Notes on Django and making it work with an already existing mysql database.

I needed to make a website that would give content of some results for RNA. Here are as many instructions I can give on setting up the webserver running django and mysql on apache.

INDEX

A. Setting up Django

First you have to install django.
There are two ways:
1) Eric Fiorenzano's way.(http://www.eflorenzano.com/blog/post/first-two-django-screencasts/) which uses the svn source and uses a script (pylink) to create a correct path between python and django.

```
mkdir /home/esguerra/development/python
cd /home/esguerra/development/python
```

Download django from their svn source, or git repository:

```
git clone http://github.com/django/django.git
```

Download the pylink and pyunlink scripts, move them to your bin folder, and make them executable.

```
git clone git://gist.github.com/21649
git clone git://gist.github.com/21650
mv 21649/pylink.sh /home/esguerra/bin
mv 21650/pyunlink.sh /home/esguerra/bin
rm -rf 21649/ 21650/
chmod +x ~/bin/*.sh
cd django


pylink.sh django
```

2) The ubuntu and fedora ways.

```
sudo apt-get install python-django python-mysqldb
sudo yum django install
```

The next step is to install the apps necessary for the webframework to give special services, in our case we need pagination and sorting.

```
cd /home/esguerra/development/python
git clone http://github.com/ericflo/django-pagination.git
git clone http://github.com/directeur/django-sorting.git
```

Remember to pylink them and open a python terminal and make sure that they work by typing:

```
pylink.sh pagination
>>>import pagination
```

Also the bin utilities of django have to be linked to your bin folder:

```
cd /home/esguerra/development/python/django/django/bin
ln -s `pwd`/* ~/bin
```

He has yet another app to update all applications residing in the python development folder.

curl http://djangosnippets.org/snippets/844/download/ > updateapps.py
python updateapp.py

cd /home/esguerra/development/python
mkdir toolbox
cd toolbox
touch __init__.py
cd ..
pylink.sh toolbox

I have just learned that django-pagination and django-sorting are integrated into fedora, so now you can install all needed dependencies at once:

sudo yum install Django django-pagination django-sorting mysqld MySQL-python

To check that it works

python
>>> import django
>>> import pagination
>>> import django_sorting

One problem with the Django book, and perhaps with the Django philosophy, is that they want to take over the database, so, in case you already have a database and are familiar with MySQL, you just loose a lot of time reading how to set up the models, that is the main reason of me writing this document.

If you forgot to create a superuser for admin you'll have to use:

manage.py createsuperuser --username=joe --email=joe@example.com
My username is admin, and the password is the easiest one I use.

1) Start the project:

bash> django-admin startproject rnadimer

You'll have to edit the settings.py file to get your database, username, password, and applications (e.g. rnabpsdb/steptables) configured.

2) Create an application:

bash-3.2? python manage.py startapp steptables

Notice that I use steptables in plural, since django configures automatic calls from Steptable (Singular)

3) To import the database one uses the utility inspectdb: (Notice that some tinkering has to be done)

bash-3.2? python manage.py inspectdb > tablas/models.py

4) Then you sync it:

```
bash-3.2? python manage.py syncdb
```

5) To make the databases application modifiable with the admin utility you have to create a admin.py file at the databases folder, that is, in the application directory which was created in 2) called /steptables.

```
touch steptables/admin.py
```

6) It's important to add an id field (with auto-increment) to your existing database if it doesn't exist. I did it altering my table in mysql:

```
mysql> ALTER TABLE steps ADD COLUMN id INT(11) AUTO_INCREMENT PRIMARY KEY;
```

And also made sure that there was a reference to it in models.py, that is, I pasted the following line to the Steps class in models.py :

```
   id = models.IntegerField(primary_key=True)
```

7) Now you have to define a view (that's the place where I've been having trouble with the pagination and the optimal table display), so you have to modify views.py. After defining the view, you have to define the urls.py.

8) Make a folder to store the html code for your django site.

```
mkdir /home/html
mkdir /home/html/django_templates
```

9) You have to tell django where your templates are at, so you have to modify this in settings.py.

```
TEMPLATE_DIRS=('/home/html/django_templates',)
```

10) To see what's going on you have to run:

```
bash-3.2? python manage.py runserver
```

And then set your browser to http://127.0.0.1:8000/

B. Setting up Apache.

1) Install mod_wsgi.
Just follow the instructions of how to use django with apache and mod_wsgi.
Now in Fedora 14 it's just:

```
bash-3.2? sudo yum install mod_wsgi
```

2) Make sure that apache is up and configured to start at boot.

```
bash-3.2? sudo chkconfig –-level 345 httpd on
bash-3.2? sudo service httpd status
```

3) Open port (80) using iptables.

4) Restart after succesful configuration.

```
sudo service httpd restart
```

5) Create a file for calling mod_wsgi in /etc/httpd/conf.d/wsgi.conf
The following line should go into wsgi.conf:

```
LoadModule wsgi_module modules/mod_wsgi.so
```

6) Setup your virtual host configuration file at /etc/httpd/conf.d/vhost.conf

C. Installing an App.

1) To create an app.
bash3.2? python setup.py sdist

2) To install a created package
bash3.2? python setup.py install

D. MySQL.

After starting for the first time mysqld a root password has to be created.

sudo chkconfig –level 345 mysqld on
sudo service mysqld start
/usr/bin/mysql_secure_installation //will create root password

Note: the name of the database turned out to be case-sensitive, that is why rnabpsdb is now in lowercase letters.

Login to mysql and create new user.

mysql -u root -p //This will ask for the password you created before
mysql> CREATE DATABASE rnabpsdb; //This creates the database
mysql> GRANT ALL PRIVILEGES ON rnabpsdb.* TO 'esguerra'@'localhost' IDENTIFIED BY 'esguerra'; //This will create the password esguerra for user esguerra to use this database.
mysql> GRANT ALL PRIVILEGES ON *.* TO 'esguerra'@'localhost' with grant option; //This allows esguerra to create databases and erase them.

mysql> create table steps(stack_type varchar(6), step_id varchar(8), count int(10), shift double(5,4), shift_std double(5,4), slide double(5,4), slide_std double(5,4), rise double(5,4), rise_std double(5,4), tilt double(6,4), tilt_std double(6,4), roll double(6,4), roll_std double(6,4), twist double(6,4), twist_std double(6,4), volume double(6,4), rmsd double(6,4), rmsd_std double(6,4));

mysql> load data local infile 'averages4DB.tab' into table steps;

mysql> ALTER TABLE steps ADD COLUMN id INT(11) AUTO_INCREMENT PRIMARY KEY;

Instead of creating tables the usual way, it can also be done using the mysql-workbench, you just have to download and install it.

Once you have your database you can make a backup of it with:
bash# mysqldump -ppassword rnabpsdb > rnabpsdb.sql

And to restore the backup in the existing database:
bash# mysql -u username -ppassword rnabpsdb < rnabpsdb.sql

F. urls.py.

Into main urls.py
(r'^steptables/', include('steptables.urls')),

this calls urls.py in steptables folder, so I have to create a urls.py there.

```
bash> touch steptables/urls.py
```

```
bash> mkdir templates
bash> cd templates
bash> touch base.html
bash> mkdir steptables
bash> cd steptables
bash> touch index.html
```