Spring Boot :

Spring Boot is a popular Java framework for building stand-alone, production-grade Spring-based applications. Here's an explanation of key points about Spring Boot, along with code examples for each point:

1. **Spring Boot Overview**:
   - Spring Boot simplifies the development of Spring applications by providing pre-configured templates and sensible default settings.

2. **Creating a Simple Spring Boot Application**:
   - You can create a simple Spring Boot application with a single Java class.

   ```java
   import org.springframework.boot.SpringApplication;
   import org.springframework.boot.autoconfigure.SpringBootApplication;

   @SpringBootApplication
   public class MyApplication {
       public static void main(String[] args) {
           SpringApplication.run(MyApplication.class, args);
       }
   }
   ```

3. **Spring Boot Auto-Configuration**:
   - Spring Boot provides automatic configuration for various components, reducing the need for extensive XML or Java configuration.

4. **Spring Boot Starters**:
   - Starters are pre-configured dependencies that simplify the inclusion of common libraries and frameworks.

   Example in `pom.xml` (Maven):
   ```xml
   <dependencies>
       <dependency>
           <groupId>org.springframework.boot</groupId>
           <artifactId>spring-boot-starter-web</artifactId>
       </dependency>
   </dependencies>
   ```

5. **Spring Boot Web Application**:

- Spring Boot makes it easy to create web applications. Here's a basic REST controller:

```java
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {
    @GetMapping("/hello")
    public String hello() {
        return "Hello, World!";
    }
}
```

6. **Configuration Properties**:
   - Spring Boot allows you to configure your application using `application.properties` or `application.yml`.

   Example (`application.properties`):
   ```properties
   server.port=8080
   ```

7. **Embedded Web Server**:
   - Spring Boot includes an embedded web server (e.g., Tomcat) for running your application.

8. **Spring Boot Actuator**:
   - Actuator provides production-ready features like health checks, metrics, and application info.

9. **Spring Data JPA**:
   - Spring Boot simplifies data access using Spring Data JPA. You can create repositories easily.

10. **Spring Boot Security**:
    - Spring Boot provides security features for authentication and authorization.

    Example:
    ```java
    @Configuration
    public class SecurityConfig extends WebSecurityConfigurerAdapter {
        @Override
        protected void configure(HttpSecurity http) throws Exception {
            http
    ```

```
            .authorizeRequests()
            .antMatchers("/public/**").permitAll()
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll();
    }
}
```

11. **Spring Boot Data REST**:
    - Spring Boot Data REST allows you to expose JPA repositories as RESTful services with minimal code.

12. **Spring Boot Testing**:
    - Spring Boot provides testing utilities to write unit and integration tests easily.

    Example (JUnit test with Spring Boot):
    ```java
    @RunWith(SpringRunner.class)
    @SpringBootTest
    public class MyApplicationTests {
        @Test
        public void contextLoads() {
            // Your test logic here
        }
    }
    ```

13. **Spring Boot DevTools**:
    - DevTools provides automatic application restart and other development tools.

14. **Packaging as a JAR or WAR**:
    - Spring Boot can package your application as a self-contained JAR or a traditional WAR file.

15. **Spring Boot CLI**:
    - Spring Boot CLI allows you to build and run Spring applications using a command-line interface.

Spring Boot simplifies the development of Spring-based applications, making it a popular choice for building Java applications. The framework provides a wide range of features and integrations to help developers create robust and production-ready applications quickly.