Overview of Spring & Spring Boot

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

• Understand what Spring & Spring Boot are

Spring Boot
Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

Spring Framework
Provides core support for dependency injection, transaction management, web apps, data access, messaging, and more.

Spring Data
Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.

Spring Cloud
Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.

Spring Cloud Data Flow
Provides an orchestration service for composable data microservice applications on modern runtimes.

Spring Security
Protects your application with comprehensive and extensible authentication and authorization support.

See spring.io/projects
Learning Objectives in this Lesson

• Understand what Spring & Spring Boot are

• Recognize key components in Spring Boot

See [spring.io/projects/spring-boot](http://spring.io/projects/spring-boot)
Overview of Spring & Spring Boot
Overview of Spring & Spring Boot

• Spring is a dependency-injection framework & an inversion of control container for developing web apps on the Java platform

See spring.io
Overview of Spring & Spring Boot

- Dependency injection (DI) enables an object/service to receive other objects/services it depends on

Overview of Spring & Spring Boot

- Dependency injection (DI) enables an object/service to receive other objects/services it depends on.
- DI enables “autowiring” in lieu of manual wiring.

See [www.baeldung.com/spring-autowire](http://www.baeldung.com/spring-autowire)
Overview of Spring & Spring Boot

- Dependency injection (DI) enables an object/service to receive other objects/services it depends on
  - DI enables “autowiring” in lieu of manual wiring
- Separates the concerns of constructing objects & using them, leading to loosely coupled programs

Loose coupling:
1. Less Interdependency
2. Less coordination
3. Less information flow

Tight coupling:
1. More Interdependency
2. More coordination
3. More information flow

See [www.geeksforgeeks.org/coupling-in-java](http://www.geeksforgeeks.org/coupling-in-java)
Overview of Spring & Spring Boot

- With inversion of control (IoC) the framework runs the main execution thread(s)

See en.wikipedia.org/wiki/Inversion_of_control
Overview of Spring & Spring Boot

- With inversion of control (IoC) the framework runs the main execution thread(s)
- Implements the “Hollywood Principle”

Don’t call us, we’ll call you

See [www.dre.vanderbilt.edu/~schmidt/Coursera/articles/hollywood-principle.txt](www.dre.vanderbilt.edu/~schmidt/Coursera/articles/hollywood-principle.txt)
Overview of Spring & Spring Boot

• Spring contains various projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring Boot</td>
<td>Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.</td>
</tr>
<tr>
<td>Spring Framework</td>
<td>Provides core support for dependency injection, transaction management, web apps, data access, messaging, and more.</td>
</tr>
<tr>
<td>Spring Data</td>
<td>Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.</td>
</tr>
<tr>
<td>Spring Cloud</td>
<td>Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.</td>
</tr>
<tr>
<td>Spring Cloud Data Flow</td>
<td>Provides an orchestration service for composable data microservice applications on modern runtimes.</td>
</tr>
<tr>
<td>Spring Security</td>
<td>Protects your application with comprehensive and extensible authentication and authorization support.</td>
</tr>
</tbody>
</table>

See [spring.io/projects](http://spring.io/projects)
Overview of Spring & Spring Boot

- Spring contains various projects
- We focus on Spring Boot 2.0

See www.baeldung.com/new-spring-boot-2
Key Components in Spring Boot
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications.

See [spring.io/projects/spring-boot](http://spring.io/projects/spring-boot)
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

See [en.wikipedia.org/wiki/Client_(computing)](en.wikipedia.org/wiki/Client_(computing))
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

HTTP is a request–response protocol in the client–server computing model

See en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

Maps HTTP requests to endpoint methods that handle these requests

See [spring.io/guides/gs/rest-service](spring.io/guides/gs/rest-service)
Key Components in Spring Boot

• The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

Repository Class Extending CRUD Services

Dependecy Injection

Client

Controller

Service Layer

Model

Provide the “business logic” to operate on data sent to & from the data access layer & the client

See medium.com/stackavenue/why-to-use-service-layer-in-spring-mvc-5f4fc52643c0
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

Controllers & services can be connected to their dependent components automatically

See en.wikipedia.org/wiki/Dependency_injection

See en.wikipedia.org/wiki/Dependency_injection
Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

**Diagram:**

- **Client**
  - HTTPS request
- **Controller**
  - Service Layer
  - Repository Class Extending CRUD Services
- **Service Layer**
- **Model**
- **Database**

**Legend:**

- Dependency Injection
- Provide access to persistent data that is mapped to JPA with model classes

Key Components in Spring Boot

- The Spring Boot project makes it easy to create stand-alone, production-grade Web-based applications

---

These interactions are typically just Java method calls

See blogs.oracle.com/javamagazine/mastering-the-mechanics-of-java-method-invocation
End of Overview of Spring & Spring Boot