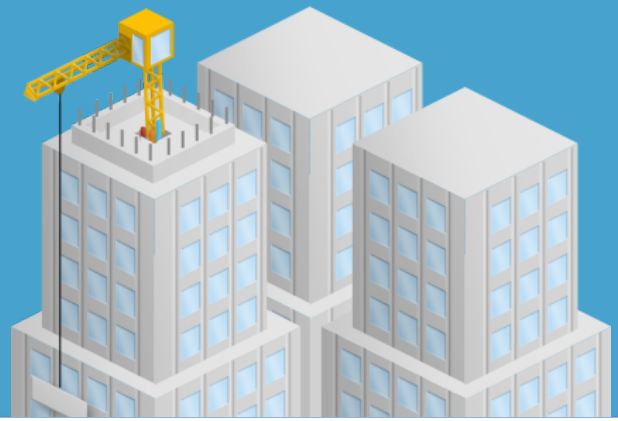# Typesafe

The software platform for applications that scale

The Play Framework at LinkedIn

Recently, LinkedIn expanded their use of the Typesafe Platform to include the Play Framework. Typesafe provided support and consulting services, helping LinkedIn take advantage of Play's extensibility to tune it for their unique needs.

About LinkedIn

LinkedIn is a publicly held company operating the world's largest professional network on the Internet with over 160 million members in over 200 countries and territories.

They are a Typesafe customer and have been using Scala for their core social network graph and other critical components of their product. They've also released open source projects based on Scala, such as Norbert, a cluster management solution, and Kafka, a distributed publish-subscribe system. Scala is one part of Typesafe's flagship product, the Typesafe Platform. Recently, LinkedIn expanded their use of the Typesafe Platform to include the Play web framework.

The Challenge

LinkedIn faced challenges with traditional Java web frameworks. For example:

- It takes too long to make a change, reload and test, wasting developer time
- Stateful, thread-per-request frameworks perform poorly and are difficult to scale, wasting cloud resources

LinkedIn uses some non-Java platforms, such as Ruby on Rails or node.js. However, most of their server-side code is written in Java, so they needed a strong, modern UI framework for Java developers.

The Solution

LinkedIn turned to Play, a component of the Typesafe Platform. Play is a modern framework for the JVM with a focus on developer productivity. It supports both Java and Scala, allowing customers to choose the best language for their situation.

LinkedIn is using the Java API, keeping it simple for their developers to adopt quickly.

"We see Play as a cleanly-architected UI framework with nice APIs that is asynchronous and hot-reloadable from the ground up."

Dean Thompson,
Principal Architect at LinkedIn

Where LinkedIn Deploys Play

LinkedIn uses Play as a front-end server for **UI development**. Examples include:

- Talent Direct, a tool to help customers improve their talent branding and engage the appropriate audience on LinkedIn an application allowing LinkedIn customers to create and vote in polls and surveys

LinkedIn expects to adopt Play more widely over time, as they develop new features on the site.

LinkedIn is also investigating Play as the underlying technology for its **backend services**. They are in the process of moving their entire service architecture to RESTful JSON, and are looking at Play to help with this process.

Rapid Code and Test Cycle

For developers, a killer Play feature is its ability to dynamically recompile the application when they press "reload" in the browser. LinkedIn developers save their changes in their editor, switch to the browser, and press reload. If there are compilation errors, they show up right in the browser. If the build succeeds, the new page shows up.

This simple feature saves countless hours when developers are working to get a web application just right.

To keep it snappy when developers press reload, Play uses sophisticated incremental compilation techniques to rebuild only those classes that have changed.

Extensible to Custom Asset Types

Play's hot reload capabilities can be extended to support new asset types.

LinkedIn has a REST framework using Apache Avro. Java classes are generated from declarative data model and endpoint descriptions.

LinkedIn created an infrastructure for building these REST servers with Play. They also extended Play to support hot-reloading the Avro schemas; after changing a schema, LinkedIn developers just press reload in the browser to regenerate the Java classes and rebuild the app. Any errors appear immediately in the browser.

This saves a developer a couple minutes every time they make a change.

New kinds of template can be added to Play via its module system. LinkedIn implemented a Play module to support Dust templates. As always with Play, Dust templates can be hot-reloaded.

Asynchronous, Stateless Scalability

Play offers two big-picture wins when it comes to scalability.

It supports **asynchronous** requests, meaning it does not require you to tie up a thread while requests are in progress. Requests can be suspended while the app server waits on an external service or database, and then resumed when results are available. This greatly reduces memory requirements for an outstanding request (typically, the thread Platform uses far more memory than the request-servicing code).

As part of the integration process at LinkedIn, Play's Java API for asynchronous request handling was improved and the changes were incorporated into the mainline Play codebase and the Typesafe Platform product.

Play has a **stateless** design, meaning it does not rely on server-side session state. Instead, session state is kept in a cookie and is therefore available to any server node that receives a request.

In a stateless architecture, adding capacity is as easy as adding more application servers. Play gets this right by default and avoids complexity, such as session affinity, that can result in scalability problems.

Flexibility

When evaluating any framework, flexibility is key; because Play has the right architecture out of the box, LinkedIn didn't need to worry about extensive custom modifications that could impede their ability to move to future releases of the framework.

Play has defined extension points so those customizations can be external modules rather than patches to the core framework. Apache Avro schema and Dust template support were implemented as modules.

Finally, in partnership with Typesafe, LinkedIn was able to influence functionality in the framework so that it could be incorporated into the mainstream product.

LinkedIn can be confident that bug fixes and enhancements will be included in future releases of the Typesafe Platform and that they won't be left stranded on an aging version.

Play, from Typesafe

Play has the right technical architecture to meet LinkedIn's needs:

- hot reload
- asynchronous and stateless
- extensible via modules

Typesafe makes this raw technology practical to use in a fast-moving company. The Typesafe Platform offers LinkedIn a long-term supported release that rolls in bugfixes and enhancements, so customers don't end up maintaining a custom fork. Typesafe's extensive product support allowed LinkedIn to quickly add extensions to the Play framework, and coordinate tweaks to the core framework with the upstream maintainers.

Working with Typesafe, LinkedIn is able to save developer time and save server resources, on top of a future-proof, maintainable, and supported platform.

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