My First Django Project

Adrian Kosmaczewski

2008-01-11

So here it is, my first Django project: the gazillionth blog engine on the planet!. As if there weren't enough, right? :) Actually it was a practical and easy way to learn the Django project, and the result is pretty neat. Feel free to download it, play with it, and give me your feedback. Here's a sample screenshot in Safari:

Creating this project I have had a practical experience comparing both Django and, of course, Rails. The subject is not new in this blog; however, this time I could play with both frameworks and as such, I can bring my small amount of confusion in this big framework tar pit.

Personally, I found Django and Rails much more different than I previously thought. Both frameworks tackle similar problems, sometimes in similar ways; however; but this does not make for a real similarity between both. They have different philosophical approaches to the problem. This must surely have to do with the underlying programming language and their philosophies.

That said, I found Rails much easier to use. It took me far, far less time to do a similar small application in Rails than it took me to do it on Django. And I must say that when I first used Rails, I had never seen a line of Ruby in my life. I will list some pros and cons that I found in the process:

Good things about Django:

- The "automatic administration system"; impressive! (but you can get it in Rails with some external plugins like Streamlined… but in any case it's a handy thing to have it already in the system!)
- The form subsystem; great!
- Django does not force you a folder layout; nice!
- The Django Book + documentation: really great resources, nothing to say about that.
- The deployment procedure: really much easier than Rails! We're in something really close to PHP here, while Capistrano has much more to do with Java's Maven or even makefiles :)
- The native support for RSS feeds! This is a godsend.

Bad things about Django:

- No native support for “environments”, like in Rails; you cannot separate easily the settings for development, testing and production (and you can add more environments if you want!)
- No native support for REST architectures, or at least “easy” AJAX + API support (like the one you find in Rails)
- The “syncdb” system that ships with Django is, at most, primitive, compared to Rails migrations.
- There doesn’t seem to be a built-in infrastructure for tests.

  There is a primitive built-in test infrastructure. You can add “doctests” to your views and models, and this way you’re “documenting and testing” at the same time. Sorry, but I don’t buy this. Tests are tests, docs are docs. Rails, thanks to rakefiles and unit, functional and integration tests, seems is a much more advanced platform in this sense. You can get even stats of your projects with it! You can test them! Extract the docs! Everything! This is something lacking in Django. Really. Not to speak about the lack in Python of something similar to RSpec, which just by itself justifies the choice of Ruby and Rails in this field.
- Lack of integrated logging. Can you believe this? I had to find an external source of inspiration for this.
- The naming of “Views” for “Controllers” (or said otherwise, the MTV instead of MVC thing) does not make sense to me. Even in WebObjects and Cocoa “views” are, well, “views”…
- The commands for creating, starting or using an application from the command line are slightly harder to remember (or maybe in Rails are just far too easy to remember!)
- The template system; I do not want to learn another language for that. Django uses a really limited one for this matter, and I prefer Rails’ (default) one in this case.
- Finally, Python; I just can’t get used to this language’s syntax. But I can live with that ;) It’s a matter of personal taste, simply.

Of course, these impressions have nothing to do with the power of the platform! Python is a powerful language, and many of the “newbie” observations above have to do with my own lack of knowledge about it. But first impressions count!

In any case, I liked playing with Django & Python; I’m happy to have learnt something new!

Update, 2009-02-26: I’ve posted the code in Github now, and also I’ve fixed its compatibility problems, and now it works with Django 1.0.2.