Why PredictionIO Loved Using Play to Build its Popular Machine Learning Server

About PredictionIO
PredictionIO is a popular open source Machine Learning server that developers use to create “predictive” features in web and mobile applications. Showcased as one of the featured machine learning products on GitHub, PredictionIO powers hundreds of applications for its community of more than 5,000 developers. That’s pretty amazing adoption for a product that’s less than two years old.

The Challenge
When the team at PredictionIO first set out to build its platform two years ago, they were intent on using a language and web framework that could best support the ongoing requirements to process huge amounts of data.

“In Big Data, if you want to build distributed algorithms on Hadoop, you write a bunch of map and reduce functions, and you need to be able to run on hundreds of machines,” said PredictionIO’s CEO, Simon Chan.

This requirement to be Reactive -- to scale horizontally and predictably run “computationally expensive” big data and machine learning algorithms across multiple machines -- led Chan’s team to immediately favor JVM languages as it was drawing up its application infrastructure blueprint. PredictionIO chose Scala as its JVM language over Java primarily because of the advantages it brings to functional programming.

“Functional is the native language for data scientists,” explained Chan. “So in the end we chose Scala. Everything is immutable, and we believed in the cleanness of the source code and our ability to maintain it ongoing.”

And once the team had committed to Scala, it was an easy choice to use Play as the framework to build the open source machine learning server. In fact, today when a user downloads the PredictionIO server, the Play Framework is included. The user doesn’t need to worry about scaling their machine learning apps - they just deploy more PredictionIO servers running on Play. PredictionIO can be used on-prem or in the cloud, in AWS marketplace.

PredictionIO’s Five Favorite Things About Working With Play

Fastest Path to Reactive - “When you follow best practices of using Play, you build Reactive Applications,” said Chan.
“We didn’t have to think about how to achieve a non-blocking, stateless architecture, because the Play framework took care of it. And those are really critical criteria of a machine learning server.”

**Simple Learning Curve** - “The learning curve to use Play framework is easy, and it has better documentation than any other Scala-based framework. That’s super important as an open source product, because we want to collaborate with contributors outside of our organization.”

**Built for Scale** - “Building with Play gave us big advantages in terms of easily maintainable code that scales,” said Chan. “It’s stateless, we can deploy on production, and we can bundle it well in PredictionIO for our customers.”

**Non-Blocking** - “The architecture we built on Play is non-blocking,” said Chan, “which is definitely important for us. We process hundreds of messages per minute, per application.”

**The Little Things** - “When you run Play run and modify the source code, it automatically compiles during the dev,” said Chan. “That’s just one of the tiny details that made the development more fun and easy for our team in Play.”

*The Typesafe Reactive Platform is a modern software platform that makes it easy for developers to build scalable software applications. It combines Play Framework, Akka runtime, the Scala programming language, and robust developer tools in a simple package that integrates seamlessly with existing Java infrastructure. Commercial support and maintenance is available for the Typesafe Reactive Platform through the Typesafe Subscription.*