Step-by-Step: Installing Django

Author: Josh VanderLinden <codekoala@gmail.com>

Date: 11 Aug 2008

Homepage: http://www.codekoala.com/

URL: http://www.codekoala.com/blog/2008/aug/11/step-step-installing-django/

Being the Django and Python zealot that I am, I often find myself trying to convert those around me to this awesome development paradigm. Once I break them, these people often ask me a lot of questions about Django. Over the past few months I've noticed that one of the biggest sticking points for people who are new to Django is actually getting it up and running to begin with. In response, this is the first in a series of articles dedicated to getting Django up and running.

What is Django?

The Django Web site describes Django as "a high-level Python Web framework that encourages rapid development and clean, pragmatic design." Basically, Django is just about the most amazing thing for Web development. I have tinkered with several different Web technologies, but nothing seems to even come close to what Django can do for me.

What is Python?

Python is a programming language used in numerous aspects of computing these days. It has a very simple yet powerful syntax. It's an easy language for beginners to pick up, but it provides adequate levels of power for the more experienced developers out there. If you have never programmed anything before, or you have dabbled with something like BASIC, Python should be fairly straightforward. If you are a programming veteran, but have only worked with languages like C, C++, Java, etc, you might struggle a bit with the syntax of the language. It's not difficult to overcome the differences in a couple hours of hands-on development.

Let's get started.

Installing Python...

Having Python installed is critical--Django does not work without Python. I'm guessing that you're relatively familiar with the procedures for installing software packages on your particular operating system. However, I will share a few notes to point you in the proper direction if you're lost. If nothing else, just head over to the Python download page to download anything you need to install Python. I whole-heartedly recommend using the latest stable version of Python for Django, but you should be able to get by with as early a version as 2.3.

...On Windows

Simply grab the latest version of the Python installer. It is currently version 2.5.2. Once the installer has downloaded successfully, just run through the installation wizard like any other setup program.

...On Mac OS X

Recent Mac OS X computers come with Python pre-installed. To determine whether or not you actually have it, launch the Terminal (Applications > Utilities > Terminal) and type `python -c "import sys; print sys.version"`. If Python is already installed, you will see the version you have installed. If you have a version that is less than 2.3, you should download the newest version. If you don't have Python installed, you will get a "command not found" error. If you're in this boat, just download the latest version of the Python Universal installer and install it.

...On Linux

Most Linux distributions also have Python pre-installed. Just like with Mac OS X, you can check to see by opening up a terminal/konsole session and running the command `python -c "import sys; print sys.version"`. If you have Python installed, you will see its version. If you get an error message when running that command, or you have a version earlier than 2.3, you need to download and install the latest version of Python.

If you're running a Debian-based distribution (like Ubuntu, sidux, MEPIS, KNOPPIX, etc), you can probably use `sudo apt-get install python` to get Python. If you're running an RPM-based Distribution, you can probably use something like Yum or YaST to install Python.

A sure-fire way to install Python on any Linux system, however, is to install from source. If you need to do this, you simply:

1. download the source for the latest version of Python
2. extract it: `tar jxf Python-2.5.2.tar.bz2`
3. go into the newly-extracted directory: `cd python-2.5.2`
4. configure it: `./configure`
5. compile it: `make`
6. install it: `make install`

(I've only installed Python from source one time, so I might be wrong)

Setting Up Your PYTHONPATH...

Generally speaking, if you didn't have Python installed before starting this tutorial, you will need to setup your `PYTHONPATH` environment variable. This is a variable that lets Python know where to find useful things (like Django).

...On Windows

- Open up your System Properties (Win+Break or right click on "My Computer" on your desktop and select Properties)
- Go to the "Advanced" tab
- Click the "Environment Variables" button
- If you have permission to change system variables, click the "New" button in the bottom pane. Otherwise, create the `PYTHONPATH` variable for your user account using the "New" button in the top (User variables for [username]) pane.
- Set the variable name to `PYTHONPATH`
- Set the variable value to `C:\Python25\Lib\site-packages` (replace `C:\Python25\` with whatever it is on your system if needed)
- Save it

You may also need to add the `python` executable to your `PATH`. If you can successfully run `python` from a command prompt window, you don't need to worry about it.

If you can't run `python` from a command prompt, follow the procedure above, but use the `PATH` variable instead of `PYTHONPATH`. `PATH` most likely already exists, so you just need to append/prepend the existing value with something like `C:\Python25\` (again, this might need to change depending on where you installed Python)

...On Mac OS X

Your `PYTHONPATH` should already be setup for you.

...On Linux

Usually you just need to edit your `~/.bash_rc` script to setup your `PYTHONPATH` environment variable. Go ahead and open that up in your preferred text editor and make sure there's something in it like:

```
export PYTHONPATH=/usr/lib/python2.5/site-packages:$PYTHONPATH
```

Save any changes necessary and run the following command:

```
source ~/.bash_rc
```

This will take care of updating your current session with any changes you made to your `~/.bash_rc`.

Installing Django

Once you have Python and have verified that you have version 2.3 or later, you are ready to install Django. Currently, the latest stable release is 0.96.1, but this is grossly out-dated. Django 1.0 will be released on September 2nd 2008, so the "unstable" copy of Django is pretty close to what 1.0 will have to offer. There are some incredibly useful improvements in the unstable version that I don't think I could do without anymore, so that's what I'll talk about installing here.

First, you need to have a subversion client. On Windows, the most popular one is called TortoiseSVN. On Mac OS X, I have played with a few, but I think Versions is a pretty decent one. Linux also has several to choose from, but if you're using Linux, you're probably going to use the command line anyway (right?).

For brevity, I will just use the subversion commands necessary to accomplish this task (instead of discussing all GUI interfaces to subversion).

The exact location that Django should be installed differs from system to system, but here are some guidelines for typical setups:

- Windows: `C:\Python25\Lib\site-packages`
- Linux: `/usr/lib/python2.5/site-packages`
- Mac OS X: `/Library/Frameworks/Python.framework/Versions/2.5/lib/python2.5/site-packages`

If you want a definite location, run the following command:

```
python -c "from distutils.sysconfig import get_python_lib; print get_python_lib()"
```

Once you know that location, go there in your command prompt or terminal session. Then execute this command `svn co http://code.djangoproject.com/svn/django/trunk/django django`. You will see loads of output, showing all of the files that you are downloading as you install Django.

As soon as that process completes, you should run `python -c "import django"` to make sure everything worked properly. If the command doesn't display an `ImportError`, you're good. Otherwise, you need to try again.

Getting Access to Django Scripts...

Once you can successfully `import django`, you might want to make sure you can run the `django-admin.py` script that comes with Django.

...On Windows

This process is very similar to what we did with the `PYTHONPATH` environment variable earlier.

- Open your System Properties again
- Go to the Advanced tab
- Click the Environment Variables button
- Find your `PATH` environment variable (either for your user or system-wide)
- Make sure that the variable value *contains* something like `C:\Python25\Lib\site-packages\django\bin`
- Save any changes
- Open a fresh command prompt
- Try to run `django-admin.py`. If you're successful, you're ready to get started with Django. Otherwise, you need to fix your path to `django/bin` or just call the `django-admin.py` script using an absolute path when needed.

...On Mac OS X

You can run a command similar to this:

```
sudo ln -s /Library/Frameworks/Python.framework/Versions/2.5/lib/python2.5/site-packages/django/bin/django-admin.py /usr/local/bin
```

...On Linux

If you have "root" privileges on your Linux system, you can execute a command like:

```
sudo ln -s /usr/lib/python2.5/site-packages/django/bin/django-admin.py /usr/local/bin
```

If you don't have "root" privileges, you can setup your own `/usr/local/bin`:

```
mkdir ~/bin
```

Make sure your `~/.bash_rc` contains something like:

```
export PATH=$HOME/bin:$PATH
```

Then update your current session with any changes you made to `~/.bash_rc` by running this command:

```
source ~/.bash_rc
```

And that should do it! Now you *should* be ready to get started with Django.

Feel free to leave a comment if you're having problems installing Django. Good luck!

Check out Installing Django on Shared Hosting.

Comments

The Apprentice said...

Thanks for the cool walk through, and the one on one help, other than the SVN nonsense that I was struggling with it all worked like a charm

Posted: 2008-08-13 19:03:14

Nate said...

Thanks for the step by step. I've been struggling with trying to get my PATH and PYTHONPATH vars set up right for a day now, and I finally found your article - the first to explain that the djangobin folder needs to be in PATH. Thanks!!

Posted: 2009-04-10 19:10:58.763957