

CommonConsensus Documentation

Dustin Smith

Revised 26 Oct 2009

Contents

1	Getting up and running	1
1.1	Obtaining the source	2
1.2	Configuring and running the server	3
2	Modifying CommonConsensus	3
2.1	Changing the data	3
2.2	Editing the server	4
2.3	Editing the client	4
2.4	Editing the data directly from the database	5

1 Getting up and running

There are two components to COMMONCONESNUS: the *game client*, a Flash “swif” file, `CommonConsensus.swf`, and the *web server*, a collection of Python files using the Django web application framework. All of the data is stored in a SQLite database, which takes the form of the text file `common_consensus.db`.

To run the server, you will need:

- Python 2.6+ (uses JSON library)
- Django 1.0+
- SQLite 3
- Bazaar client, to obtain the latest code

On Debian-based machines, this should be as simple as issuing the command:

```
sudo apt-get install bazaar python-django python-sqlite sqlite3
```

For other platforms, if you already have `python` and `python-setuptools` installed, you should be able to install `bzr` and `django` using `easy_install`.

One of the main features of the Django library is that it provides a way to access the database without writing SQL (Structured Query Language, pronounced “sequel”) commands. This is known as the ActiveRecord pattern in computer programming: a table roughly corresponds to a class. This means, for example, if you have a table named `Concepts`, Django will allow you to create a temporary new record by the conventional Python syntax for instantiating an object, `c1 = Concepts(name='big cats')`.

If you plan on modifying the *client* binary (students in Henry’s class don’t have to), you will need to compile Adobe Flex code. Flex is a combination of ActionScript 3 and MXML files. These are proprietary formats but ActionScript 3 is based on the same EMCAScript standard as JavaScript, and MXML files are simply XML structures used to describe visual objects and widgets.

The project is built using Flash Bulder 4 (previous versions were called *Flex* Builder). You can obtain a trial version of this beta Eclipse-based IDE from [here](#).

1.1 Obtaining the source

COMMONCONESNUS is still in development stages, so I recommend that you obtain the latest code from the bazaar repository, hosted on launchpad, as opposed to downloading a static snapshot of the code in a current state.

```
bzr branch lp:commonconsensus
```

This will create a folder `trunk` in your current directory with two subdirectories:

```
trunk/  
  /flexserver  
  /game-client
```

From within the root of the repository, the `trunk` folder, you can always obtain the latest version (called the “head”) of the source code by typing:

```
bzr pull
```

1.2 Configuring and running the server

Move into the `flexserver` directory (e.g., `cd trunk/flexserver`). Many of the files in this directory are standard Django¹ scripts: `manage.py`, `settings.py`, and `urls.py`.

There is only one change you need to make before running the server, and this is in the `settings.py` file. Open the file and change the value of the variable `BASE_DIR` to the full path to the `COMMONCONESNUS` server directory. If you are on a Linux environment, this will be the path returned by the command `pwd` when executed in the same directory as the settings file.

After you have updated the `BASE_DIR` variable, you can launch the Python server by typing:

```
python manage.py runserver
```

If you are configured correctly, this will allow your files to be served on the local machine. Check it out by opening this page in your browser:

```
http://localhost:8000
```

You can login to the game with the username and password `test`, which are pre-populated in this development version of the client.

If you want others to be able to access your site or want to host on another port² or be able to access it from a remote server, you will have to specify the host after the `runserver` command. For example:

```
python manage.py runserver xoth.media.mit.edu:80
```

Will launch `COMMONCONESNUS` on `http://xoth.media.mit.edu`. **Note:** if you run `COMMONCONESNUS` on a port different from 8000 or on a server other than your local machine, you will need to specify this in the client by editing the `BASE_URL` variable in the file `game-client/commonconsensus/util/Settings.as` and recompiling.

2 Modifying CommonConsensus

2.1 Changing the data

Add `admin/` onto the the server and open it in your browser. For example, with the default configuration, you'll want to go to:

¹For a good tutorial about how to build an app in Django, go here

²For example, if you host on the port 80, the default HTTP port, then you do not have to specify the port in the URL

http://localhost:8000/admin

Login with **test** for both username and password.

2.2 Editing the server

While `DEBUG=True` in `flexserver/settings.py`, Django will allow you to edit the web application *while* the server is running—it will automatically reload the changed document.

The Django server log, which is printed to the terminal running Django, contains valuable debugging information. Whenever a method in `flexserver/server/views.py` has been called, then its name will be printed to the terminal along with its response code. Some common response codes are:

- 200: Successfully returned request
- 404: Document not found
- 500: Error within the method

Any errors in the python code will result in the opaque 500 error, which alone is not very helpful when you are debugging the project. In Debug settings, Django outputs very helpful information on the error page that can be used to diagnose the problem. Here are some debugging strategies:

- Use the python `print` command to print variables and status messages to the terminal. This does not work if there was an error *before* your print statement is evaluated.
- Insert `assert False` into your code to manually trigger a Django debug page, along with diagnostic information.
- Put `import pdb; pdb.set_trace()` at the top of your method to have python *step through* each line of method and print the response. This is very effective, but requires active participation (you type 'n' to the terminal to advance evaluation to the next line, or *Return* to repeat the last command).

2.3 Editing the client

You should have Adobe Flash/Flex Builder installed or you can use the Flex compiler `mxmlc` but this does not come with an integrated, graphical development environment. Flex Builder is available both as a standalone program or as an Eclipse builder plugin. Students can get a free license for the Professional addition from [here](#).

Once you have it installed, click *File* → *Import* → *Flex Builder Project*, and then select the `game-client` directory.

Alternatively, if you have the `mxm1c` program installed (it will install with Flex Builder, but will not be in your default path), you can run the Makefile script to compile the Flash `swf` by typing `make`.

2.4 Editing the data directly from the database

You can open the SQLite3 database with the following command:

```
sqlite3 common_consensus.db
```

To browse the schema, then type

```
.schema
```

To see the structure for all of the tables. You can execute arbitrary SQL commands from the interface. You can type on multiple lines, and SQLite will wait until it sees a `;` or a system command. Type `.help` to see all system commands; to exit SQLite, type: `.quit`.