

Django -vs- Grails -vs-?

Asked 16 years, 3 months ago Modified 3 years, 4 months ago

Viewed 20k times



23

I'm wondering if there's such a thing as Django-like ease of web app development combined with good deployment, debugging and other tools?



Django is a very productive framework for building content-heavy sites; the best I've tried and a breath of fresh air compared to some of the Java monstrosities out there. However it's written in Python which means there's little real support in the way of deployment/packaging, debugging, profilers and other tools that make building and maintaining applications much easier.

Ruby has similar issues and although I do like Ruby **much** better than I like Python, I get the impression that Rails is roughly in the same boat as Django when it comes to managing/supporting the app.

Has anyone here tried both Django and Grails (or other web frameworks) for non-trivial projects? How did they compare?

python

django

frameworks

Share

asked Sep 16, 2008 at 19:05

Improve this question

Follow

community wiki

[Rob](#)

9 Answers

Sorted by:

Highest score (default)



32

You asked for someone who used both Grails and Django. I've done work on both for big projects. Here's my Thoughts:



IDE's: Django works really well in Eclipse, Grails works really well in IntelliJ Idea.



Debugging: Practically the same (assuming you use IntelliJ for Grails, and Eclipse for Python). Step debugging, inspecting variables, etc... never need a print statement for either. Sometimes django error messages can be useless but Grails error messages are usually pretty lengthy and hard to parse through.

Time to run a unit test: django: 2 seconds. Grails: 20 seconds (the tests themselves both run in a fraction of a second, it's the part about loading the framework to run them that takes the rest... as you can see, Grails is frustratingly slow to load).

Deployment: Django: copy & paste one file into an apache config, and to redeploy, just change the code and

reload apache. Grails: create a .war file, deploy it on tomcat, rinse and repeat to redeploy.

Programming languages: Groovy is TOTALLY awesome. I love it, more so than Python. But I certainly have no complaints.

Plugins: Grails: lots of broken plugins (and can use every java lib ever). Django: a few stable plugins, but enough to do most of what you need.

Database: Django: schema migrations using South, and generally intuitive relations. Grails: no schema migrations, and by default it deletes the database on startup... WTF

Usage: Django: startups (especially in the Gov 2.0 space), independent web dev shops. Grails: enterprise

Hope that helps!

Share Improve this answer


answered Dec 23, 2009 at 22:42

Follow

community wiki
[hendrixski](#)

2 Just to add a bit of update to my post: For plugins in Django, start with Pinax. It's the only real repository of integrated plugins. With grails there's a command for just listing all plugins and installing what you want. – [hendrixski](#) Jan 11, 2010 at 18:05

2 Grails db migration plugin, [grails-plugins.github.com/grails-database-migration/docs/manual/...](https://github.com/grails-database-migration/docs/manual/) – James McMahon Apr

-
- 2 IMHO number of bugs that are present in Grails vs Django core should also be compared specially when grails has lots and lots of them. Very simple stuff like running unit tests multiple times from the same grails environment is broken and it takes eternity to load and run them. Also the learning curve with grails is a lot more than django unless you are building something trivial. With the monsters like Spring and Hibernate makes learning grails very very steep for someone who has not worked before with these monsters. – [Divick](#)
Aug 15, 2013 at 8:35 
-

Also conventions in Grails are sometimes too much ... that it makes it difficult to follow what's going on when looking at someone else's code. – [Divick](#) Aug 15, 2013 at 8:35

- 1 It has been almost 4 years since the question was first asked and answered. Has either one of them gained on the other significantly? – [lifebalance](#) Sep 24, 2014 at 10:57
-



10



However it's written in Python which means there's little real support in the way of deployment/packaging, debugging, profilers and other tools that make building and maintaining applications much easier.

Python has:

1. a [great interactive debugger](#), which makes very good use of Python [REPL](#).
2. [easy_install](#) and [virtualenv](#) for dependency management, packaging and deployment.

3. [profiling features](#) comparable to other languages

So IMHO you shouldn't worry about this things, use Python and Django and live happily :-)

Lucky for you, newest version of [Django runs on Jython](#), so you don't need to leave your whole Java ecosystem behind.

Speaking of frameworks, I evaluated this year:

1. [Pylons](#) (Python)
2. [webpy](#) (Python)
3. [Symfony](#) (PHP)
4. [CakePHP](#) (PHP)

None of this frameworks comes close to the power of Django or Ruby on Rails. Based on my colleague opinion I could recommend you [kohana](#) framework. The downside is, it's written in PHP and, as far as I know, PHP doesn't have superb tools for debugging, profiling and packaging of apps.

Edit: Here is a very good [article about packaging and deployment of Python apps](#) (specifically Django apps). It's a hot topic in Django community now.

Share Improve this answer

edited Sep 16, 2008 at 23:35

Follow

answered Sep 16, 2008 at 22:02



zuber

3,487 ● 2 ● 26 ● 19

The link to the article about packaging and deployment doesn't work anymore (page gives 404), does anyone know where it went? – [Epcylon](#) Feb 14, 2010 at 18:38

Although I agree that Python/Django is heads and tails better than any PHP framework out there, the simple fact is that PHP has much greater adoption and mindshare. Unless you plan on being the sole developer for some toy project, PHP is the only way to go. It's professional suicide to depend on Django, which has few, if any, job opportunities, and fewer still developers for hire. If you don't believe me, look in any job list. The difference in opportunities is several orders of magnitude. – [Cerin](#) May 27, 2010 at 14:16 ✎

@Chris S - I'm a fairly successful developer working solely with Django. I can assure you that depending on Django is NOT a professional suicide. – [zuber](#) Jun 14, 2010 at 10:37

Granted, there are some niches where it might work. There seem to be a lot of opportunities if you live in a metro area of California or New York, or work as a freelancer. Everyone else isn't so lucky. – [Cerin](#) Jun 14, 2010 at 13:03

Disagree Symfony2 is essentially a port in terms of architecture and principles to Spring MVC. So its very close to Grails in that sense too. If you are looking for a solid PHP framework forget Kohana, Symfony2 is the way to go. – [Feras](#) Mar 27, 2013 at 17:03



10

The statement that *grails deletes the database on start-up* is completely wrong. It's behavior on start-up is completely configurable and easy to configure. I generally



use create-drop when running an app in dev mode. I use update when I run in test and production.



I also love the bootstrap processing that lets me pre-configure test users, data, etc by environment in Grails.

I'd love to see someone who has really built and deployed some commercial projects comment on the pros / cons. Be a really interesting read.

Share Improve this answer

edited Jan 2, 2013 at 13:34

Follow

community wiki

2 revs, 2 users 89%

Brian Hibbert



Grails.

10

Grails just looks like Rails (Ruby),but it uses groovy which is simpler than java. It uses java technology and you can use any java lib without any trouble.



I also choose Grails over simplicity and there are lots of java lib (such as jasper report, jawr etc) and I am glad that now they join with SpringSource which makes their base solid.

Share Improve this answer

edited Sep 19, 2016 at 8:18

Follow

It's one of the most powerful things in the JEE environment I know! I know a lot business applications which really running on Grails. So, ++ for Grails. – [Martin K.](#) May 21, 2009 at 22:10

- 2 Groovy is not "Simpler" than Java. Such a statement means that you just haven't dug far enough in the language to realize its full potential and complexity. – [Daniel Honig](#) Jun 19, 2009 at 19:09
-

Grails is neat, sadly the documentation is often either missing or just plain wrong. Plus the error messages that get generated when things go wrong are pretty worthless. However, it's got good potential once it matures and gets a wider community. – [TM.](#) Jan 4, 2010 at 5:40



3



I have two friends who originally started writing an application using Ruby on Rails, but ran into a number of issues and limitations. After about 8 weeks of working on it, they decided to investigate other alternatives.

They settled on the [Catalyst Framework](#), and Perl. That was about 4 months ago now, and they've repeatedly talked about how much better the application is going, and how much more flexibility they have.

With Perl, you have all of CPAN available to you, along with the large quantity of tools included. I'd suggest taking a look at it, at least.

Share Improve this answer

answered Sep 16, 2008 at 23:45

Follow



Christopher Cashell

800 ● 3 ● 11 ● 22

How often is Catalyst used in industry? – [svlada](#) Mar 16, 2012 at 12:20

- 1 I would say it depends very heavily on what industry you're looking for. Most of the bigger companies are still pretty heavily Java oriented for web stuff. In smaller companies and the start-up world, you find a lot more Perl, Python, and Ruby in general. I can say I saw a job posting for Catalyst show up on my local Linux User Group mailing list a few weeks ago, but I haven't kept that close an eye on such things lately.
– [Christopher Cashell](#) Mar 20, 2012 at 21:23
-



The "good deployment" issue -- for Python -- doesn't have the Deep Significance that it has for Java.

3



Python deployment for Django is basically "move the files". You can run straight out of the subversion trunk directory if you want to.



You can, without breaking much of a sweat, using the Python [distutils](#) and build yourself a distribution kit that puts your Django apps into Python's site-packages. I'm not a big fan of it, but it's really easy to do.

Since my stuff runs in Linux, I have simple "install.py" scripts that move stuff out of the Subversion directories into `/opt/this` and `/opt/that` directories. I use an

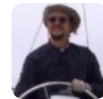
explicit path settings in my Apache configuration to name those directories where the applications live.

Patching can be done by editing the files in place. (A bad policy.) I prefer to edit in the SVN location and rerun my little install to be sure I actually have all the files under control.

Share Improve this answer

answered Sep 18, 2008 at 22:47

Follow



S.Lott

391k ● 82 ● 517 ● 788



1



cakephp.org

Cakephp is really good, really close to ruby on rails (1.2). It is in php, works very well on shared hosts and is easy to implement.

The only downside is that the documentation is somewhat lacking, but you quickly get it and quickly start doing cool stuff.

I totally recommend cakephp.

Share Improve this answer

answered Sep 16, 2008 at 19:49

Follow



Trausti Thor

3,774 ● 35 ● 43

-
- 1 I heard that Symfony is much better for big projects! symfony-project.org Do you have any experience with it? – Martin K.
May 21, 2009 at 22:12
-

I did have a look at symfony but it did not catch me, and I am still with Cakephp and doubt I will change that unless moving to Ruby on Rails – [Trausti Thor](#) Jun 9, 2009 at 14:56

Symfony 2 is much better than previous version you should give it a try [symfony.com](#) – [svlada](#) Apr 28, 2012 at 6:53



1

Personally I made some rather big projects with Django, but I can compare only with said "monstrosities" (Spring, EJB) and really low-level stuff like Twisted.



Web frameworks using interpreted languages are mostly in its infancy and all of them (actively maintained, that is) are getting better with every day.



Share Improve this answer

answered Sep 17, 2008 at 10:12

Follow



[zgoda](#)

12.9k ● 4 ● 39 ● 46



1

By "good deployment" are you comparing it with Java's EAR files, which allow you to deploy web applications by uploading a single file to a J2EE server? (And, to a lesser extent, WAR files; EAR files can have WAR files for dependent projects)



I don't think Django or Rails have gotten quite to that point yet, but I could be wrong... zuber pointed out an article with more details on the Python side.

[Capistrano](#) may help out on the Ruby side.

Unfortunately, I haven't really worked with either Python or Ruby that much, so I can't help out on profilers or debuggers.

[Share](#) [Improve this answer](#)

answered Sep 17, 2008 at 13:19

[Follow](#)



Powerlord

88.7k ● 17 ● 129 ● 177
