Application Development Frameworks

Java EE vs *Spring Boot* & *Spring Cloud*

@benbravo73

“Compare & contrast the use of **Spring** and tell me why **Spring Boot / Cloud** is important”

Brief

Outline

- Choosing a Framework
  - What’s Important?
  - Cloud-native apps
  - Microservices
  - Spring Boot vs Java EE
  - Cloud features
  - Adoption
- Summary

Developers need **frameworks** to help them do their work

They simply want to **Build, Ship** & **Run** their code

Choosing a Framework

What's important?

Build, ***Ship*** *&* ***Run*** *means:*

- ***Develop it***
- ***Test it***
- ***Package it***
- ***Deploy it***
- ***Monitor it***
- ***Scale it***

There's a lot to do, and it's complex work

Good frameworks make it ***fast*** *and* ***easy***

Choosing a Framework

What’s important?

*For businesses, **fast** and **easy** development also means:*

Increased productivity ☑

Better quality ☑

More frequent releases ☑

Greater engagement ☑

Easier experimentation ☑

Cloud is now vital part of application delivery and scalability

Apps need to be "**cloud-native**" by default

Choosing a Framework

Cloud-native apps

Cloud-native *apps are*

- ***Self contained***
- ***Robust***
- ***Self-healing***
- ***Elastic***

Cloud-native apps can be deployed into the cloud and scaled with ease

So how do you build a cloud-native app?

Choosing a Framework

Cloud-native apps

*The current best option for building **cloud-native apps** is **Microservice** architecture*

*Modern frameworks support microservice architecture **out of the box***

But what constitutes a microservice?

Choosing a Framework

Microservices

Microservices *are services that are...*

- ***Small***
- ***Isolated***
- ***Reusable***
- ***Disposable***

Because they're small you may need ***hundreds*** *so they must be easy to make and manage!*

Choosing a Framework

Microservices

*In **Java** there are two competing service development frameworks…*

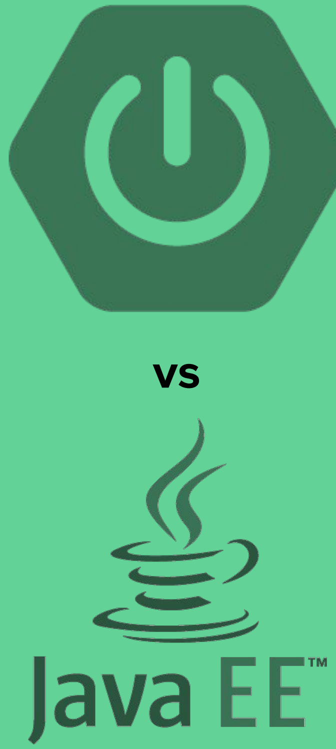
- ***Java Platform Enterprise Edition [JEE]***
- ***Spring Framework [inc Boot & Cloud]***

Java Enterprise Edition is an Application Server specification for full-stack monolithic web-apps

Java EE™

Spring Boot is a Java based framework for building microservices

Let’s compare **Spring Boot** and **Java EE** when developing microservices



Choosing a Framework

Spring Boot vs JEE

***Scenario:** The boss wants a “hello-world” microservice and she want’s it now…*

*Let’s compare the effort required to meet this challenge using **Java EE** vs **Spring Boot***

> Note the number of steps for each and the time required to complete them

Java ENTERPRISE EDITION

1. Choose a JEE Application Server with the right mix of features, standards support, etc. (weeks*)
2. Install the JEE application server on the target machine
3. Add the vendors JEE JARs to your Maven Repository
4. Create a new WAR project (mvn archetype)
5. Add the Jar dependencies you need to your POM (i.e. the vendors JEE jar)
6. Open the project in your IDE
7. Create a new Controller class
8. Add @Path to your class
9. Add @GET to your method
10. Add your servlet-mapping url-pattern instructions to your servlet.xml file
11. Build and package your WAR file
12. Take your WAR file to the application server
13. Install it in the correct folder on the app server
14. Reboot / restart the app server
15. Check the log / console to see if your particular service has started successfully
16. Call localhost to check your message is returned

Congratulations, your service is up!

Elapsed Time 2-4 hours*

Spring Boot

1. Go to **start.spring.io** and download a starter project
2. Open the project in your IDE
3. Create a new Controller class
4. Add @RestController to your class
5. Add @RequestMappping to your method
6. Build your JAR `(mvn package)`
7. Run `java -jar <filename>`
8. Check your message is returned

Congratulations, your service is up!

Elapsed Time **15** mins

Choosing a Framework

Spring Boot vs JEE

Spring Boot:

Builds easier *thanks to pre-configured dependencies and auto-configuration*

Ships easier *thanks to simple JAR packaging*

Runs easier *thanks to embedded servers*

Microservice developers find **Spring Boot** easier, cleaner, and more productive

*“**Spring Boot** blew my mind! I used to spend days getting a service running”*

(Real quote from a Java developer on Reddit)

But what about the **cloud?**

Choosing a Framework

Cloud features

***Microservices** need a supportive environment, one with cloud specific features…*

- **API Gateway**
- **Service Discovery**
- **Load balancing**
- **Externalised Config**
- **Metrics & Monitoring**
- **Persistence & Messaging**
- **Etc…**

Spring Cloud adds several cloud-native features to Spring Boot

You don’t get any of these features with **JEE**

Java ENTERPRISE EDITION

~~Service Discovery~~ ☒

~~External Config~~ ☒

~~API Gateway~~ ☒

~~Circuit Breakers~~ ☒

~~Load Balancers~~ ☒

~~Event Bus~~ ☒

~~Monitoring & Logging~~ ☒

Spring Cloud

Service Discovery ☑

External Config ☑

API Gateway ☑

Circuit Breakers ☑

Load Balancers ☑

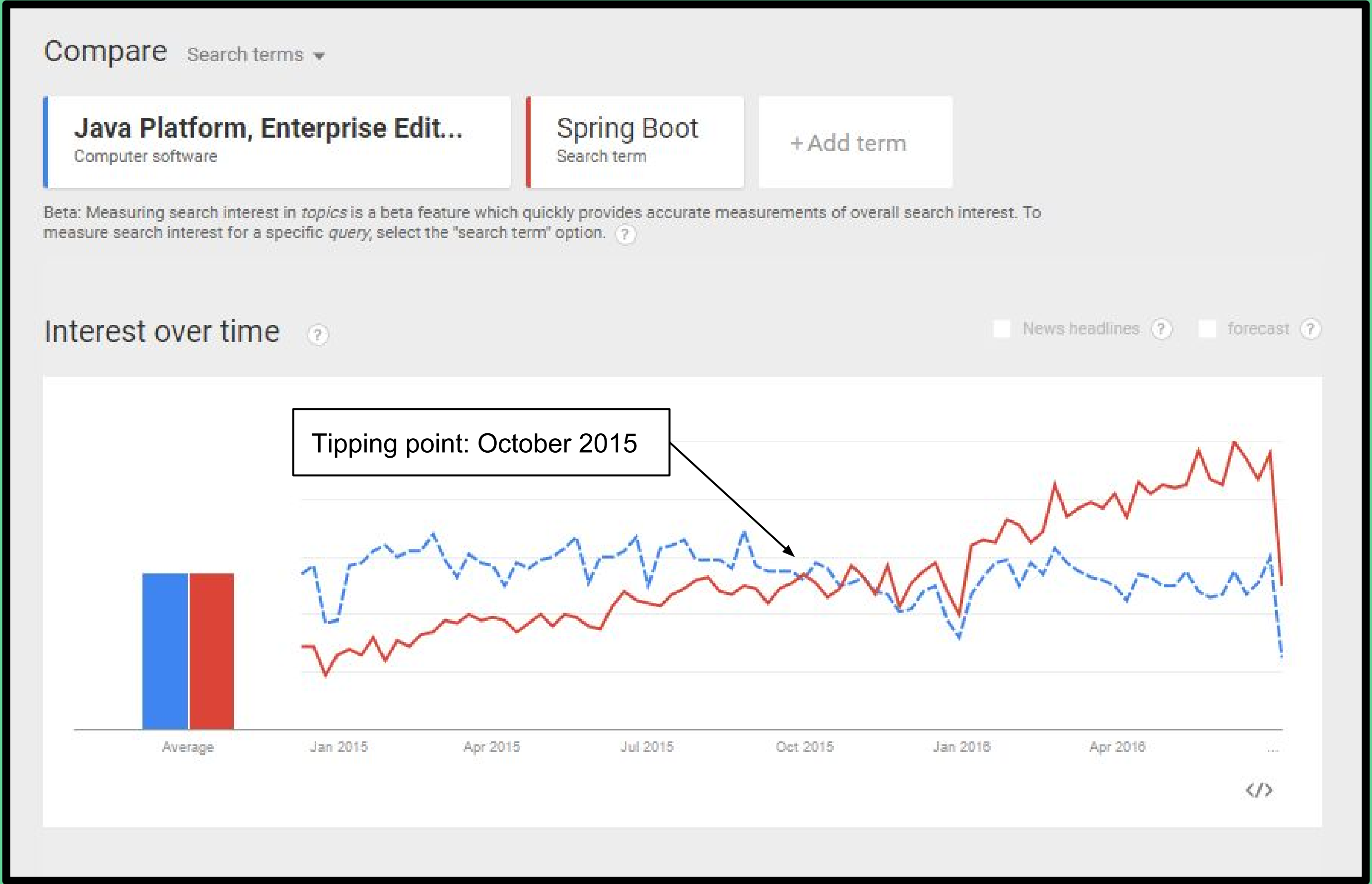
Event Bus ☑

Monitoring & Logging ☑

Spring Boot is **trending!**

Adoption is growing **rapidly** across all sectors and industries

Meanwhile, **JEE** adoption is in decline



Compare Search terms ▾

Java Platform, Enterprise Edit...
Computer software

Spring Boot
Search term

+ Add term

Beta: Measuring search interest in *topics* is a beta feature which quickly provides accurate measurements of overall search interest. To measure search interest for a specific *query*, select the "search term" option. ?

Interest over time ?

□ News headlines ? □ forecast ?

Tipping point: October 2015

Google Trends (JEE vs Spring Boot)

Summary

Java Enterprise Edition

- JEE is ill-equipped for cloud-native microservice development
- JEE is missing several crucial cloud capabilities
- JEE config & packaging slows Build, Ship and Run cycles
- JEE adoption is in decline

Spring Boot & Spring Cloud

- Spring Boot & Spring Cloud accelerate your cloud development
- Building microservices with Spring Boot is easier, cleaner and more productive than in JEE
- Spring Cloud adds powerful cloud-native capabilities
- Adoption of Spring Boot and Spring Cloud is growing rapidly

Like to know more?

@benbravo73

Check out my Spring Boot & Spring Cloud **CQRS Microservice** project...

On Wordpress: https://goo.gl/ijjREg

On GitHub: https://goo.gl/uGGISx

On SlideShare: http://goo.gl/gnoy6j