

Report about Play! Framework



Play! Is an open source java web application framework initially released in 2007. Play! Is aimed at focusing on developer productivity and making working on a web application easier by using hot code reloading and display of errors in the browser. Play is written in Java¹ and Python¹. Its module is written in Scala¹ and templates are written in Groovy¹. Play follows the model-view-controller architectural pattern (MVC)², and allows the developer to follow REST³ approach which increases its efficiency.

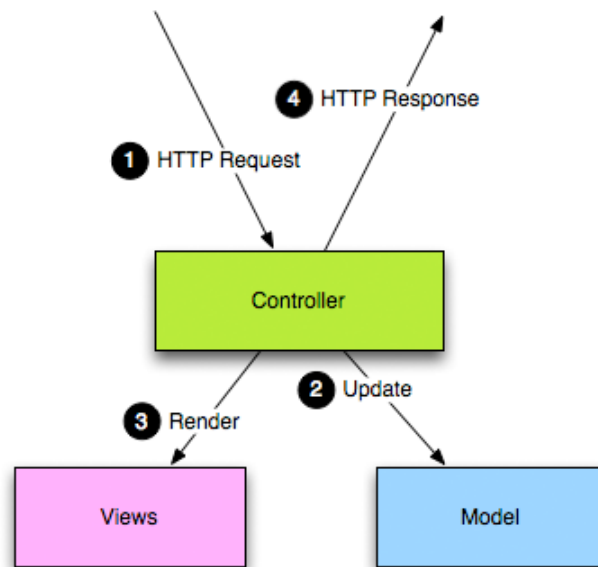


Diagram Illustrating Model-View-Controller pattern

Learning Curve

The first thing people consider when choosing a framework to work in is the learning curve i.e.: how hard it is to learn the framework and get used to it. The learning curve on Play framework is very shallow. It is very easy to get used to and takes little time to learn. People's reviews about Play show that Play has the easiest learning curve amongst other Java web frameworks, and that any user with little Java experience can get used to it really quickly due to its simplicity and its focus on developer productivity.

1. Java, Python, Groovy and Scala are Programming Languages.
2. MVC is a software engineering architectural pattern separating domain logic from the user interface.
3. REST is a style of software architecture following a set of constraints.

Community

Another point to take into consideration while choosing the framework is the support you can get after installing the framework. This support can come from the framework's community and other users or developers working with the framework. The Play framework's community includes a [google group](#) with over 4800 members subscribed. It also includes the framework [blog](#) and users on [stack overflow](#) website. The community members on the google group are very active and have a very quick response rate. Tens of questions are asked each day on the group and you will usually receive a response or more on the same day. You can also get support for older versions of play and newer release candidates (Play! 2.0). Therefore it makes it easier to fix bugs and solve problems you face while working in the framework.

Configuration:

It is certain that the configuration of any application affects its popularity and usability among users. Setting-up and configuring a framework is also a main point to take under consideration when choosing a framework. This point is as important to the developers as the language of the framework or how good is the documentation, because such matter can cause troubles when it is needed to change or improve equipments or to setup new ones. However, Play is designed in a smart way that typing the command *run* does everything and starts the server and the web app, it is as simple as these three letters. Also because it is a stateless framework a live and running project can be simply updated with a newer version by just simple steps. This puts Play way ahead of its competitors and gives the pleasure to any developer using it. Moreover, configuring the application its self is also pretty clear and simple. Other setups are needed also like databases or SSL or even the mail do get complicated in frameworks in general. However the documentation contains explanation to everything and makes it looks simple enough despite of the complications of such settings.

Compatibility

Play! Framework is impressive in many ways that it can be listed with the few popular and powerful Java frameworks out there. Such frameworks usually get the support from different third party developers, and it is the same case with Play. The list of modules that Play supports is eye popping. These modules that are created by many developers who are proving that Play can do a lot are in fact make Play much easier and faster to develop with. Having such feature in a framework assures any developer that this framework is going to stay for a good period of time with the required support and advancement. This does not stop just here, Play does support many other integrations and is also supported with popular services. Google App Engine supports and hosts Play web apps. It supports Play with a handful of features,

a list is provided in Play website. With Play providing dynamic environment where editing the code and then refreshing the page is enough without further complications, Java Servlet does not have a place with Play. Such an important and widely spread API among other frameworks similar to Play had to be dropped to maintain this dynamic environment. According to Guillaume Bort (Play's creator and lead developer), it had to be done to maintain scalability and speed. But as usual there are several replacements and some ways to make Play works with Java Servlet.

Developing environment is crucial matter when it comes to start working with a new framework. Wasting time and effort in a new framework is a concern to most developers. But it is not the case with this framework. With a single command line you are good to go starting building your master piece with Play on Eclipse IDE, the most popular IDE and a good friend for developers. Eclipse works perfectly with all its properties which makes a new Play developer feels like home. Other IDEs are also compatible with Play such as NetBeans and famous text editors like Textmate do also work perfectly with Play and everything is explained in the website on how to setup and configure step by step in a clean and simple terms.

Hosting a Play app can be as simple as working with it. The website provides two Hosts offering really good plans with acceptable prices to host and support projects developed by Play. The hosting planes come with extra services to make it easier to add or delete any code while the app is live and running. Not to forget that Play works like a charm on MAC OS, WIN OS and Linux OS, with the easy and fast setup and configuration.

Best Practices:

After a deep search for the best practices of PLAY! framework, What was common is that nobody agreed for a certain application that was the best thing that play could develop, but what was also common is that most of those who developed with PLAY! Agreed that "The Play framework makes it easier to build Web applications with Java" and the points below are what makes PLAY! Better than other frameworks.

- Edit your Java files, save, refresh your browser and see the results immediately! No need to compile, deploy or restart the server.
- Play is a real "Share nothing" system. Ready for REST, it is easily scaled by running multiple instances of the same application on several servers.
- A clean template system based on Groovy as an expression language. It provides template inheritance, includes and tags.
- When an error occurs, play shows you the source code and the exact line containing the problem. Even in templates.

- All you need to create a cool web application Provides integration with Hibernate, OpenID, Memcached... And a plugin system.
- Pure Java Code with Java, use any Java library and develop with your preferred IDE. Integrates nicely with eclipse or netbeans.
- Really fast Starts fast and runs fast!

Authentication, authorization and security:

The Play framework is designed with security in mind. And [this link](#) from play's manual shows how to handle security issues like dealing with sessions and protecting the user's private data, also dealing cross-site scripting to protect your site from being exposed to attacks like redirecting visitors to pages rather than your page if attackers were able to access your html, also dealing with SQL Injection to protect your database from attackers that might execute queries not intended by the developer which would lead to expose critical data or destroy others, and finally dealing with cross-site request forgery which is simply tricking the user to click on a certain link or app that they authenticated and then inherits their identity. Furthermore the security module of PLAY! is customizable, you can edit it the way you want.

Having an Administration area authentication will be needed and Play provides a module for it, like for instance if trying to access any admin action you will be asked for the log in info, Not only that but also, Play provides developers with the ability to customize the authentication process. By creating our own version of a certain class you will be able to specify exactly how users should be authenticated. And every single detail of how to code this is mentioned in [this link](#) of Play's manual either about the default authentication process or creating a custom one.

The security module not only provides authentication but authorization management too. It's called profiles and it allows you to give the users several roles depending on the desired design. Again every detail related to the coding is mentioned in their manual either for using the predefined module or the customizable one.

Descriptive Errors

During the stage of coding a project, developers face a lot of difficulties that can waste a lot of time and effort. It is important to make everything organized and

maintained to make sure that the code is usable. One of the obstacles that faces developers during coding is errors. Play is trying to make building a web app as simple as possible, and when it comes to errors Play has made it easy to handle them. In Eclipse the IDE, errors and exceptions are presented the old fashioned way which most of Eclipse users are familiar and used to. Also in-browser errors are presented in a well formatted page where it is easy to figure out the error and fix it in minimum time.

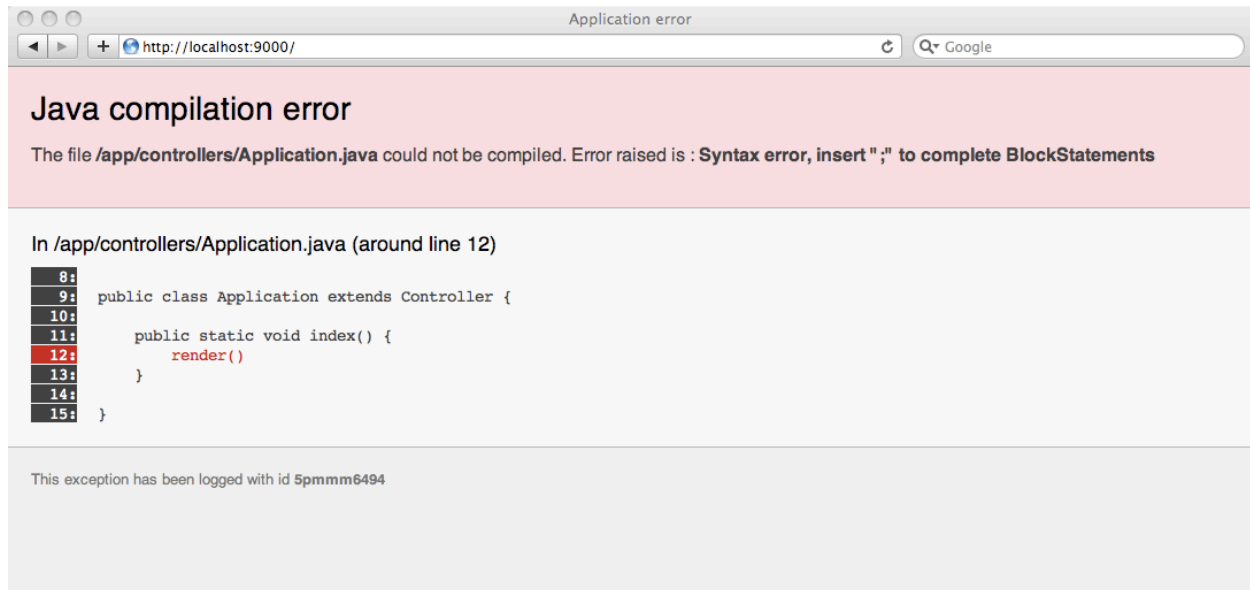
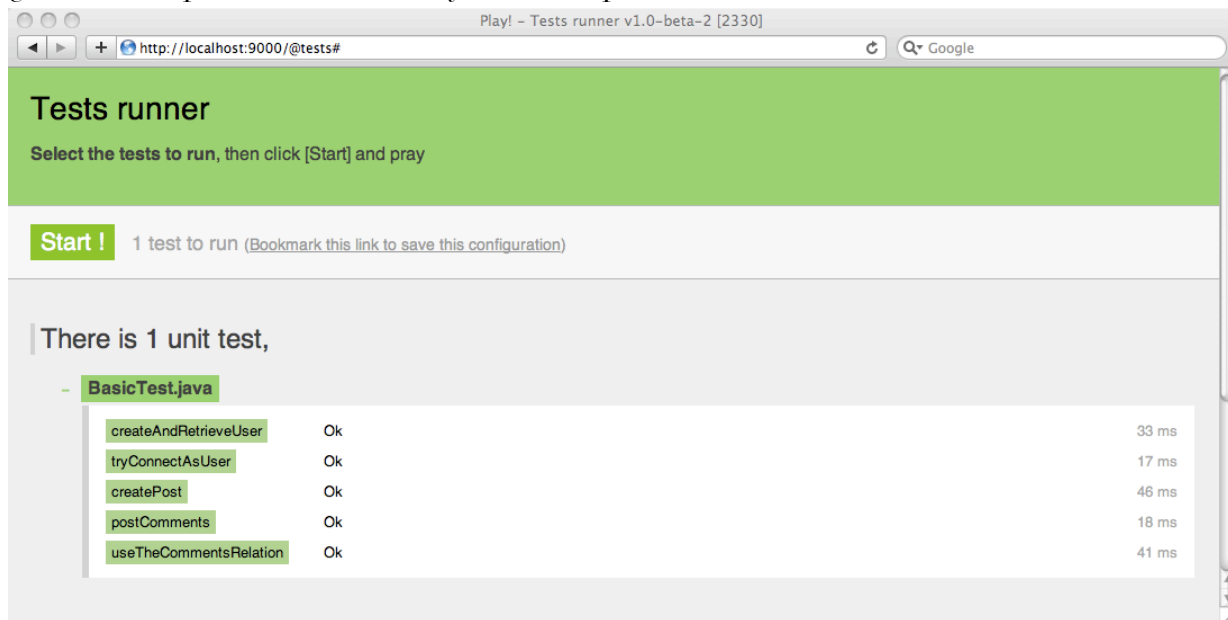


Diagram illustrating how errors are shown in Play!

As Play has proven that it is a high class framework that can be employed to build a complex and powerful projects, it has provided every main tool a developer need to make it easy and time efficient to build such projects. It is important to any experienced developer to make a usable code which can be rewritten and improved in no time. Making test cases before starting the stage of coding is one important step to such experienced developers.

Play provides an amazing tool called Selenium where it is possible to write test cases in a simple format similar to HTML and make several scenarios to make sure that what will come out will be a perfect and professional piece of code. This is what

good developers need, and Play makes it possible.



Documentation

Play! is well documented and every single detail is stated briefly inside their [manual](#) which is provided on their official web site, as well as a [cheat sheet](#) which is like java docs in a way. Also to mention that I made a lot of researches and non stated that there were issues regarding documentation.

Scalability

Yes, Play applications can handle very high traffic since it is easy to balance the load between several servers.

Popularity

Yes, Play is Static typed which means a superior error report checking, code refactoring and debugging it simplifies the development phase with its runtime ability to parse java files directly. It is simpler than most frameworks and it proves to be better than Grails, Tapestry and Wicket.

Websites developed using Play! :

<http://www.masterbranch.com>

<http://www.trendiction.de>

<http://www.plancruncher.com>

<http://www.jigsawbox.com>

<http://www.gendi.fr>

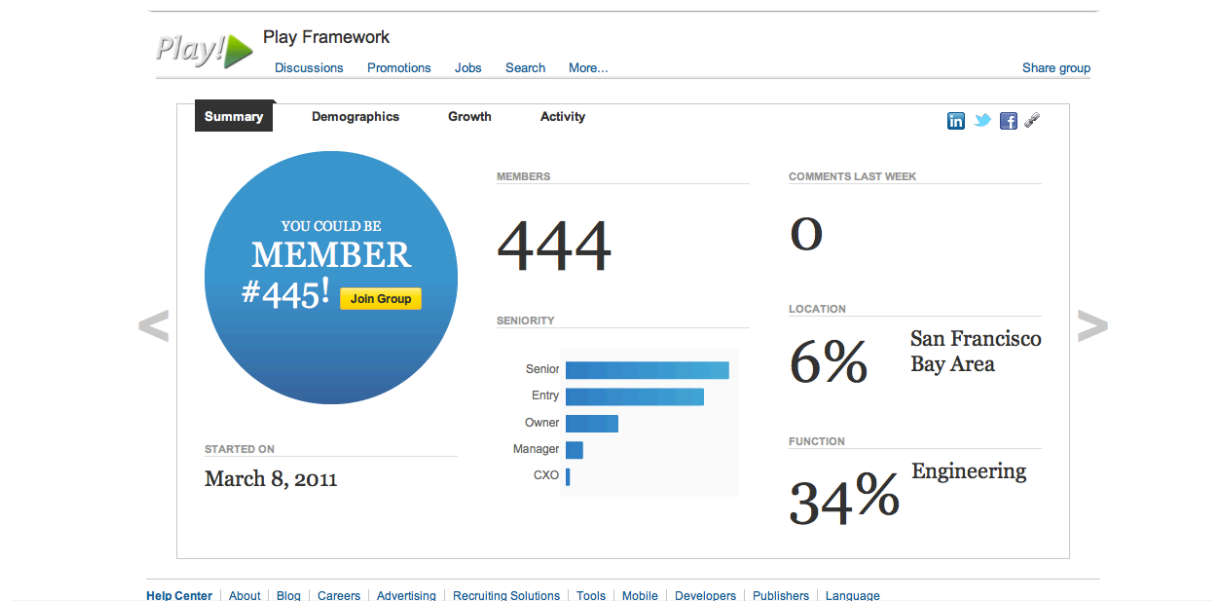
<http://www.zibbet.com>

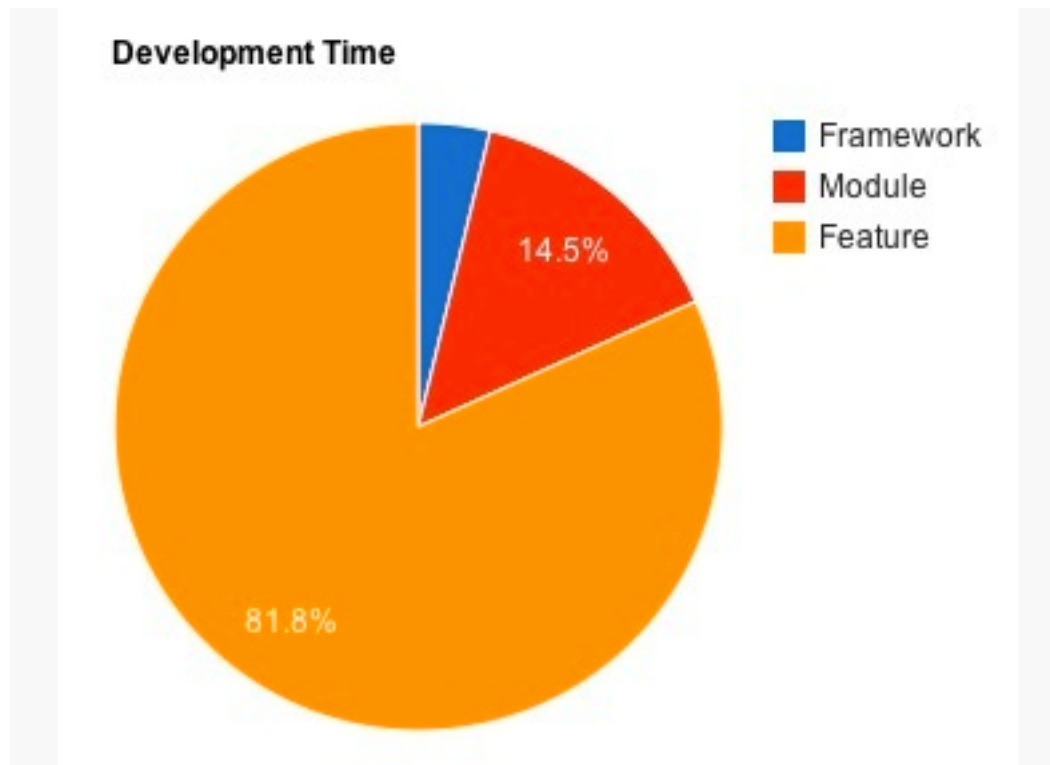
<http://www.quora.com>

Active Releases

The latest released version is 1.2.4

Statistics





References

http://groups.google.com/group/play-framework/browse_thread/thread/8d7250dfe856014b

<http://stackoverflow.com/questions/1597086/any-experience-with-play-java-web-development-framework>

<http://stackoverflow.com/questions/2646681/play-framework-real-world-production-experiences>

http://groups.google.com/group/play-framework/browse_thread/thread/24aab4dcc64d95f3

<http://news.ycombinator.com/item?id=898738>

<http://www.playframework.org/download>