Sri Lanka Institute of Information Technology

APPLICATION FRAMEWORKS

SPRING BOOT

LECTURE 07

Faculty of Computing
Department of Computer Science and Software Engineering
Module Code: SE3040

Agenda

1. Maven
2. Spring Boot

MAVEN

- Maven is a popular **build tool** for Java-based projects.
- Maven simplifies the process of managing project dependencies and building applications.
- Maven provides a wide range of plugins for performing common build tasks.
- Maven supports the concept of repositories, which are centralized locations for storing and sharing project artifacts.
- Maven makes it easy to share and reuse code across projects by providing a standardized way to manage dependencies and build settings.

BUILT TOOLS

- Built tools are software programs that automate the process of building and packaging software applications.
- Built tools help developers to manage dependencies, compile code, run tests, and package applications into distributable artifacts.
- Maven is a popular built tool for Java-based projects that simplifies the process of managing project dependencies and building applications.
- Maven uses a declarative XML-based configuration file called pom.xml to manage project dependencies and build settings.

BUILT TOOLS...CNT

- Other popular built tools for Java-based projects include Gradle, Ant, and Ivy, each with their own strengths and weaknesses.

- Choosing the right built tool for a project depends on factors such as project complexity, team preferences, and community support.

SPRING FRAMEWORK

SPRING FRAMEWORK

- Spring is a widely-adopted open-source framework for building enterprise applications
- Spring Boot features and Spring framework offer a robust, lightweight infrastructure for Java applications
- Comprehensive programming and configuration model for web and non-web application parts
- Spring framework provides many APIs to boost developer productivity, including transaction management and integration, data access and security, server-side technology abstraction, etc.
- One of the most versatile and powerful frameworks in Java

SPRING FRAMEWORK...CNT

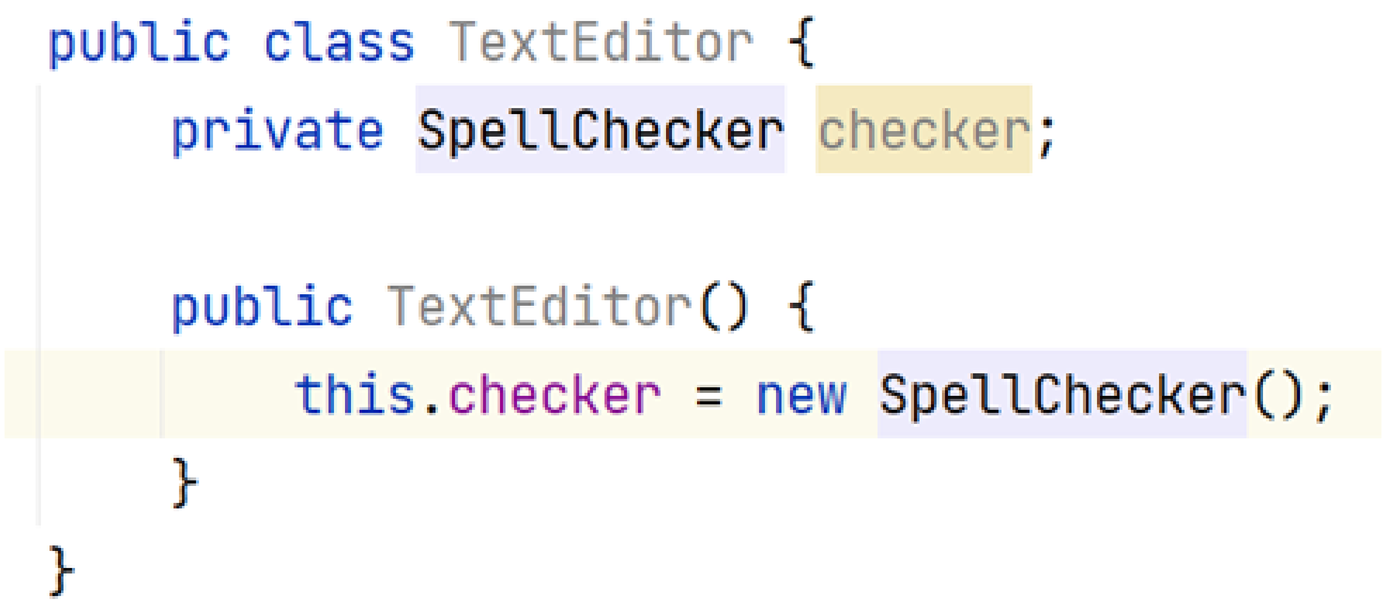
- Focuses on several areas of application development to simplify Java EE development and help developers be more productive
- Introduces a paradigm for building applications with POJOs so that business objects are not tied to any specific framework or runtime environment
- Most famous for its inversion of controller container for dependency injection.

WHAT ARE THE MAIN FEATURES OF SPRING?

- The most fundamental aspect of Spring and Spring Boot is **Dependency Injection (DI)** or **Inversion of Control (IoC)**
- We can create loosely coupled applications that can be easily tested and maintained using these design patterns. The Spring framework also includes several out-of-the-box modules, namely:
  - Spring MVC
  - Spring Security
  - Spring ORM
  - Spring Test
  - Spring AOP
  - Spring Web Flow
  - Spring JDBC.
- These modules make web applications more functional and reduce development time significantly.

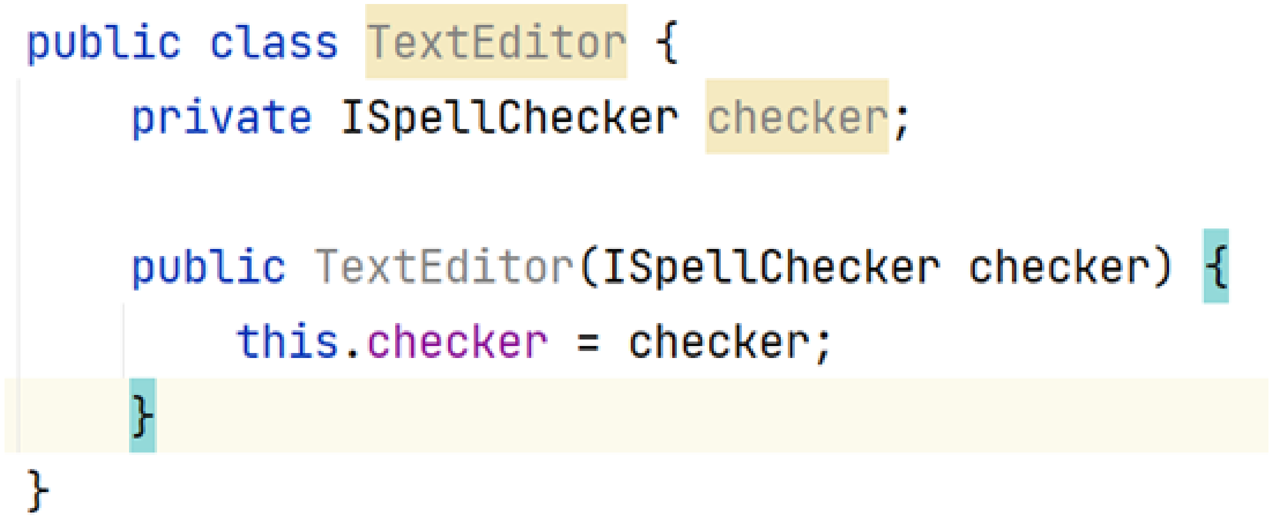
DEPENDENCY INJECTION (DI)

- A type of Inversion of control.
- Passing the dependency at runtime (mostly) into the class without concreate dependencies.
- Resulting context is low coupling between classes.



```java
public class TextEditor {
    private SpellChecker checker;

    public TextEditor() {
        this.checker = new SpellChecker();
    }
}
```



```java
public class TextEditor {
    private ISpellChecker checker;

    public TextEditor(ISpellChecker checker) {
        this.checker = checker;
    }
}
```

DEPENDENCY INJECTION (DI)... CNT

Maven dependency

```
<dependency>
 <groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
 <version>4.0.0.RELEASE</version>
</dependency>
```

Injections

- Setter based - @Autowired on top of the setter
- Constructor based - @Autowired on top of the constructor
- Field based - @Autowired on top of the field (highly discouraged)

DEPENDENCY INJECTION (DI)... CNT

Some more annotations

@Component

- Making class Spring container aware as a Component.

@Service

- Making class Spring container aware as a Service.

@Repository

- Making class Spring container aware as a DAO.

@RestController

- Making class Spring container aware as a REST controller.

@Configurations

- Spring aware configuration class.

@Autowired

SPRING USE CASES:

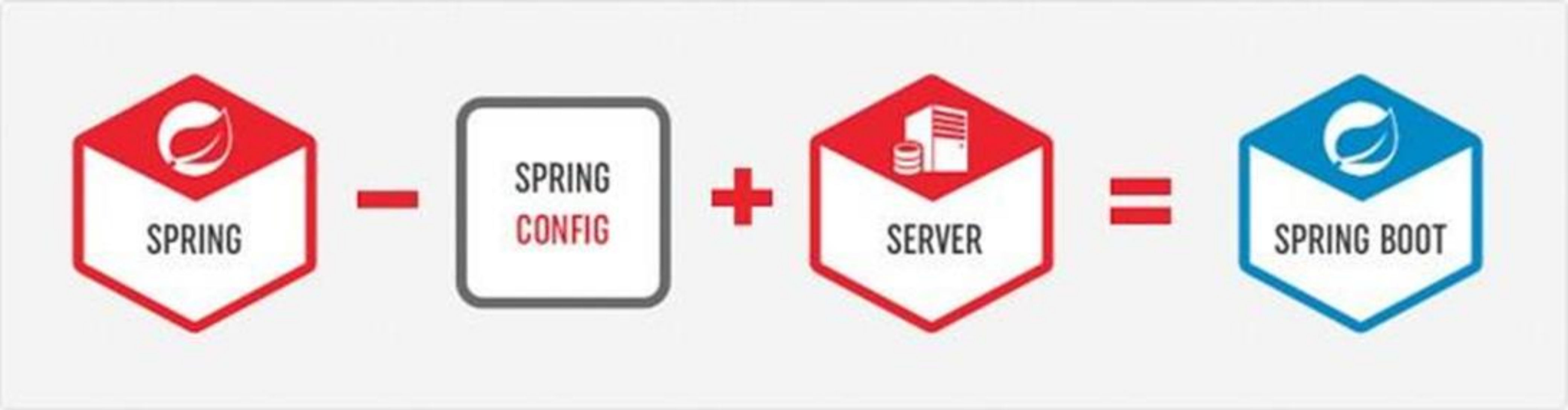
Spring framework can be used for several tasks, including

- Developing serverless applications
- Building scalable microservices
- Securing the server-side of your application
- Asynchronous application development
- Automating tasks by creating batches
- An event-driven architecture

Nice framework but the amount of configuration it has, made it cumbersome to use for rapid application development

SPRING BOOT

- Fully featured robust framework mainly targeted for Microservices application development.
- A solution for cumbersome configuration Spring Framework has.
- Support for microservices.
- Easy integration with multiple other libraries and frameworks (Cloud, Circuit breakers)
- Embedded server for development and deployments.



MAIN FEATURES OF SPRING BOOT?

- Embedded server eliminates the need for complex application development
- Starter dependencies that facilitate building and configuring apps
- Automated Spring configuration
- Metrics, health check, and other reports
- Support for microservices.
- Everything in Spring Boot is pre-configured. We simply need to use the proper configuration to use a specific functionality. If we want to create a REST API, we can use Spring Boot.

MICROSERVICES WITH SPRING BOOT

- Spring Boot is a popular framework that simplifies the development and deployment of microservices
- It provides a suite of tools and features that address the challenges of microservices, including:
  - Embedded web server
  - Auto-configuration
  - Health checks
  - Distributed tracing
  - Service discovery and registration
  - Load balancing
  - Configuration management

THAT'S ALL FOLKS !

ANY QUESTIONS ?