"Big Picture" for Spring-boot

1. Stakeholders

The stakeholders of Spring Boot are made up of the program's acquirers, assessors, communicators, developers, maintainers, suppliers, support staff, system administrators, testers, and users. Spring Boot's acquirers, assessors, communicators, developers, maintainers, support staffers, and testers are all part of a company known as Pivotal, which assists companies in developing new software. Its suppliers are Java SE and Spring Core since those are the systems that the software is actually built on top of. Since Spring Boot is built on top of Java SE, it has a large user base with Java developers. Examples of companies implementing Spring Boot into their own systems include Netflix, KLM Royal Dutch Airlines, ING, and Philips.

2. Functionality

2.1 Overall Domain

Spring Boot makes it possible to develop Spring based applications that run without needing very much Spring configuration.

2.2 Essential Functional Aspects

- Auto Configuration: Spring Boot allows users to create stand-alone Spring applications without requiring code-generation or XML configuration
- Web Development: Spring Boot simplifies the development of web applications. It contains a starter-web module that can quickly run these applications.

2.3. Essential Non-Functional Aspects

- Embedded features: Spring Boot embeds Tomcat, Jetty, and Undertow within its project to simplify the development of web applications.
- Type-Safe Configuration: Spring Boot ensures that application configuration is type-safe.
- Security: The applications developed using Spring Boot, spring applications and web applications on HTTP servers, are secure by default.

2.4 Uniqueness

- Specifically designed to simplify developing web and soring applications in Java
- Created and maintained by the developers of the Spring framework
- Can learn functionality of Spring Boot from Spring experts and developers

3. Key Developers

- According to Spring's website, its developers focusing specifically on Spring Boot are:
    - Madhura Bhave
    - Brian Clozel

- The top contributors to Spring Boot on GitHub are:
  - Andy Wilkinson- 5,351 commits
  - Stephane Nicoll- 3,780 commits
  - Phil Webb- 2,521 commits
  - Dave Syder- 1,571 commits

4. Open Issues

4.1 Devtools fails to watch for changes during a short period while restarting the application

According to the issue on GitHub,

File watcher is used to monitor ./classPath files, and update the files to the remote endpoint. But in this issue, this feature is not available in the short period after restarting. More specifically, if we decide to add a class to the file and restart at once, or shortly, after the restart, it will lose the new added class.

By adding two time intervals, it could solve this bug to some extend, but could not eliminate this bug.

> spring.devtools.restart.poll-interval
>
> spring.devtools.restart.quiet-period

4.2 Resource from filesystem not found when application is spring-boot packaged

According to the issue[#20111] on GitHub, The filesystem could be compiled under windows system and Linux system but when package the jar and try to run it (**java -jar.jar)**, it could only be compiled under Linux system. The compile error shows below errors.

> "java.lang.IllegalArgumentException: name"

The same issue happens in issue#19098, which could work on MacOS but fail on CentOS. This may be the reason of virtual machine. But developers haven't figured out a practical solution for this.

4.3 CassandraHealthIndicator runs a query that fails on some Consistency Levels

According to issue#17768, his team changed the source code of `CassandraHealthIndicator` . They changed the default READ consistency from `LOCAL_QUORUM` to `THREE` . But after this, spring-boot could not finish query successfully. The basic idea behind this is that spring-boot can't have a high consistent level. In this scenario, we need to create a specific schema to store health query request. It is against the idea of "easier to use".

4.4 Filters are not applied to Actuator endpoints when management port is different

According to issue#16098, originally, spring-boot supports filters for Actuator endpoints, but in this issue, if user wants to customize port number, `management.server.port != server.port`, filter would fail. Based on our assumption, port is often bound to a specific annotation, and developers suggest to manage the port in `@ManagementContextConfiguration`.

4.5 Remote Spring Boot devtools unable to restart application if a dependency has autowire candidates that the main application uses

According to issue#8321, their application is trying to hot restart but if some services are shared with other components, the hot restart would not success. More specifically, in a shared service module, when calling hot restart is triggered, the remote hot reload process happens in the `RestartServer` and it will only reload the classes within the service. The newly generated service could not be wired to a controller, because new service is in `RestartClassLoader` and controller is in `LauchedURLClassLoader`. A better implementation of this hot reload would be restart and reload all the related class in the same time.

Reference

1. https://github.com/spring-projects/spring-boot
2. https://spring.io/projects/spring-boot
3. https://www.quora.com/What-is-spring-boot-and-some-of-its-application
4. https://dzone.com/articles/top-5-spring-boot-features-java-developers-should
5. https://www.javatpoint.com/spring-boot-features
6. https://content.pivotal.io/springone-platform-2017/whats-new-in-spring-boot-2-0-phillip-webb-madhura-bhave
7. https://github.com/spring-projects/spring-boot/graphs/contributors