Python modules that belong to the app.
- **Static Files** - images, CSS, JS codes, etc  
(Look into the view "layout.html" on how to include your static files.)
- **Plugins** - for extensibility

## web2py Architecture



# The Controller

- Describes the logic and workflow of your app.
- Every URL gets mapped into a call to one of the functions in the controller (a.k.a. action).



`http://127.0.0.1:8000/MyApp/default/index.html`

web2py parses the above URL as:

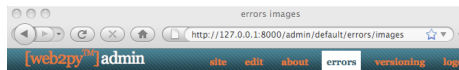
*“Call the function `index()` in a controller named `default.py` which is found in the application named `MyApp`”*

- When a controller returns a dictionary, web2py looks for a view with the name `[controller]/[function].[extension]`

e.g. view  $\rightarrow$  `default/index.html` in the above case.

- If a view is NOT found, the view file `generic.[extension]` used; where `extension`  $\equiv$  `{html, json, xml, rss}` (default is `html`).

- Each web2py app comes with a ticketing system for tracking errors.
- Errors in code are logged and tickets issued to the user.
- The admin only can retrieve and read tickets online.



### Error logs for "images"

[check all](#) [uncheck all](#) [delete all checked](#)

Delete	Ticket	Date and Time
<input type="checkbox"/>	127.0.0.1.2009-06-01.13-42-46.7db8d309-199b-48ed-9849-0718b5ab8d13	2009-06-01 13:42:46

### Error ticket for "images"

Ticket 127.0.0.1.2009-06-01.13-42-46.7db8d309-199b-48ed-9849-0718b5ab8d13

#### Error traceback

```

1. Traceback (most recent call last):
2.   File "/Users/mdi pierro/Desktop/web2py/gluon/
3.     exec code in environment
4.   File "/Users/mdi pierro/Desktop/web2py/applic
5.     1/0
6. ZeroDivisionError: integer division or module
7.

```

## Examples

① `http://127.0.0.1:8000/Hello`

② MVC in action