Philip Wernersbach

29 April 2015

Software Maintenance

Play Framework Explanation

The Play framework is a widely used Java framework for creating web applications. We chose to use it because it enforces good and well known design patterns and it allows us to focus on creating the application instead of the infrastructure for it. The Play framework follows the Model-View-Controller design pattern, or MVC for short. A controller handles all of the computation for the web application and facilitates retrieving information from models and rendering pages via views. The Play framework sends all HTTP requests to the controller, which then communicates with the database via the models, does some computation, and sends the user back a web page via the views. In other words, the controller is all of the backend logic. Models are Java classes that model tables in the database. Controllers in the Play framework can use the Models to query the database and interact with it as if database items were plain Java objects, or POJOs. The Play framework uses the Ebean API for its models, which is a standard Java API that all Java developers are familiar with so I will not go over it here. Models in the Play framework also serve as an abstraction for the database tables. Controllers interact with Models via the Ebean API, and then the Play framework translates the Ebean API to calls to the H2 database engine that powers the application. This abstraction is what makes Models so powerful. Views contain all of the page layout and display logic. After a controller queries the database via the Models and performs any necessary computation, it passes variables to the Views, which then compute the resulting HTML, pass it back to the controller, and then the controller passes it back to the client. Views in the Play framework are HTML files that can contain optional command snippets that are interpreted by the Java engine.

Internally the Controllers and Views are singletons, and the Models are the only classes for which objects can be instantiated. This allows us to retain full control of the object lifecycle, and makes the Play framework easier to use.

As I stated earlier, the Play framework allows us to focus on our application and not on the infrastructure behind it. This has allowed us to create a more complex application than we otherwise would have been able to do. We chose the Play framework because we wanted to use Java for the project since we all have experience with it, and because the Play framework is the de facto Java framework for doing this kind of project.