https://github.com/RunestoneInteractive/RunestoneServer

- Eğer her iki python birden kuruluysa Python3 default python olmalı

```
root@sude:/home/sude# python --version

Command 'python' not found, but can be installed with:

apt install python3
apt install python
apt install python-minimal

You also have python3 installed, you can run 'python3' instead.

root@sude:/home/sude# python3 --version
Python 3.6.8
root@sude:/home/sude# python2 --version

Command 'python2' not found, but can be installed with:

apt install python-minimal

root@sude:/home/sude# pip -V

Command 'pip' not found, but can be installed with:

apt install python-pip

root@sude:/home/sude# pip3 -V
pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
root@sude:/home/sude# _
```

- pip'in güncel ve pip3 olmasına dikkat edelim

Installation

1. Install python.

   First, make sure you have Python installed. Web2py has been ported to Python3, but we have not finished all of our Python 3 testing yet. If you are a developer feel free to jump in with Python 3.

---

2. Install and make a Python virtualenv

   Note, development works well with a Python `virtualenv`. If you don't have root privileges on your computer I strongly recommend you install `virtualenv` and install all of the dependencies there.

   - Documentation here: https://virtualenv.pypa.io/en/stable/
   - Video here: https://www.youtube.com/watch?v=IX-v6yvGYFg
   - For the impatient:

     ```
     $ sudo pip install virtualenv
     $ virtualenv /path/to/home/MyEnv
     $ source /path/to/home/MyEnv/bin/activate
     ```

   - You will need to do the last command every time you want to work on RunestoneServer. If you have not used Python virtual environments before I strongly recommend reading the docs or watching the video

```
root@sude:/home/sude# sudo pip3 install virtualenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/8b/12/8d4f45b8962b03
e438b4fe379d21c0cb8e/virtualenv-16.7.5-py2.py3-none-any.whl (3.3MB)
    100% |████████████████████████████████| 3.3MB 366kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-16.7.5
root@sude:/home/sude#
```

- Python'da sanal makine görevi gören başka kütüphaneler de var. Hangisine aşinaysanız onu kullanabilirsiniz.

```
root@sude:/home/sude# virtualenv sanalortam
Using base prefix '/usr'
New python executable in /home/sude/sanalortam/bin/python3
Also creating executable in /home/sude/sanalortam/bin/python
Installing setuptools, pip, wheel...
done.
```

- "virtualenv" kütüphanesini kullanarak "sanalortam" isimli bir dosya oluşturdum

```
root@sude:/home/sude# dir
sanalortam
root@sude:/home/sude# source sanalortam/bin/activate
(sanalortam) root@sude:/home/sude#
```

- artık oluşturduğum "sanalortam"ı aktive ederek onu sanal makine gibi kullanmaya başlayabilirim.

---

3. Install lots of other dependencies

Ubuntu

On a vanilla Ubuntu (16.04) installation you will need to do at least the following:

```
sudo apt-get install python-pip
sudo apt-get install libfreetype6-dev
sudo apt-get install postgresql-common postgresql postgresql-contrib
sudo apt-get install libpq-dev
sudo apt-get install libxml2-dev libxslt1-dev
```

- Ben sadece Ubuntu üzerinde çalıştığım için burda da Ubuntu üzerinden devam ediyorum.
- Tutorial Ubuntu 16 için hazırlanmış. Ben burda Ubuntu 18 server üzerinden ilerliyorum. Site için kullanılan server da Ubuntu 18.

```
(sanalortam) sude@sude:~$ history
    1  dir
    2  source sanalortam/bin/activate
    3  sudo apt-get install python3-pip
    4  sudo apt-get install libfreetype6-dev
    5  sudo apt-get install postgresql-common postgresql postgresql-contrib
    6  sudo apt-get install libpq-dev
    7  sudo apt-get install libxml2-dev libxslt1-dev
    8  history
(sanalortam) sude@sude:~$ _
```

---

4. Install web2py. The easiest way to do so is to download the **Source Code** distribution from http://www.web2py.com/init/default/download. Here is a direct link to the zip archive. After you download it, extract the zip file to some folder on your hard drive. (web2py requires no real "installation"). I avoid the web2py.app installation on OS X as it messes with the Python path. On Windows, the web2py.exe is also problematic because it won't find modules installed in a virtualenv.

- http://www.web2py.com/examples/static/web2py_src.zip
- Üstteki linkteki dosyayı indiriyoruz. Bu dosya bizim serverımız gibi düşünebilirsiniz.

```
wget http://www.web2py.com/examples/static/web2py_src.zip
```

- “wget” komutunu dosyayı indirmek için kullanabiliriz.
- Bir zip dosyası indiriyoruz. Zip dosyasını çözmek için “unzip” komutuna ihtiyacımız var.
  - ```
    sudo apt install unzip
    ```
- zip uzantılı dosyamızı çözüp kontrol ediyoruz.
  - ```
    unzip web2py_src.zip
    ```
  -
  -

```
(sanalortam) sude@sude:~$ dir
sanalortam  web2py  web2py_src.zip
(sanalortam) sude@sude:~$ cd web2py/
(sanalortam) sude@sude:~/web2py$ dir
anyserver.py  examples    gluon     MANIFEST.in      scripts        web2py.py
applications  extras      handlers  NEWINSTALL       site-packages
CHANGELOG     fabfile.py  LICENSE   README.markdown  VERSION
(sanalortam) sude@sude:~/web2py$ _
```

---

5. Get familiar with the Runestone Components, which were installed with pip. They come from https://github.com/RunestoneInteractive/RunestoneComponents and there are good quick start instructions there.

- https://github.com/RunestoneInteractive/RunestoneComponents

```
$ sudo pip install virtualenv
$ virtualenv /path/to/home/MyEnv
$ source /path/to/home/MyEnv/bin/activate
```

- You will need to do the last command **every time** you want to work on RunestoneComponents. If you have not used Python virtual environments before I strongly recommend reading the docs or watching the video
- *Note:* You might need to install `pip` based on how you have installed python.

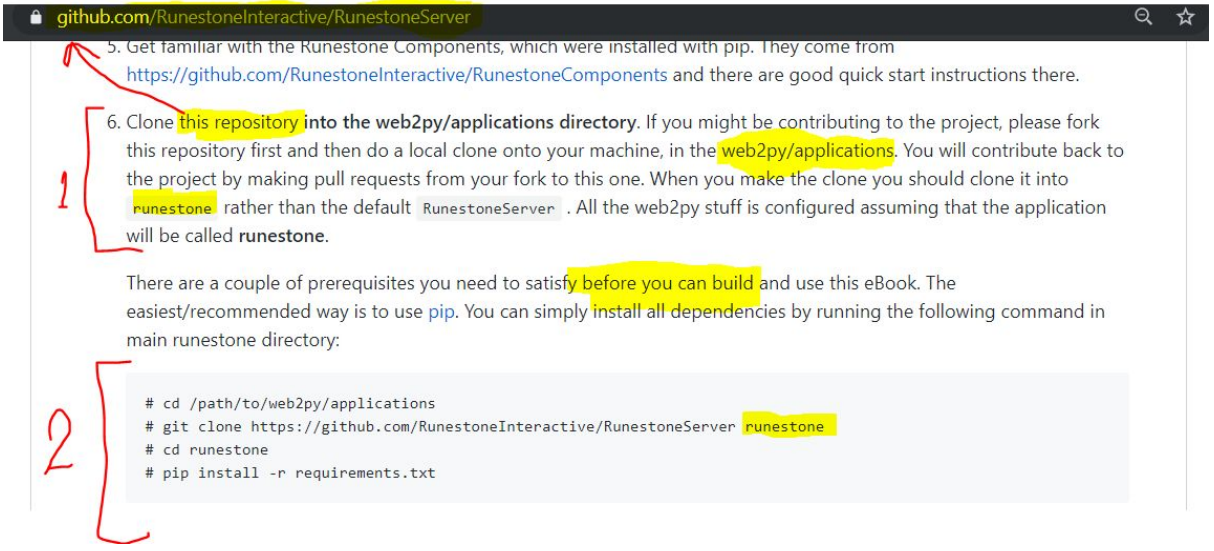
With the virtual environment installed and configured you can continue.

```
pip install runestone
```

Or, if you prefer to live on the development edge, you can check out the very latest from:

```
pip install git+git://github.com/RunestoneInteractive/RunestoneComponents.git
```

- Runestone Components linki üzerinden ilerleyip server kurulumuna sonra devam edebilirsiniz. Ben server kurulumundan devam edeceğim.



- Uygulamalarımızın içine Runestone Server'ı **runestone adıyla** indirdik

```
(sanalortam) sude@sude:~/web2py/applications$ git clone https://github.com/RunestoneInteractive/Rune
stoneServer runestone
Cloning into 'runestone'...
remote: Enumerating objects: 129, done.
remote: Counting objects: 100% (129/129), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 31768 (delta 66), reused 109 (delta 63), pack-reused 31639
Receiving objects: 100% (31768/31768), 165.19 MiB | 962.00 KiB/s, done.
Resolving deltas: 100% (21102/21102), done.
(sanalortam) sude@sude:~/web2py/applications$ cd runestone/
(sanalortam) sude@sude:~/web2py/applications/runestone$ pip3 install -r requirements.txt
Obtaining file:///home/sude/web2py/applications/runestone/rsmanage (from -r requirements.txt (line 2
4))
```

- runestone için gerekli olan paketleri yüklemek için “requirements.txt” dosyasını kullanıyoruz.
    -
```
(sanalortam) sude@sude:~/web2py/applications/runestone$ pip3 install -r requirements.txt
```

- Serverımızın içine Runestone Server uygulamamızı yükledik. Artık kitap ekleyebiliriz.

7. Clone the book that you want to use, **into the web2py/applications/runestone/books** directory. You can see some of the available books at https://github.com/RunestoneInteractive. Again, if you might contribute back to the book, please fork the book repository first and then do a local clone onto your machine.

```
(sanalortam) sude@sude:~/web2py/applications/runestone$ dir
books            controllers         doctrees     lti.config  README.rst            scripts
build            docker              ext_test     models      requirements-test.txt  static
ChangeLog.rst    docker-compose.yml  __init__.py  modules     requirements.txt       tests
CONTRIBUTING.md  Dockerfile          LICENSE      pylintrc    rsmanage               views
(sanalortam) sude@sude:~/web2py/applications/runestone$ cd books/
(sanalortam) sude@sude:~/web2py/applications/runestone/books$
```

- Foundations of Python Programming → https://github.com/RunestoneInteractive/fopp
- How to Think Like a Computer Scientist, Interactive Edition → https://github.com/RunestoneInteractive/thinkcspy
- Computer Science Principles → https://github.com/RunestoneInteractive/StudentCSP

```
git clone https://github.com/RunestoneInteractive/thinkcspy
git clone https://github.com/RunestoneInteractive/fopp
git clone https://github.com/RunestoneInteractive/StudentCSP
```

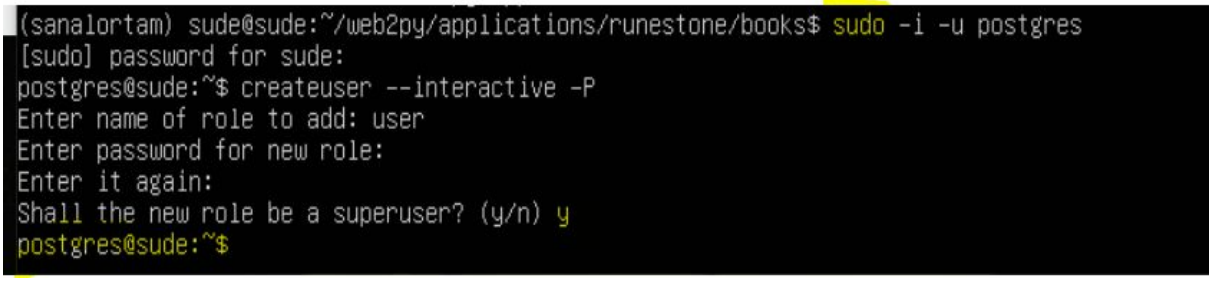
```
(sanalortam) sude@sude:~/web2py/applications/runestone/books$ dir
fopp  StudentCSP  thinkcspy
```

- Runestone ortamına göre yazılmış bazı kitapları indirdik.
- Kitapları source code gibi düşünmeliyiz. Derleyip çalıştırmadan kullanamıyoruz.
- Derleyip çalıştırma işlemini database’i initialize ettikten sonra yapmalıyız.

---

8. Set up your local database
    - Configure Postgresql (or you can try mySQL, but there may be some issues with field lengths with that.)
    - Create a database
        - For Ubuntu you will need to do the following first:

```
$ sudo -i -u postgres
$ postgres@ubuntu:~$ createuser --interactive -P
Enter name of role to add: <your name here>
Enter password for new role: <a password for this user>
Enter it again: <again>
Shall the new role be a superuser? (y/n) y
Password: <password for the default, postgres user>
```

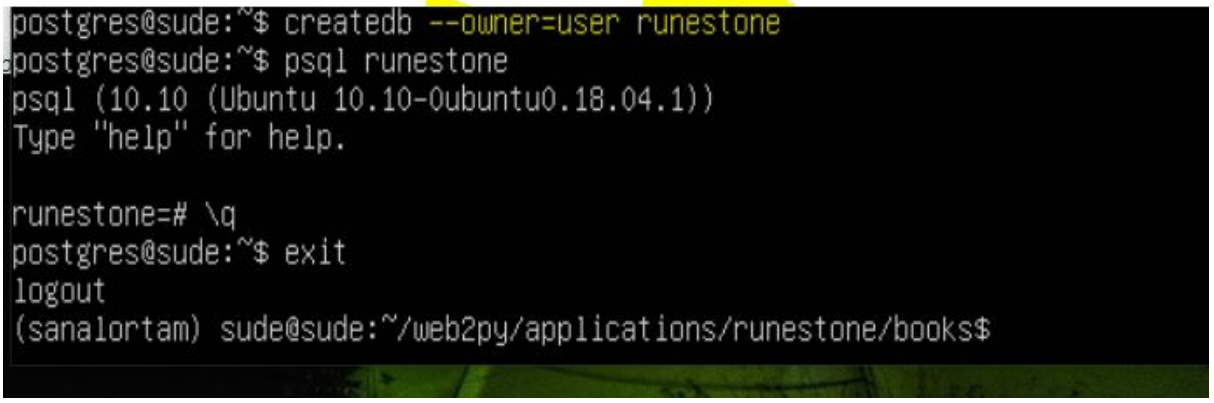


- postgres veritabanına bağlanıyoruz.
- superuser rolünde bir kullanıcı yaratıyoruz. Yarattığımız kullanıcının bilgilerini kaydedelim. Daha sonra start dosyasına ekleyeceğiz.
- Burda “user” adında “password” şifresinde sahip superuser rolünde bir kullanıcı yarattım.

        - On both Mac and Ubuntu you can now do the following; for Windows, use `"C:\Program Files\PostgreSQL\9.6\bin\createdb" --owner=<yournamehere> -U postgres runestone`. Enter the password for the default, postgres user (not for the newly-created user).

```
$ createdb --owner=<yournamehere> runestone

$ exit

psql runestone
psql (9.5.3)
Type "help" for help.

runestone=# \q
$
```

- Yarattığımız kullanıcının sahibi olacağı “runestone” isimli bir tablo yaratıyoruz.



https://www.2ndquadrant.com/en/blog/how-to-safely-change-the-postgres-user-password-via-psql/
https://www.liquidweb.com/kb/what-is-the-default-password-for-postgresql/

- Figure out your database connection string. It will be something like `postgresql://username:passwd@localhost/dbname`
- Tell web2py to use that database:
  - If you're running https, edit `settings.server_type` in `web2py/applications/runestone/models/0.py`.
  - Set and export environment variable for DBURL -- Note the url format for web2py is different from sqlalchemy. use postgres for web2py and postgresql for sqlalchemy. example: postgresql://username:pw@host/database where pw may be empty, and database is the database you created above, runestone.
  - Set and export environment variable WEB2PY_CONFIG. If set to production, it will get the database connection string from DBURL. If set to development, it will get the database connection string from DEV_DBURL. If set to test, it will get it from TEST_DBURL.
  - Set and export environment variable WEB2PY_MIGRATE. If set to Yes, web2py will check on each page load whether any database migrations are needed and perform them. If set to No, web2py will just assume that models match the database. If set to Fake, web2py will try to update the metadata it maintains about the database tables to match the models, but will not make any changes to the database; use that setting only for repairs when something has gone wrong.
  - If you want to customize other settings you can create a file `applications/runestone/models/1.py` using `models/1.py.prototype` as the template. If you have your environment variables set up as explained above you probably won't need to worry about this for your initial setup.

```
export WEB2PY_CONFIG=production # or development or test
export WEB2PY_MIGRATE=Yes
export DBURL=postgresql://username:pw@host/database
export TEST_DBURL=postgresql://username:pw@host/database
export DEV_DBURL=postgresql://username:pw@host/database
```

- Runestone isimli database oluşturduktan sonra serverın database ile bağlantısını sağlayabilmek için çevre değişkenlerini tanımlamamız gerekiyor.
- Çevre değişkenlerinin sanal ortamda atanması konusunun üzerinden devam edelim.

https://stackoverflow.com/questions/9554087/setting-an-environment-variable-in-virtualenv

34

Using only virtualenv (without virtualenvwrapper), setting environment variables is easy through the `activate` script you sourcing in order to activate the virtualenv.

Run:

```
nano YOUR_ENV/bin/activate
```

Add the environment variables to the end of the file like this:

```
export KEY=VALUE
```

You can also set a similar hook to unset the environment variable as suggested by Danilo Bargen in his great answer above if you need.

share improve this answer

answered Jan 4 '14 at 7:58

- değişkenlerimizin sistem genelinde değil sadece serverımızın kurulu olduğu ortamda tanımlı olmasını istiyoruz. Ayrıca birden fazla runestone server ve database kullanma durumumuzda her biri için kendi sanal ortamında düzenleme yapmak daha doğru olacaktır.

- Sanal ortamda çevre değişkeni tanımlamak için “activate” dosyasını “etc/environment” gibi kullanabilirsiniz.

```
51  sudo nano /home/sude/sanalortam/bin/activate
```

- Sanal ortamımızı başlattığımız dosyaya çevre değişkenlerimizi set ediyoruz.

```
export WEB2PY_CONFIG=production
export WEB2PY_MIGRATE=Yes
export DBURL=postgresql://user:password@localhost/runestone
export TEST_DBURL=postgresql://user:password/runestone
export DEV_DBURL=postgresql://user:password/runestone
```

- Tanımladığımız değişkenlerin de kullanılabilmesi için sanal ortamımızı tekrar başlatıyoruz.

```
sude@sude:~/web2py/applications/runestone/books$ source /home/sude/sanalortam/bin/activate
(sanalortam) sude@sude:~/web2py/applications/runestone/books$ echo $DBURL
postgresql://user:password@localhost/runestone
(sanalortam) sude@sude:~/web2py/applications/runestone/books$ echo $WEB2PY_CONFIG
production
(sanalortam) sude@sude:~/web2py/applications/runestone/books$
```

More on Environment Variables

There are a few environment variables that you can use to control the runestone server out of the box:

- WEB2PY_CONFIG You can set this to production, development, or test. Each mode can have a corresponding database connection environment variable. They are:
- for development use DEV_DBURL
- for test use TEST_DBURL
- for production use DBURL Yes, its not quite consistent, but its backward compatible for the way we have been doing things.

- artık hem serverımız kurulu hem de çevre değişkenlerimiz tanımlı olduğuna göre database’imizi initialize edebiliriz.
- “rsmanage initdb --” komutunu kullanıyoruz.

9. run rsmanage initdb -- This will initialize the database so you can build your first book. The rsmanage command was installed when you ran pip install -r requirements.txt in a previous step. If you are upgrading you should run pip install -e rsmanage from the applications/runestone directory.

   Important

   Database errors

   If you get an error message that the session table already exists, you need to go into the database and drop the table. If you get other error messages about tables that either exist or do not exist when they should or should not, then your database is out of sync with the data in your databases folder created by web2py. This is not a happy spot to be in. rsmanage initdb --reset will definitely get things back in order for a new installation.

   If this is an old installation and you don't want to lose any data the you can try setting the WEB2PY_MIGRATE variable to 'Fake' But, this may cause cause even more problems, so only use it if you really know what changes you have made to the database schema and why. You may need to study sql.log to figure out which talbes need to be migrated by hand.

- ben burda dependency yüzünden hata aldım. Siz de hata aldıysanız bir sonraki adımı inceleyebilirsiniz.

```
Initializing the database
Running: python web2py.py --no-banner -S runestone -M -R applications/runestone/rsmanage/initialize_
tables.py
Traceback (most recent call last):
  File "/home/sude/web2py/gluon/restricted.py", line 219, in restricted
    exec(ccode, environment)
  File "applications/runestone/models/db.py", line 54, in <module>
    os.environ.get("WEB2PY_MIGRATE", "Yes") in ["Yes", "Fake"]
  File "/home/sude/web2py/gluon/packages/dal/pydal/base.py", line 171, in __call__
    obj = super(MetaDAL, cls).__call__(*args, **kwargs)
  File "/home/sude/web2py/gluon/packages/dal/pydal/base.py", line 477, in __init__
    "Failure to connect, tried %d times:\n%s" % (attempts, tb)
RuntimeError: Failure to connect, tried 5 times:
Traceback (most recent call last):
  File "/home/sude/web2py/gluon/packages/dal/pydal/base.py", line 457, in __init__
    self._adapter = adapter(**kwargs)
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/postgres.py", line 27, in __call__
    return AdapterMeta.__call__(cls, *args, **kwargs)
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/__init__.py", line 39, in __call__
    obj = super(AdapterMeta, cls).__call__(*args, **kwargs)
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/postgres.py", line 54, in __init__
    driver_args, adapter_args, do_connect, after_connection)
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/base.py", line 369, in __init__
    super(SQLAdapter, self).__init__(*args, **kwargs)
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/base.py", line 50, in __init__
    self.find_driver()
  File "/home/sude/web2py/gluon/packages/dal/pydal/adapters/base.py", line 101, in find_driver
    str(self.drivers))
RuntimeError: No driver of supported ones ('psycopg2',) is available


Database Initialization Failed
(sanalortam) sude@sude:~/web2py/applications/runestone/books$ echo $DBURL
postgresql://user:password@localhost/runestone
(sanalortam) sude@sude:~/web2py/applications/runestone/books$ _
```

```
RuntimeError: No driver of supported ones ('psycopg2',) is available


Database Initialization Failed
(sanalortam) sude@sude:~/web2py/applications/runestone/books$
```

- bu hatayı düzeltmek için psycopg2 kütüphanesini yüklememiz gerekiyor.

```
(sanalortam) root@sude:/home/sude/web2py/applications/runestone# pip install psycopg2
```

- psycopg2 kütüphanesi postgresql için database adapter görevini görüyor.

```
(sanalortam) root@sude:/home/sude/web2py/applications/runestone/books# rsmanage initdb --
It appears you already have database migration information do you want to proceed? [y/N]: y
Initializing the database
Running: python web2py.py --no-banner -S runestone -M -R applications/runestone/rsmanage/initialize_
tables.py
Defining initial Courses
Adding Constraints and Indices
(sanalortam) root@sude:/home/sude/web2py/applications/runestone/books#
```

- Eksik kütüphaneyi de yükledikten sonra “rsmanage initdb --” komutu ile database’i initialize edebildim.

- Dependencyleri yükledikten sonra farklı hatalar alırsanız. “runestone” klasöründe “pip install runestone” ve “pip install -r requirements.txt” adımlarını tekrarlayın.Daha sonra “rsmanage initdb --” ile devam edin. Eğer database’i initialize etmeyi hiç yapamadıysanız “rsmanage initdb --reset”’i **DENEMEYİN**. Initialize etmediğiniz bir şeyi resetleyemezsiniz.

---

10. Build the book.

```
$ cd web2py/applications/runestone/books/<your book>
$ runestone build
$ runestone deploy
```

- At the end of the build step it should say `trying alternative database access due to No module named pydal` and then, if things are working correctly, start outputting the names of the chapters.

- Hali hazırda github üzerinden kitapları indirmiştik. İndirdiğimiz kitap dosyalarını source kod olarak düşünebiliriz. “runestone build” komutu ile aslında kitabımızı derlemiş, “runestone deploy” komutu ile çalıştırmış oluyoruz. Bu noktada (“runestone deploy” komutundan sonra) kontrol etmek için “runestone serve” komutunu kullanarak localhost üzerinden ilgili kitaba ulaşabilirsiniz. “runestone serve” komutu ile sadece tek bir kitabı kullanabilirsiniz.Sınıf yaratamaz, assignment oluşturamazsınız. Tam anlamıyla Runestone Interactive’i kullanabilmek için tutoriala devam etmeniz gerekir.

- “runestone serve --port=80” vs şeklinde de kullanılabilir.

```
(sanalortam) root@sude:/home/sude/web2py/applications/runestone/books# cd fopp
(sanalortam) root@sude:/home/sude/web2py/applications/runestone/books/fopp# runestone build
```

```
dumping object inventory... done
build succeeded, 1 warning.

The HTML pages are in build/fopp.
Done, build successful
(sanalortam) root@sude:/home/sude/web2py/applications/runestone/books/fopp# runestone deploy
```

```
fopp/doctrees/Tuples/toctree.doctree

sent 44,600,365 bytes  received 22,232 bytes  89,245,194.00 bytes/sec
total size is 44,501,213  speedup is 1.00
```

---

then, if things are working correctly, start outputting the names of the chapters.

11. Additional Steps for TextBook as a Service (Build your Own Course)

This step is somewhat optional even for developers, depending on what you are working on. But if you want to be able to click on the build a course button you'll need to do the following.

```
$ cd web2py
$ cp applications/runestone/scripts/start .
$ cp applications/runestone/scripts/run_scheduler.py .
```

Now you will want to edit the start script according to your setup. Then use the start script to start web2py and the scheduler together. Do not just run python web2py.py directly.

- Bu tutorialı hazırlarken “run_schedular.py” dosyası runestone’u indirirken dahil değildi. Ben de kendi eski dosyalarımı “https://github.com/elifsudegokay/yeditepe_runestone_foo” adresine ekledim. Burdan **“start”, “run_schedular.py”** ve **“routes.py”** dosyalarını web2py klasörüne kopyalayın.

```
(sanalortam) root@sude:/home/sude/web2py# dir
anyserver.py  examples     handlers         logs                routes.py          start
applications  extras       httpserver.log   MANIFEST.in         run_scheduler.py   VERSION
CHANGELOG     fabfile.py   httpserver.pid   parameters_8000.py  scripts            web2py.py
deposit       gluon        LICENSE          README.markdown     site-packages      welcome.w2p
(sanalortam) root@sude:/home/sude/web2py# _
```

- start dosyasındaki **DBUSER** ve **DBPASS** değişkenlerinin değerlerini kendi database kullanıcınıza göre düzenleyin.

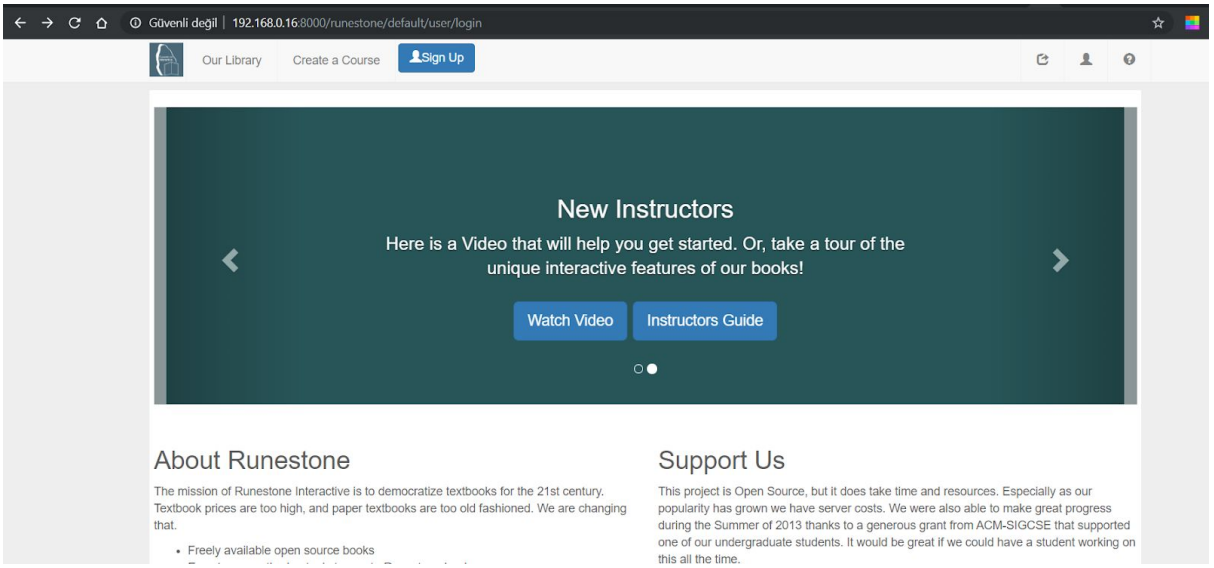
- start dosyası

- ```
  GNU nano 2.9.3                                start

  #!/bin/bash

  # A simple script to start web2py and the companion scheduler app for building
  # books.  Copy these into the main web2py folder and use them there
  #

  export DBUSER=user
  export DBPASS=password
  export DBHOST=localhost
  export DBNAME=runestone

  echo "Be sure to activate your virtual environment"
  #source ~/Environments/web2py/bin/activate
  python web2py.py --ip=0.0.0.0 --port=8000 --password='<recycle>' -K runestone --nogui -X  &
  sleep 3
  echo "starting scheduler"
  python run_scheduler.py &
  ```

- **routes.py** dosyası web2py'ın yönlendirme dosyasıdır. web2py'ı başlattığınızda normalde karşınıza web2py'ın admin sayfası çıkar. routes.py dosyasını düzenleyerek default application'ı runestone yaptığımızda websitesi açıldığında runestone interactive uygulamasına otomatik yönlendirmiş olacağız.
  - ```
    GNU nano 2.9.3                                routes.py

    routers = dict(
            BASE = dict(
                    default_application='runestone',
            )
    )
    ```
-

- start dosyasını çalıştırarak serverı başlatabilirsiniz. Yukarıda da belirtildiği gibi web2py dökümanlarından yararlanarak web2py.py dosyasını çalıştırıp serverı **başlatmayın**. En basidinden database kullanıcı adı ve şifresi start dosyasında tanımlanıyor.

- localhost:8000 üzerinden siteye erişebilirsiniz.
- Ben serverı virtualboxa kurduğum için ana-hostumda VM ipsi üzerinden bağlandım.



- Karşınıza böyle bir site gelmesi gerekiyor.