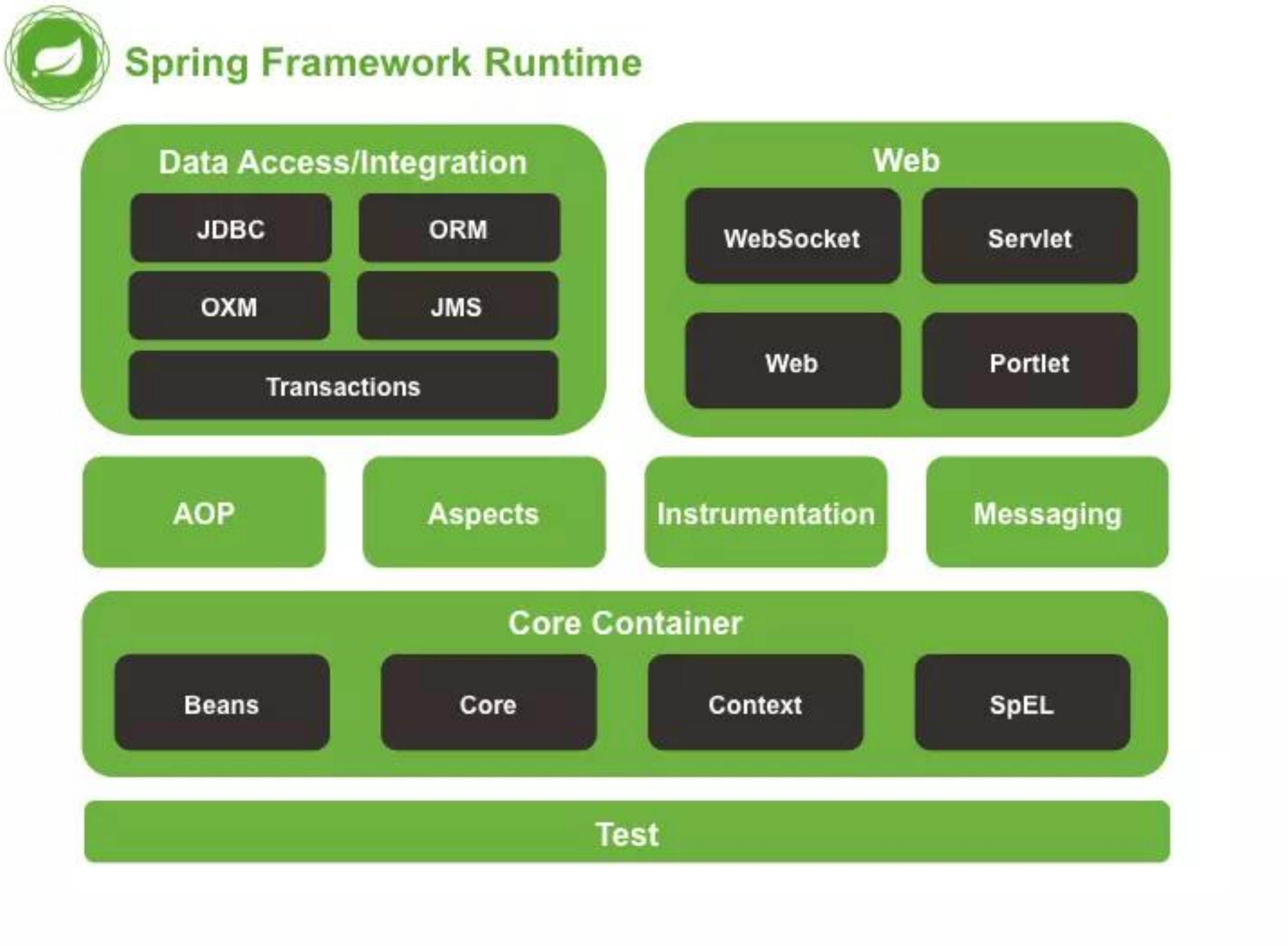
Session 1: Introduction

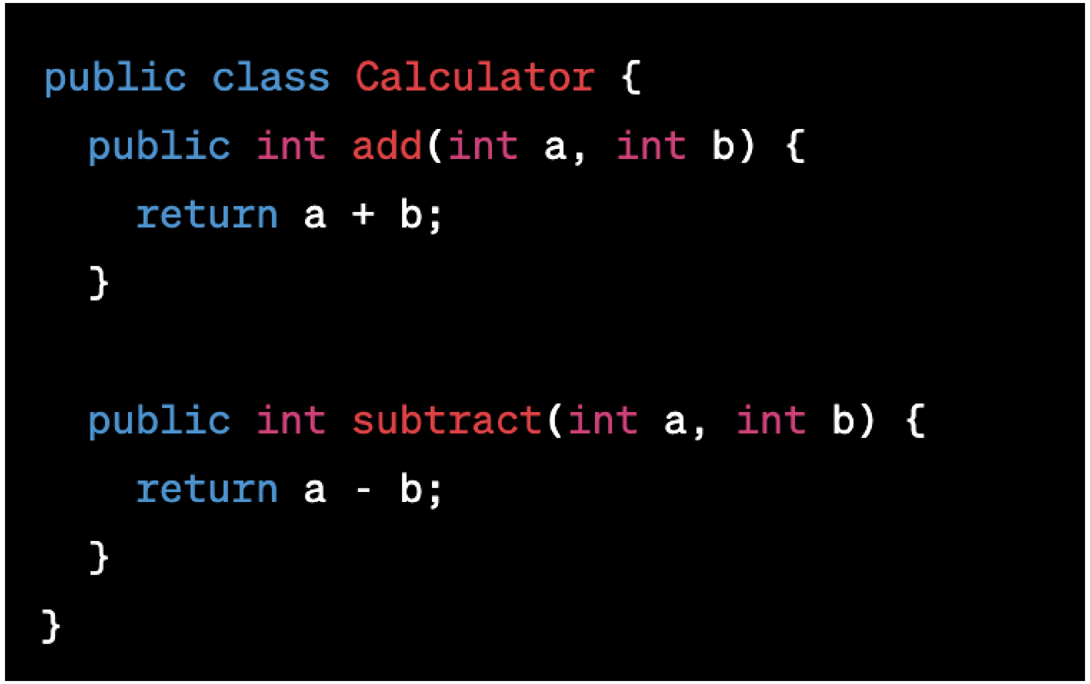
Learning goals

- Spring framework
- Spring boot and why using Spring boot?
- Maven
- Spring Initializr
  - Generate first Spring boot project
  - Run application

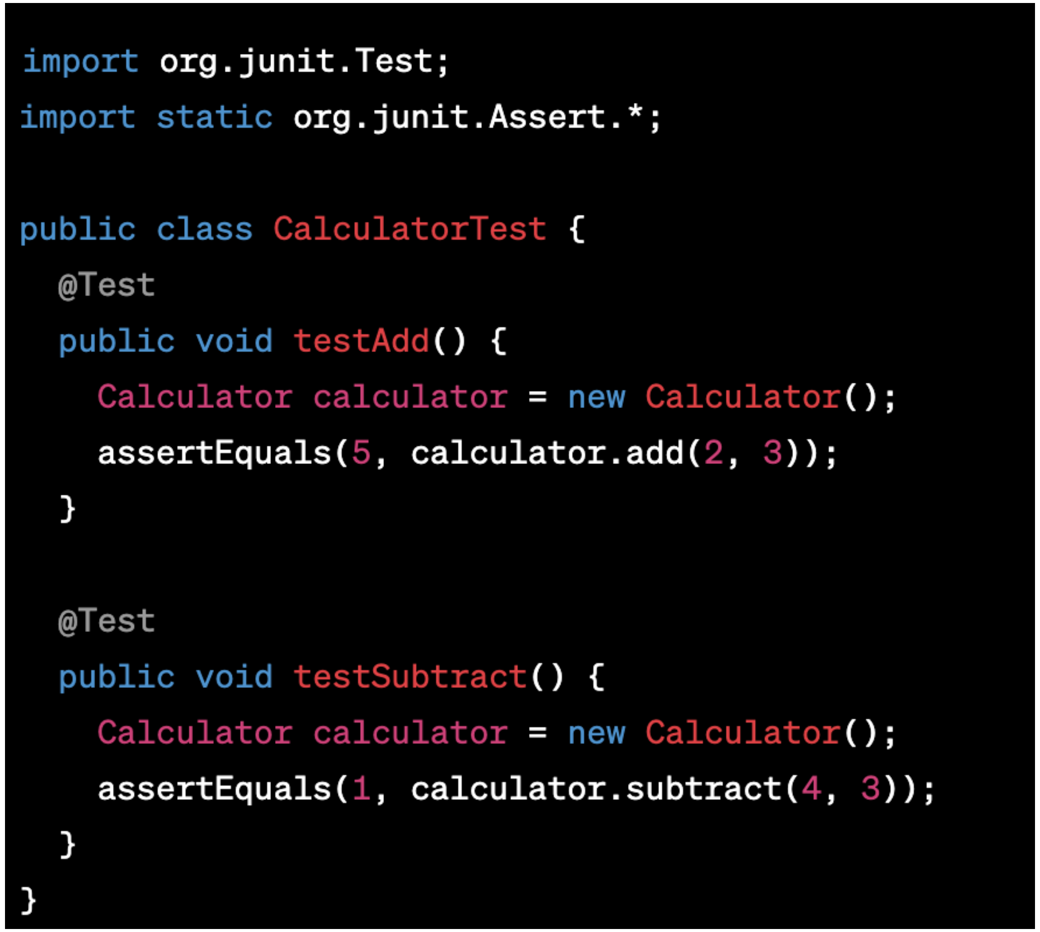
Spring framework



Spring framework - Test

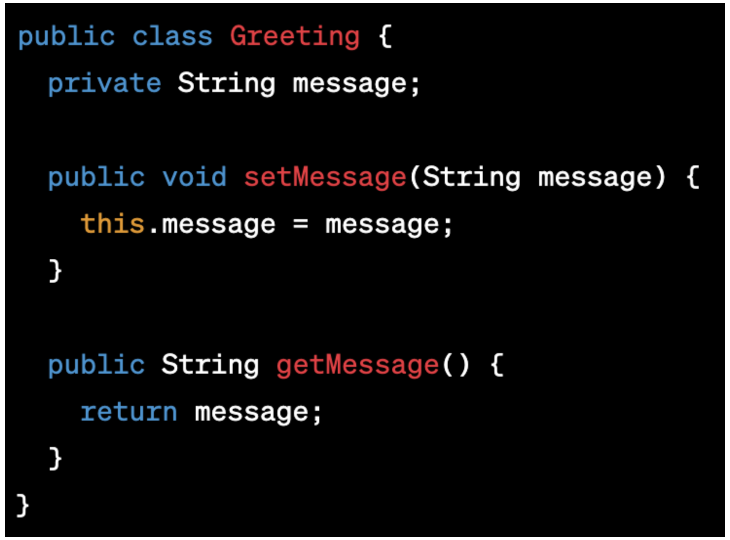


```java
public class Calculator {
  public int add(int a, int b) {
    return a + b;
  }

  public int subtract(int a, int b) {
    return a - b;
  }
}
```

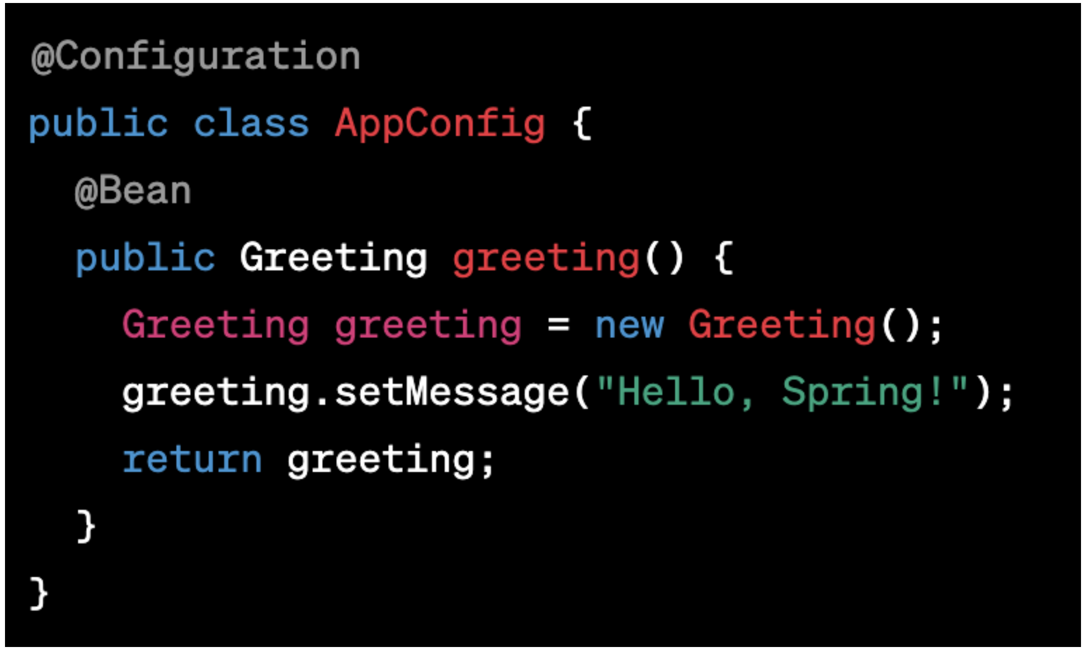


```java
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {
  @Test
  public void testAdd() {
    Calculator calculator = new Calculator();
    assertEquals(5, calculator.add(2, 3));
  }

  @Test
  public void testSubtract() {
    Calculator calculator = new Calculator();
    assertEquals(1, calculator.subtract(4, 3));
  }
}
```

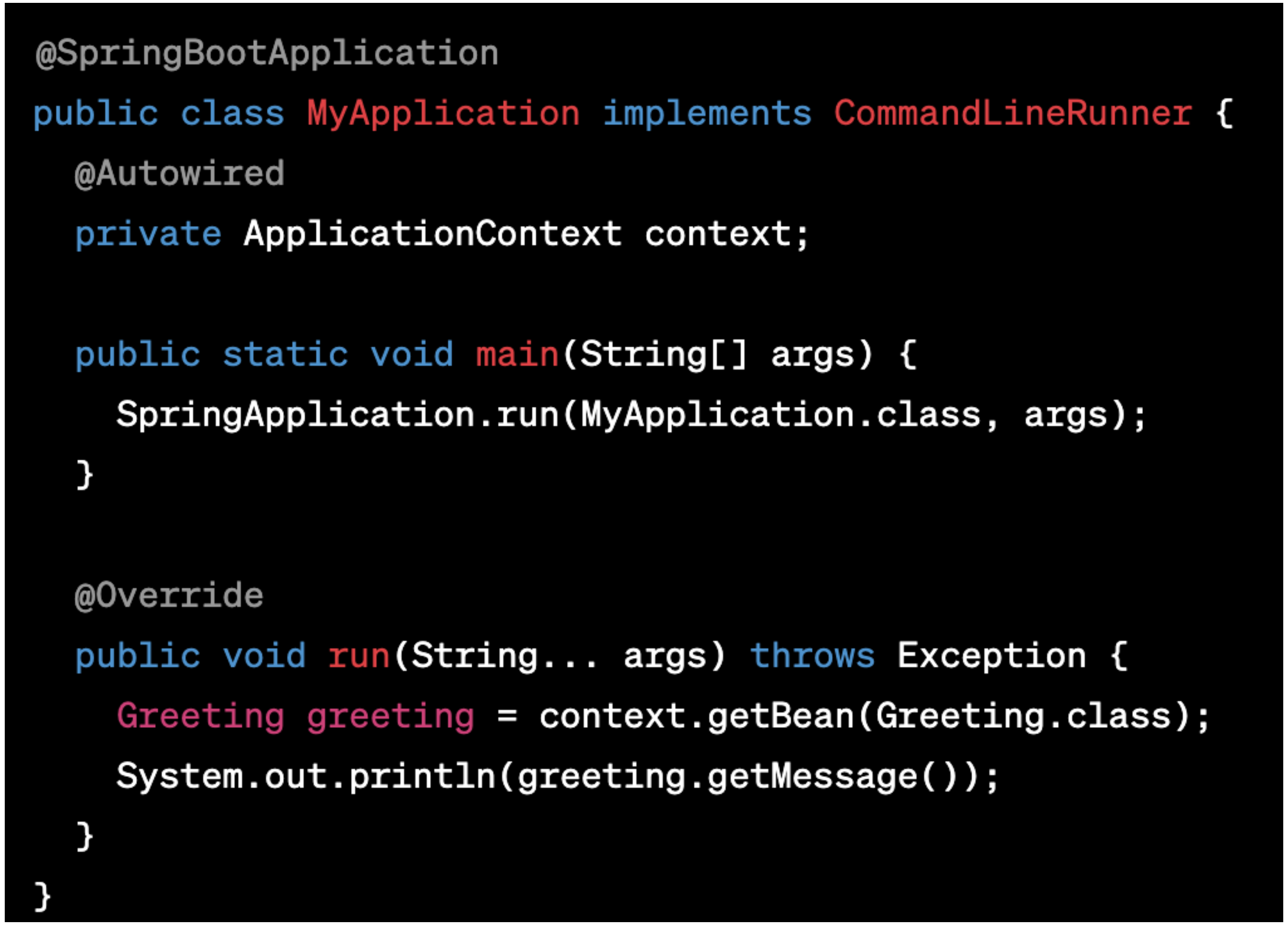
Spring framework – Spring core



```java
public class Greeting {
  private String message;

  public void setMessage(String message) {
    this.message = message;
  }

  public String getMessage() {
    return message;
  }
}
```



```java
@Configuration
public class AppConfig {
  @Bean
  public Greeting greeting() {
    Greeting greeting = new Greeting();
    greeting.setMessage("Hello, Spring!");
    return greeting;
  }
}
```

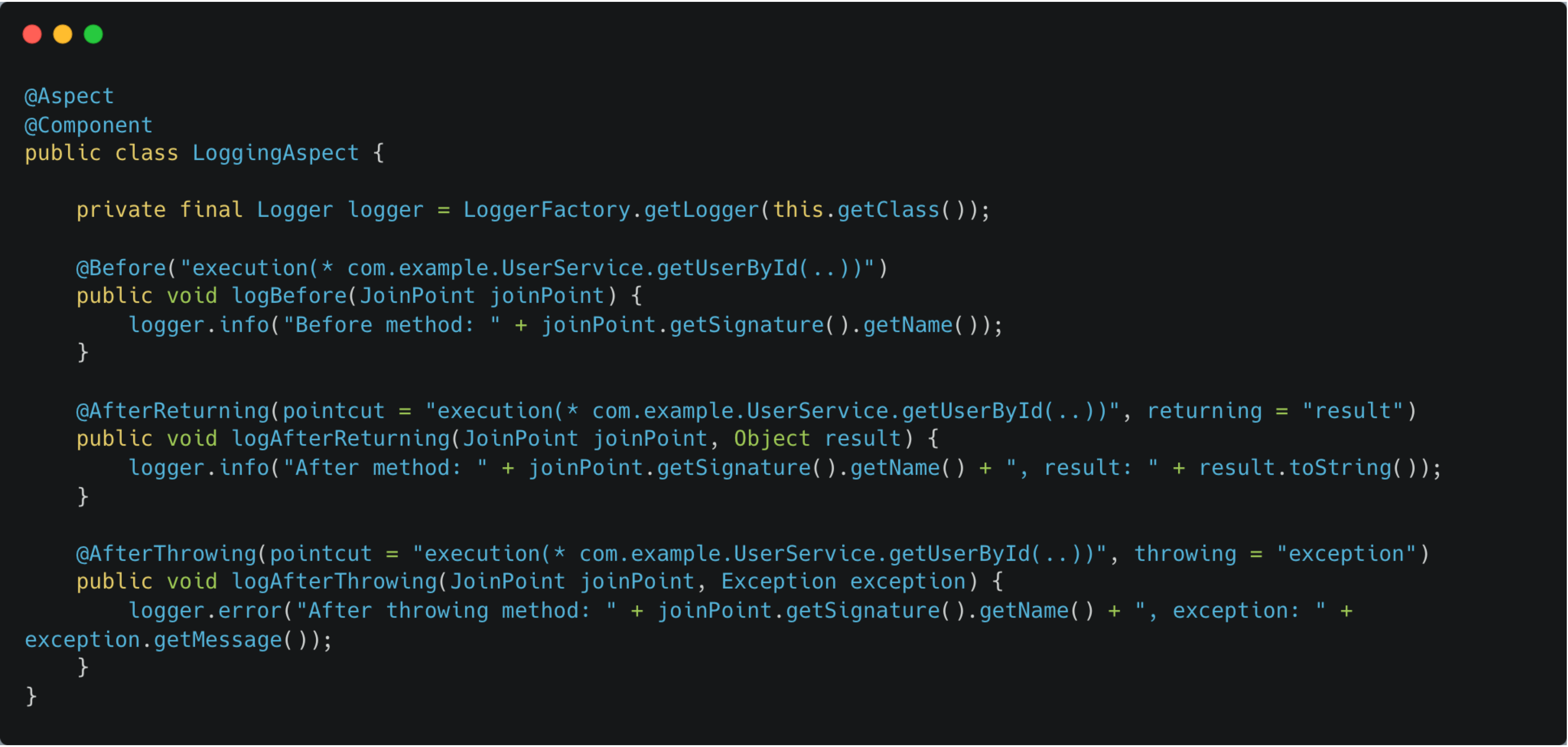


```java
@SpringBootApplication
public class MyApplication implements CommandLineRunner {
  @Autowired
  private ApplicationContext context;

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

  @Override
  public void run(String... args) throws Exception {
    Greeting greeting = context.getBean(Greeting.class);
    System.out.println(greeting.getMessage());
  }
}
```

Spring framework – Spring core

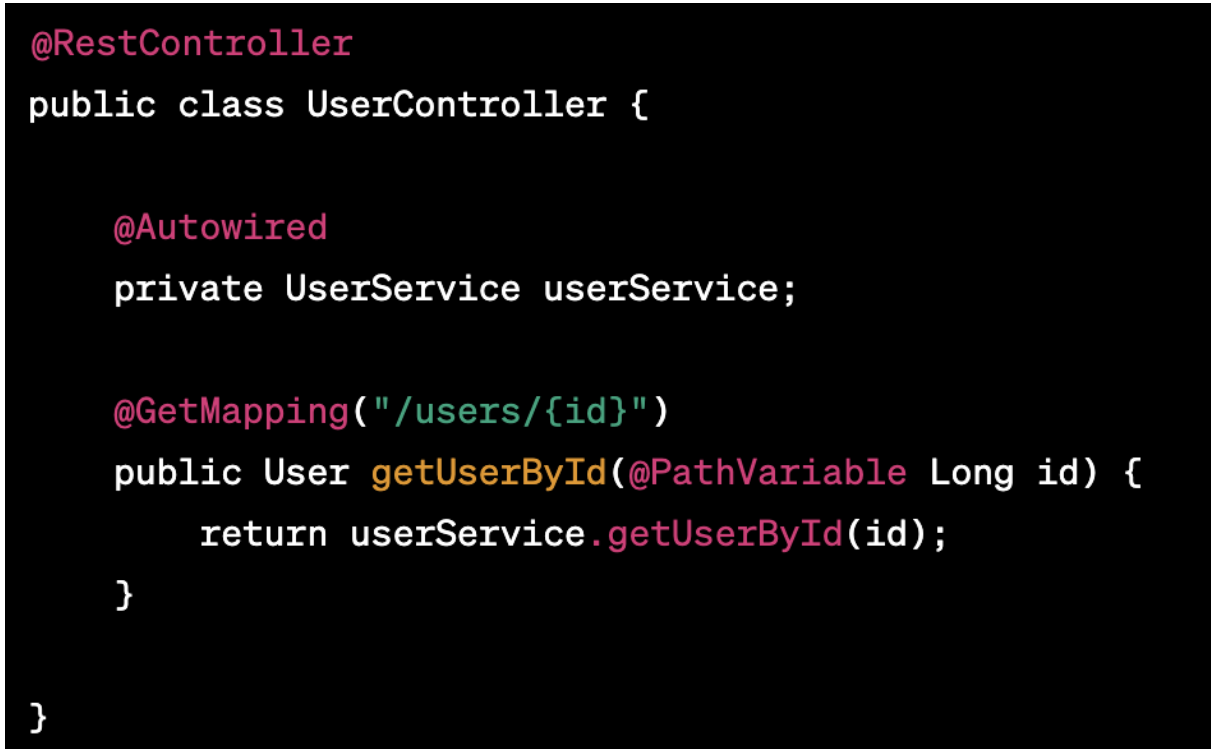
- Beans
- Autowiring
- Dependency Injection
- Inversion of Control
- IoC container
- Application Context

Spring framework – AOP



```java
@Aspect
@Component
public class LoggingAspect {

    private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @Before("execution(* com.example.UserService.getUserById(..))")
    public void logBefore(JoinPoint joinPoint) {
        logger.info("Before method: " + joinPoint.getSignature().getName());
    }

    @AfterReturning(pointcut = "execution(* com.example.UserService.getUserById(..))", returning = "result")
    public void logAfterReturning(JoinPoint joinPoint, Object result) {
        logger.info("After method: " + joinPoint.getSignature().getName() + ", result: " + result.toString());
    }

    @AfterThrowing(pointcut = "execution(* com.example.UserService.getUserById(..))", throwing = "exception")
    public void logAfterThrowing(JoinPoint joinPoint, Exception exception) {
        logger.error("After throwing method: " + joinPoint.getSignature().getName() + ", exception: " +
exception.getMessage());
    }
}
```

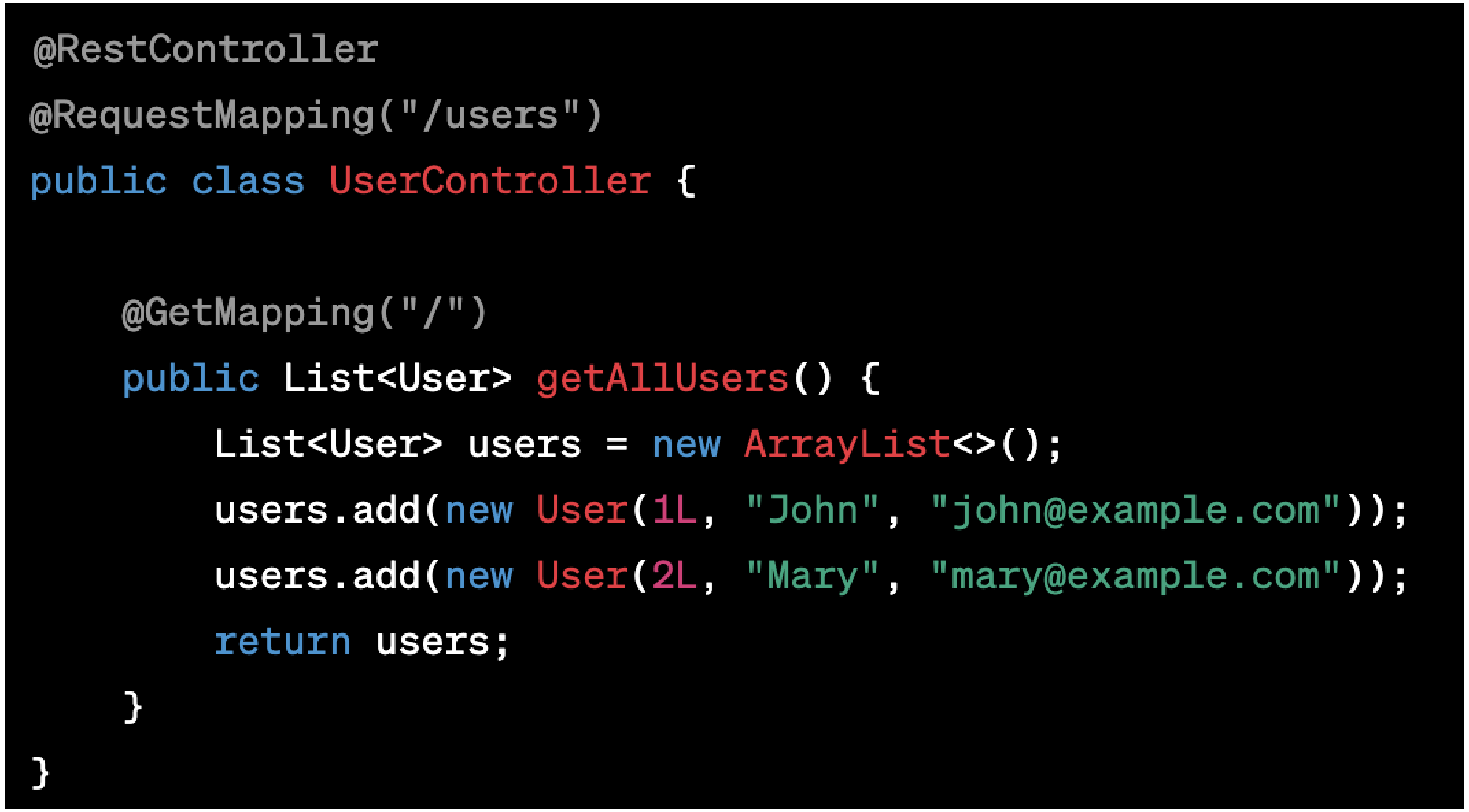
Spring framework – AOP



```java
@RestController
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping("/users/{id}")
    public User getUserById(@PathVariable Long id) {
        return userService.getUserById(id);
    }

}
```

```
INFO  c.e.LoggingAspect - Before method: getUserById
INFO  c.e.LoggingAspect - After method: getUserById, result: [User(id=<user_id>, name=<user_name>, email=<user_email>)]
```

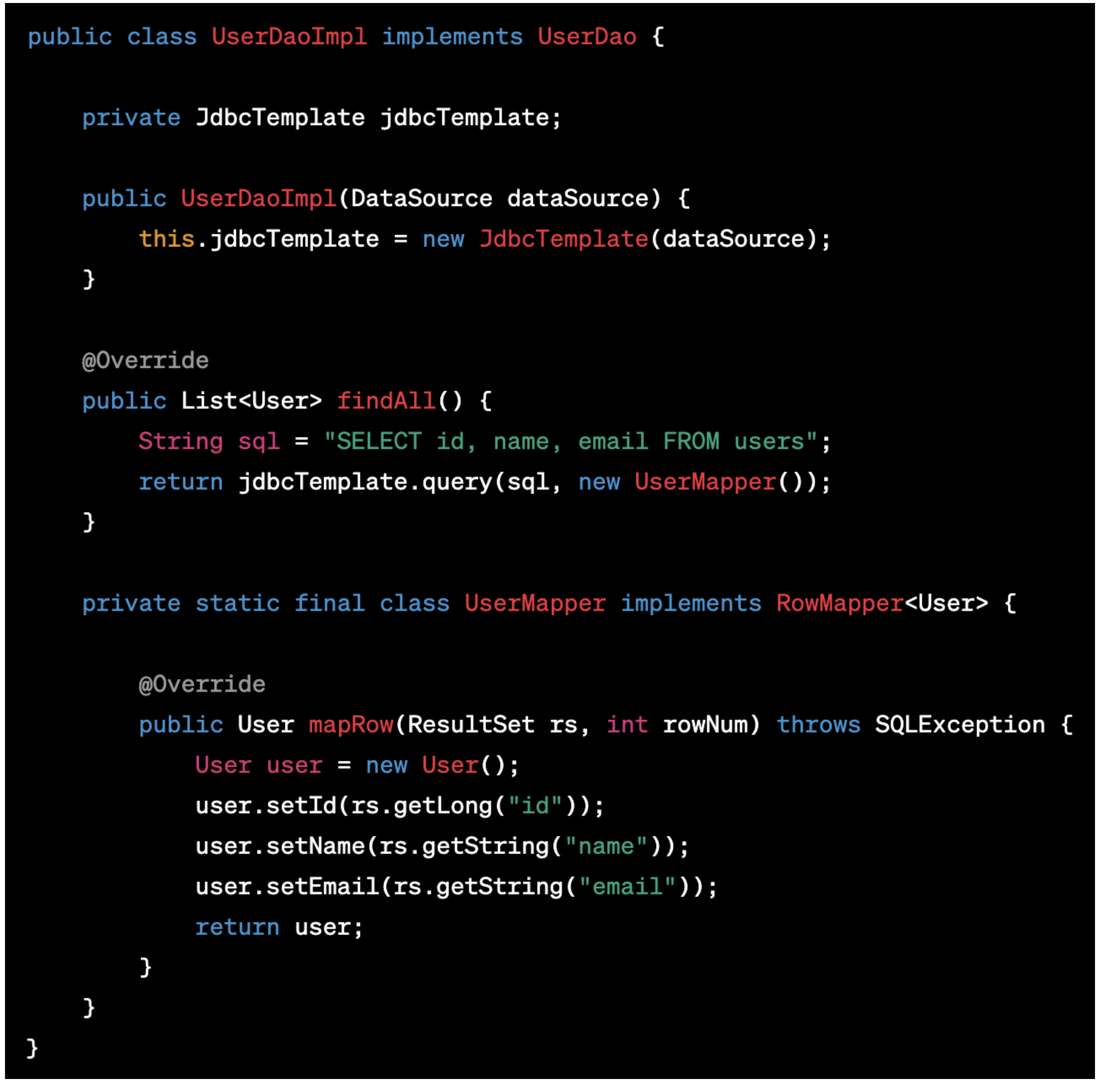
Spring framework – Web



```java
@RestController
@RequestMapping("/users")
public class UserController {

    @GetMapping("/")
    public List<User> getAllUsers() {
        List<User> users = new ArrayList<>();
        users.add(new User(1L, "John", "john@example.com"));
        users.add(new User(2L, "Mary", "mary@example.com"));
        return users;
    }
}
```

Spring framework – Data Access

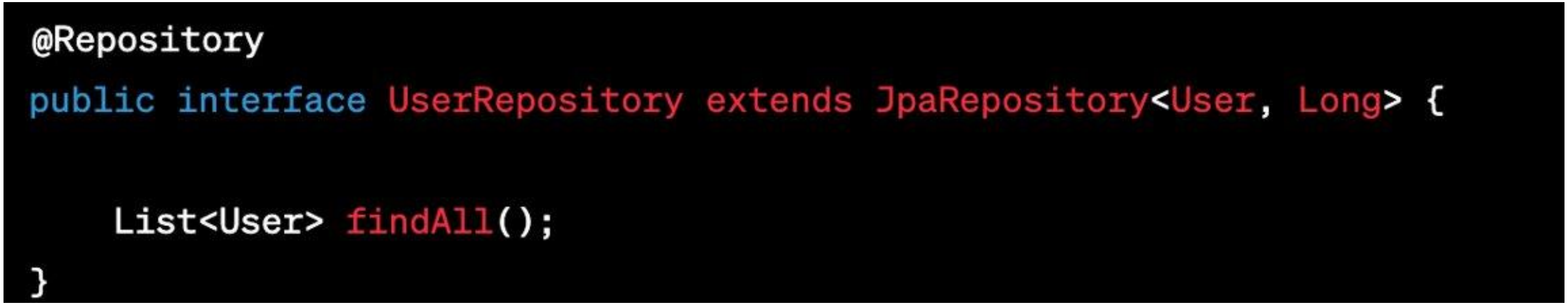
- Spring Data Jdbc



```java
public class UserDaoImpl implements UserDao {

    private JdbcTemplate jdbcTemplate;

    public UserDaoImpl(DataSource dataSource) {
        this.jdbcTemplate = new JdbcTemplate(dataSource);
    }

    @Override
    public List<User> findAll() {
        String sql = "SELECT id, name, email FROM users";
        return jdbcTemplate.query(sql, new UserMapper());
    }

    private static final class UserMapper implements RowMapper<User> {

        @Override
        public User mapRow(ResultSet rs, int rowNum) throws SQLException {
            User user = new User();
            user.setId(rs.getLong("id"));
            user.setName(rs.getString("name"));
            user.setEmail(rs.getString("email"));
            return user;
        }
    }
}
```

Spring framework – Data Access

- Spring Data Jpa



```
@Repository
public interface UserRepository extends JpaRepository<User, Long> {

    List<User> findAll();
}
```

Spring boot – What is Spring boot

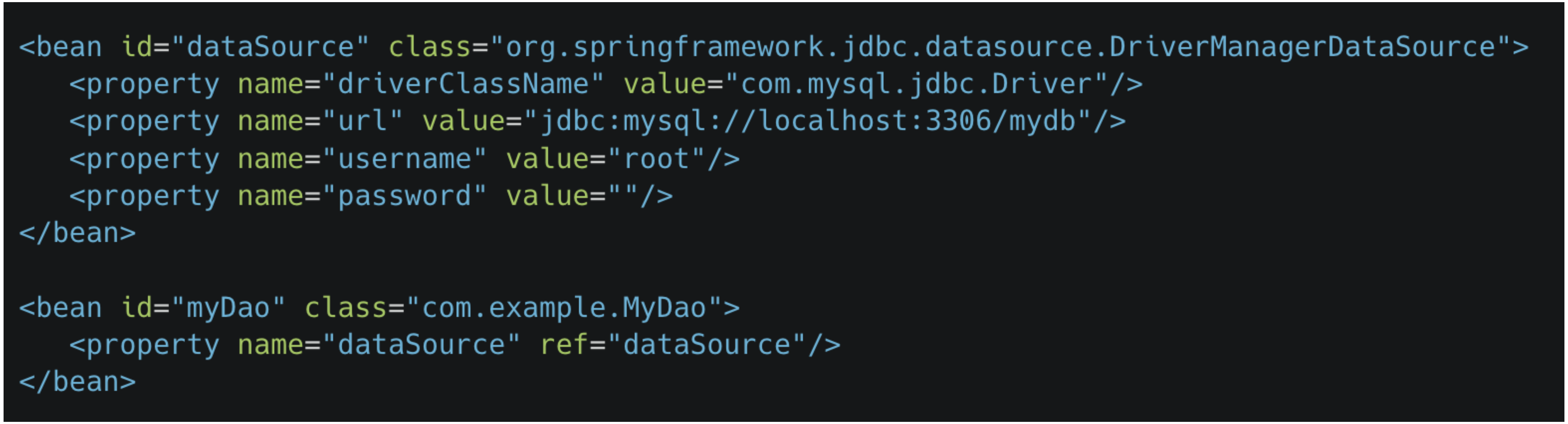
- Popular framework for building Java application
- Provides a large number of helper classes and annotations

Spring boot – Why using Spring boot?

- Minimum configuration
- Avoid complex XML configuration in Spring
- Embedded Web Servers

Spring boot – Why using Spring boot?

- Spring framework: Define the data source in the XML configuration file:

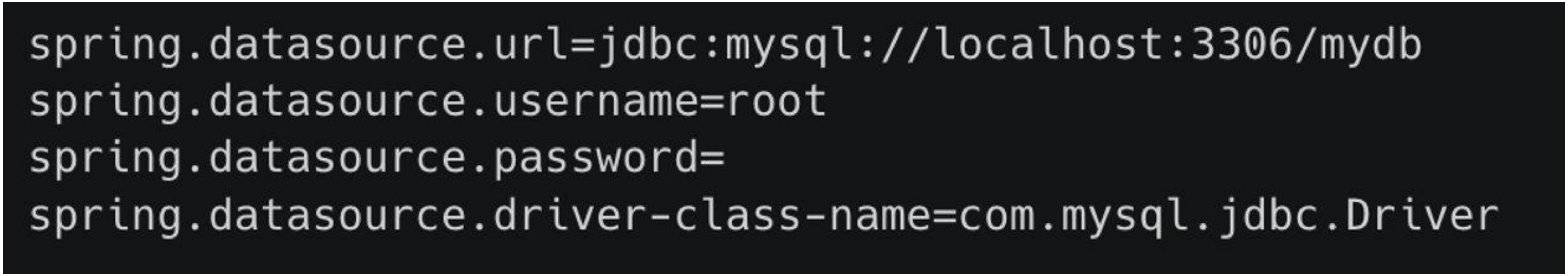


```xml
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
   <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
   <property name="url" value="jdbc:mysql://localhost:3306/mydb"/>
   <property name="username" value="root"/>
   <property name="password" value=""/>
</bean>

<bean id="myDao" class="com.example.MyDao">
   <property name="dataSource" ref="dataSource"/>
</bean>
```

Spring boot – Why using Spring boot?

- Spring boot:

Define in the **application.properties** or **application.yml**



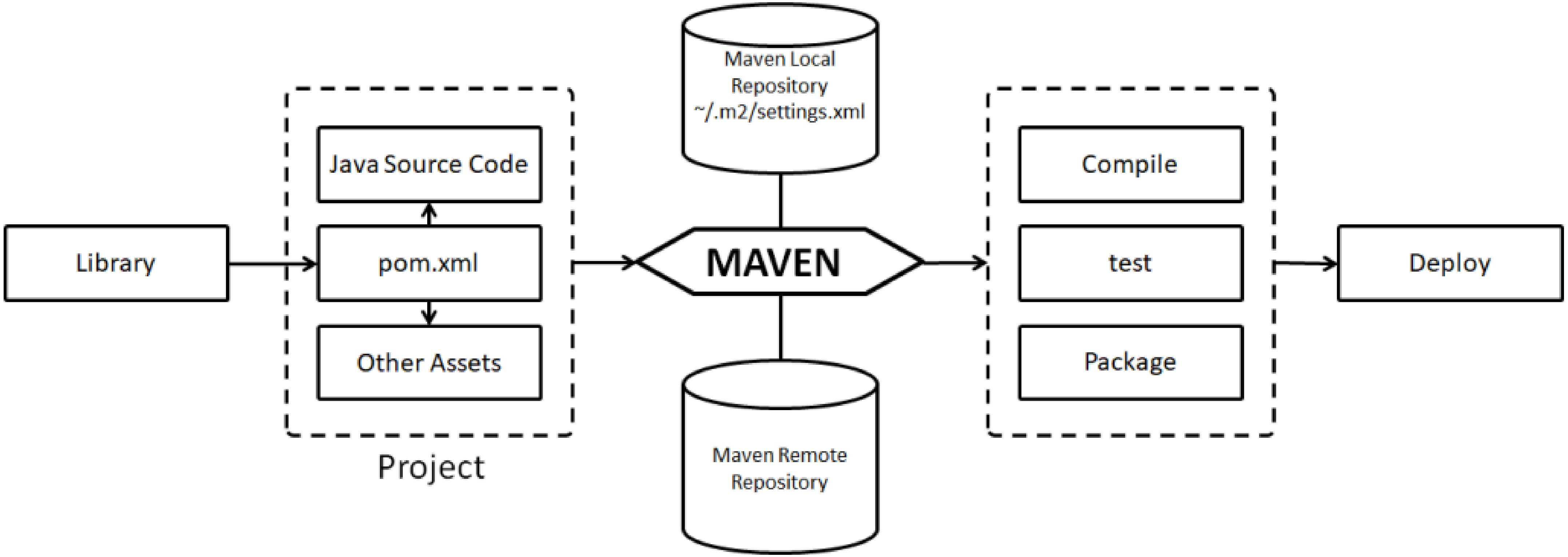
```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

Maven

- What is Maven?
- Maven Project structure
- Maven key concepts
  - Pom.xml
  - Project Coordinates
- Maven commands

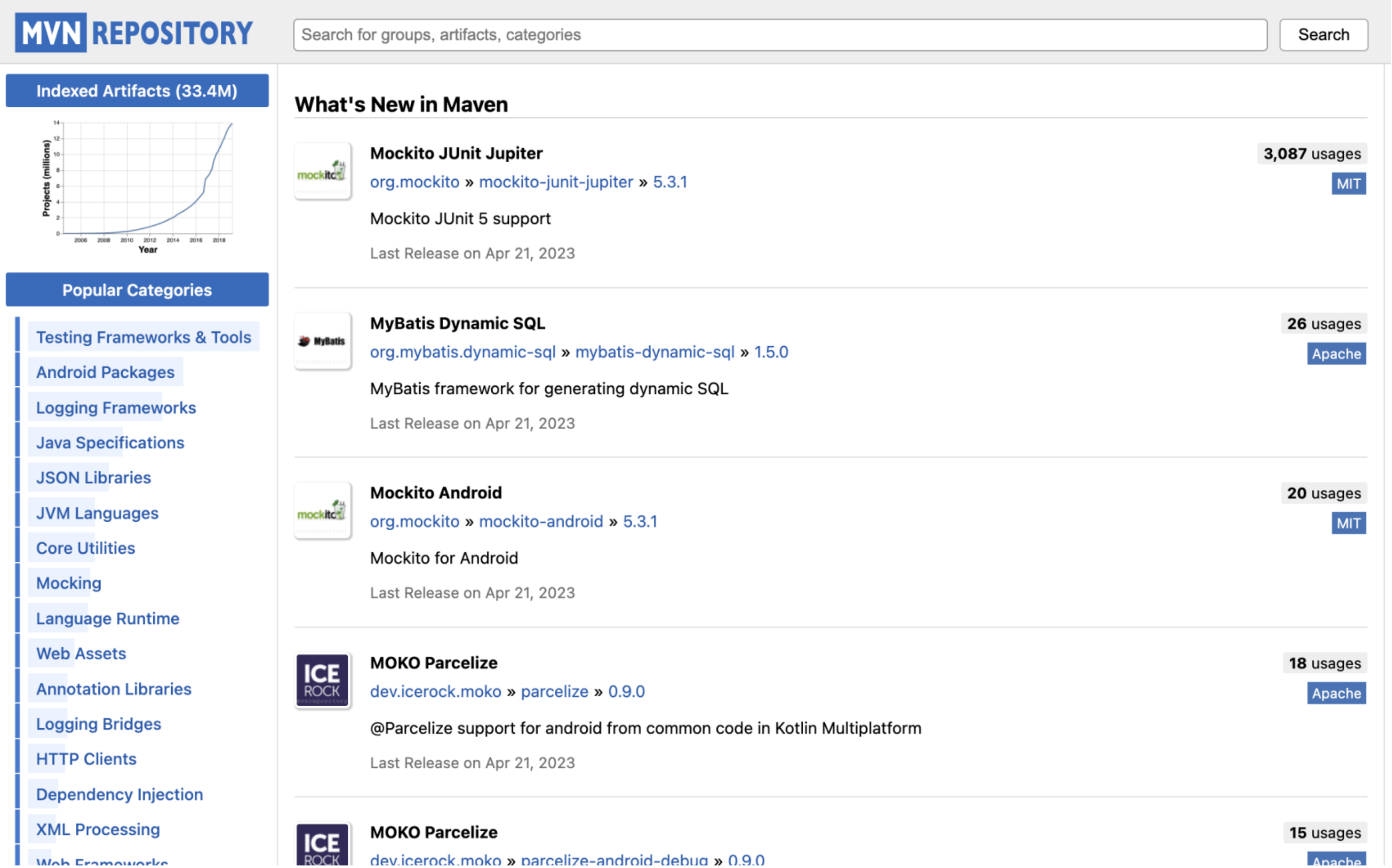
Maven – What is Maven?

Maven is a Project Management Tool. Most popular use of Maven is for build management and dependencies



Maven – What is Maven?

Remote Repository: https://mvnrepository.com/



Indexed Artifacts (33.4M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Core Utilities
- Mocking
- Language Runtime
- Web Assets
- Annotation Libraries
- Logging Bridges
- HTTP Clients
- Dependency Injection
- XML Processing
- Web Frameworks

What's New in Maven

Mockito JUnit Jupiter — 3,087 usages — MIT

org.mockito » mockito-junit-jupiter » 5.3.1

Mockito JUnit 5 support

Last Release on Apr 21, 2023

MyBatis Dynamic SQL — 26 usages — Apache

org.mybatis.dynamic-sql » mybatis-dynamic-sql » 1.5.0

MyBatis framework for generating dynamic SQL

Last Release on Apr 21, 2023

Mockito Android — 20 usages — MIT

org.mockito » mockito-android » 5.3.1

Mockito for Android

Last Release on Apr 21, 2023

MOKO Parcelize — 18 usages — Apache

dev.icerock.moko » parcelize » 0.9.0

@Parcelize support for android from common code in Kotlin Multiplatform

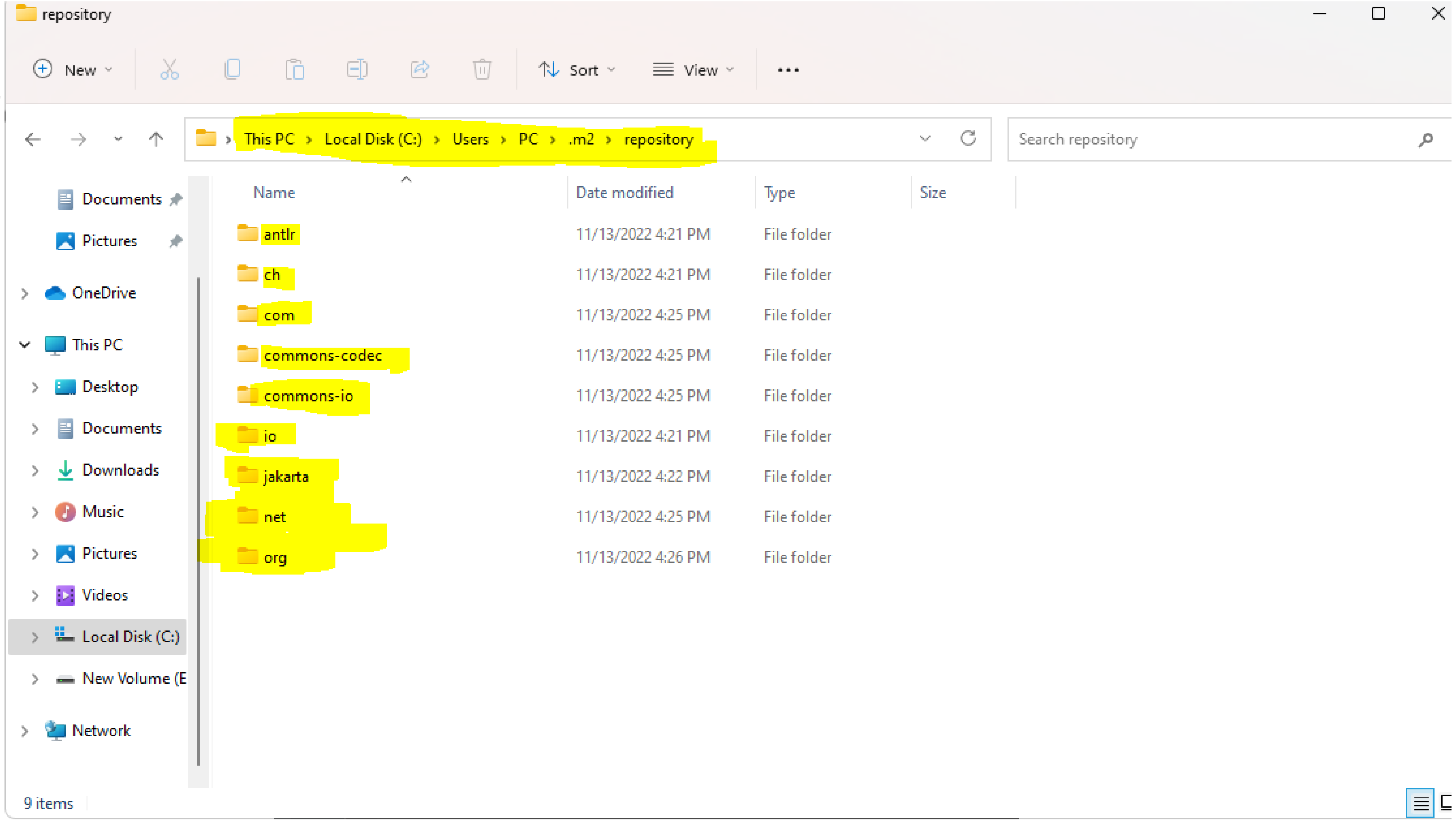
Last Release on Apr 21, 2023

MOKO Parcelize — 15 usages — Apache

dev.icerock.moko » parcelize-android-debug » 0.9.0

Maven – What is Maven?

Local Repository

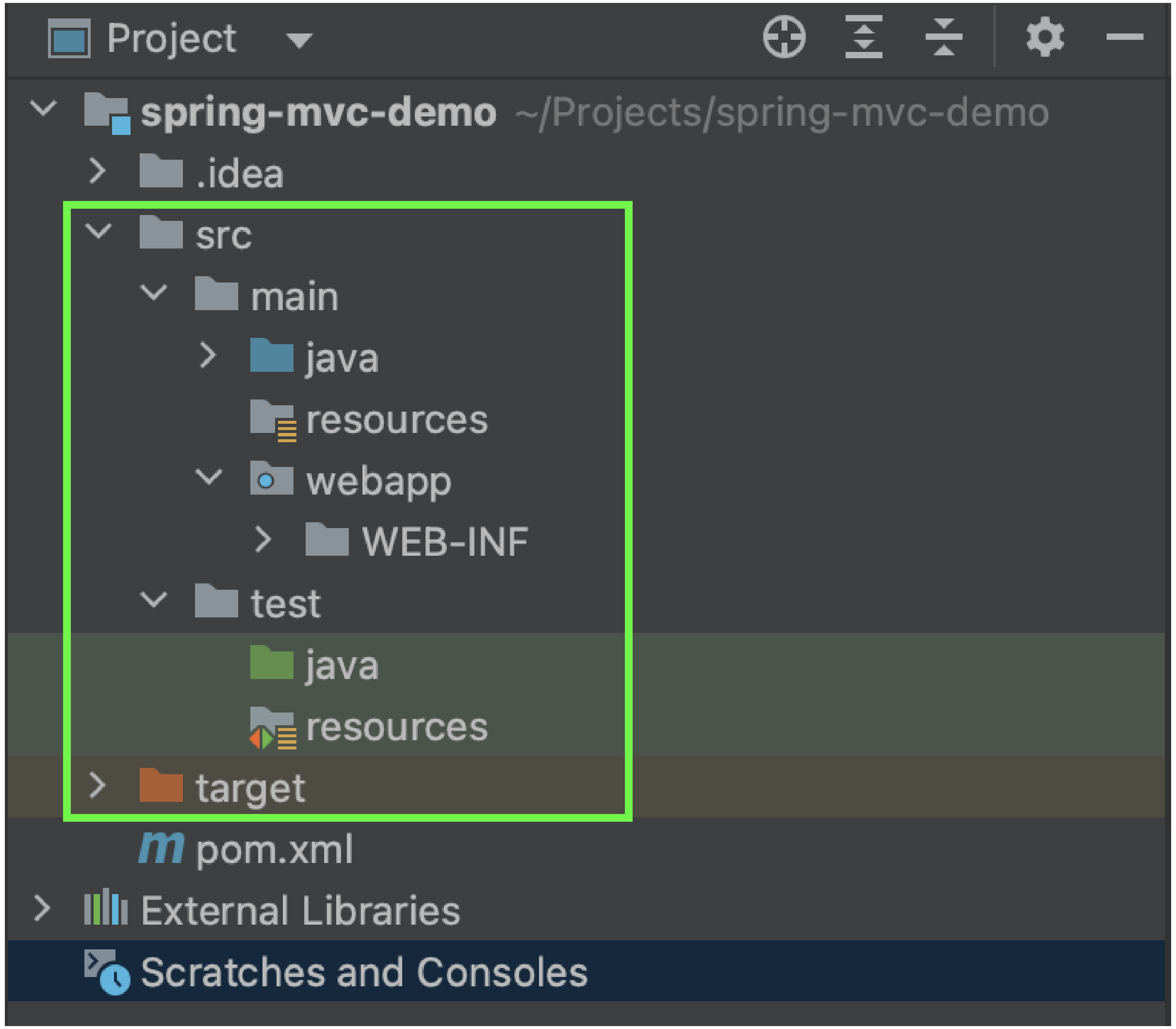


This PC > Local Disk (C:) > Users > PC > .m2 > repository

Name	Date modified	Type	Size
antlr	11/13/2022 4:21 PM	File folder	
ch	11/13/2022 4:21 PM	File folder	
com	11/13/2022 4:25 PM	File folder	
commons-codec	11/13/2022 4:25 PM	File folder	
commons-io	11/13/2022 4:25 PM	File folder	
io	11/13/2022 4:21 PM	File folder	
jakarta	11/13/2022 4:22 PM	File folder	
net	11/13/2022 4:25 PM	File folder	
org	11/13/2022 4:26 PM	File folder	

9 items

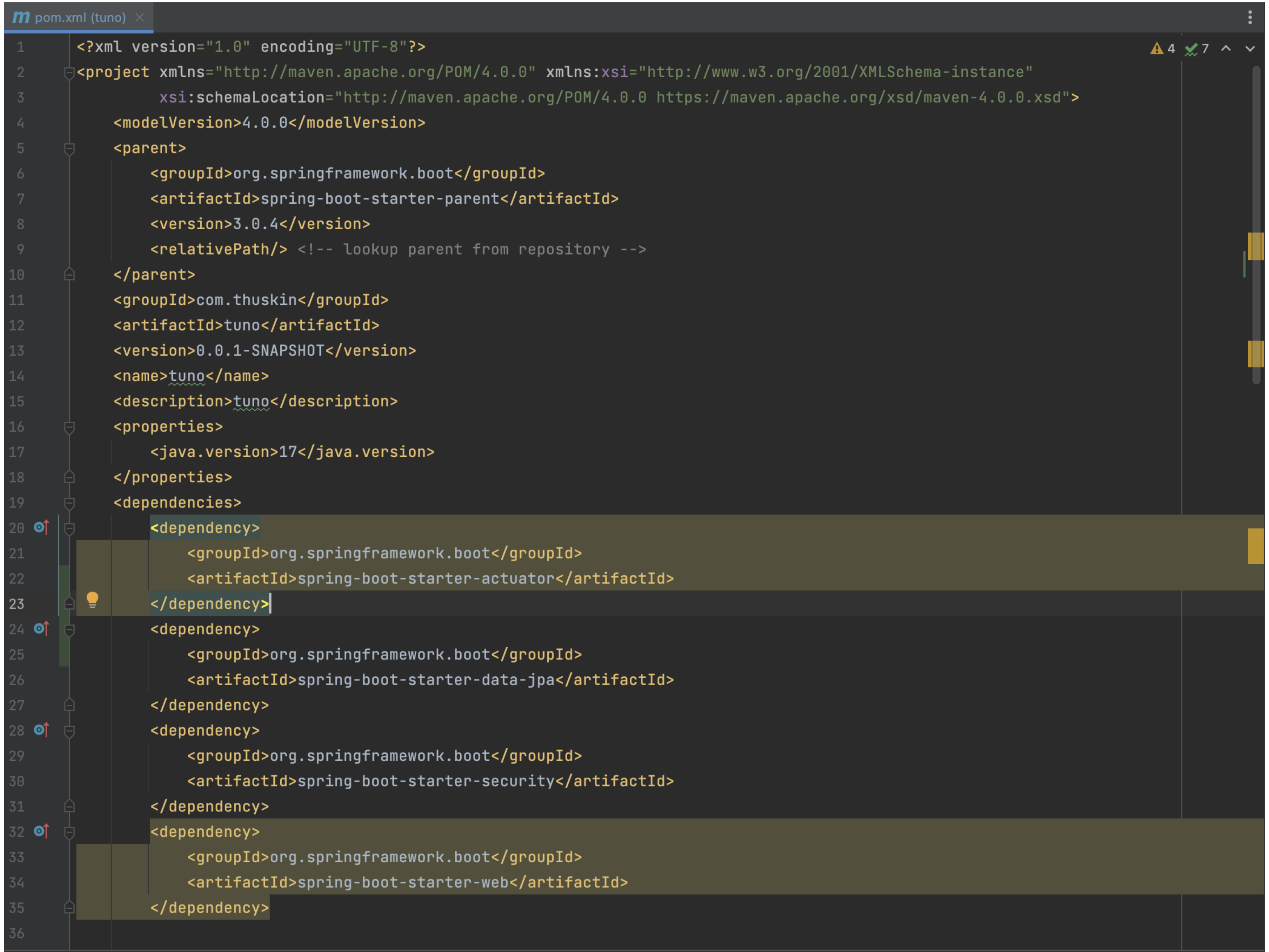
Maven – Project structure



Project

```
spring-mvc-demo ~/Projects/spring-mvc-demo
  .idea
  src
    main
      java
      resources
      webapp
        WEB-INF
    test
      java
      resources
  target
  pom.xml
External Libraries
Scratches and Consoles
```

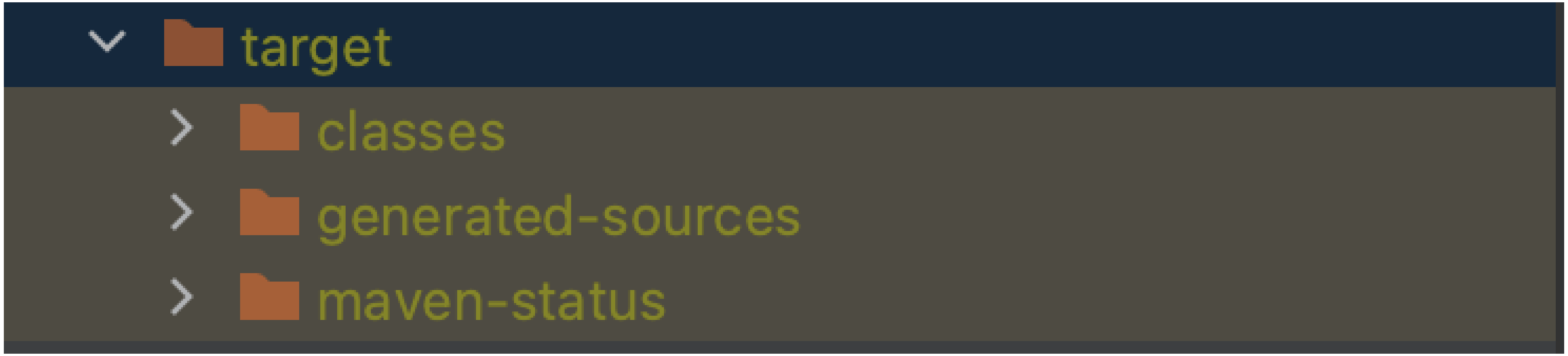
Maven – Maven Key concepts



```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.4</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.thuskin</groupId>
    <artifactId>tuno</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>tuno</name>
    <description>tuno</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

```

Maven – Commands

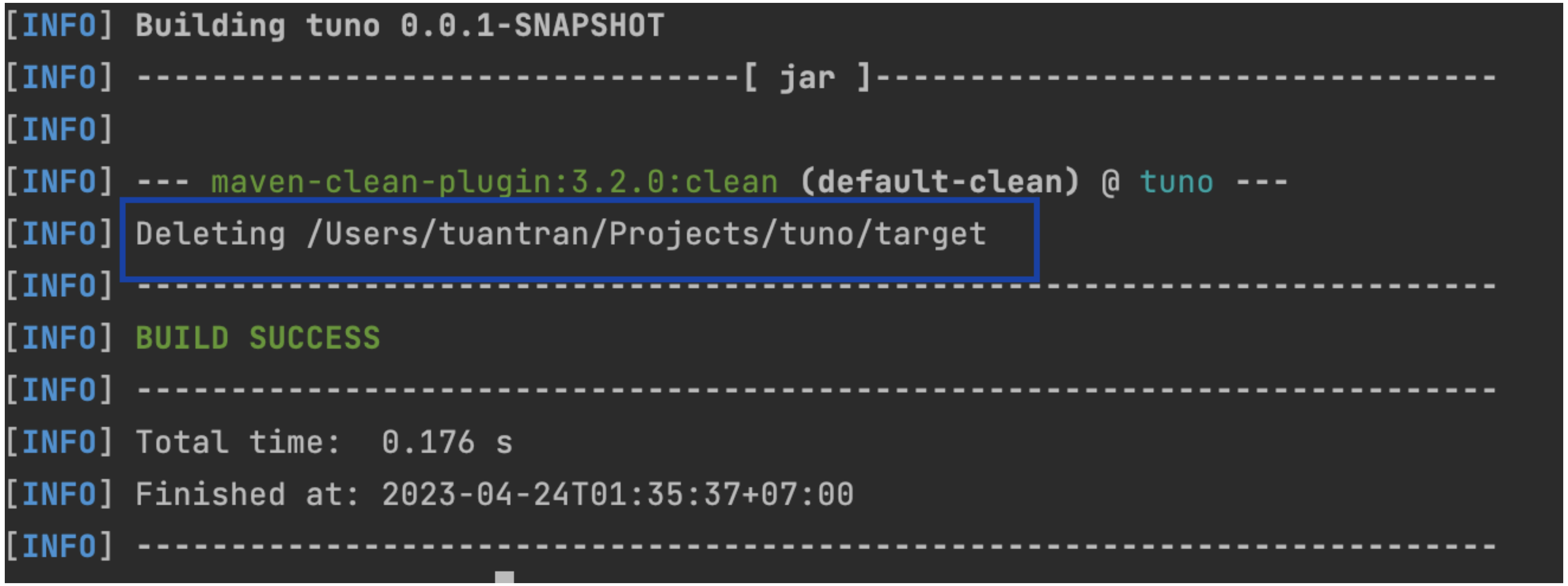
Compile



```
target
  classes
  generated-sources
  maven-status
```

Maven – Commands

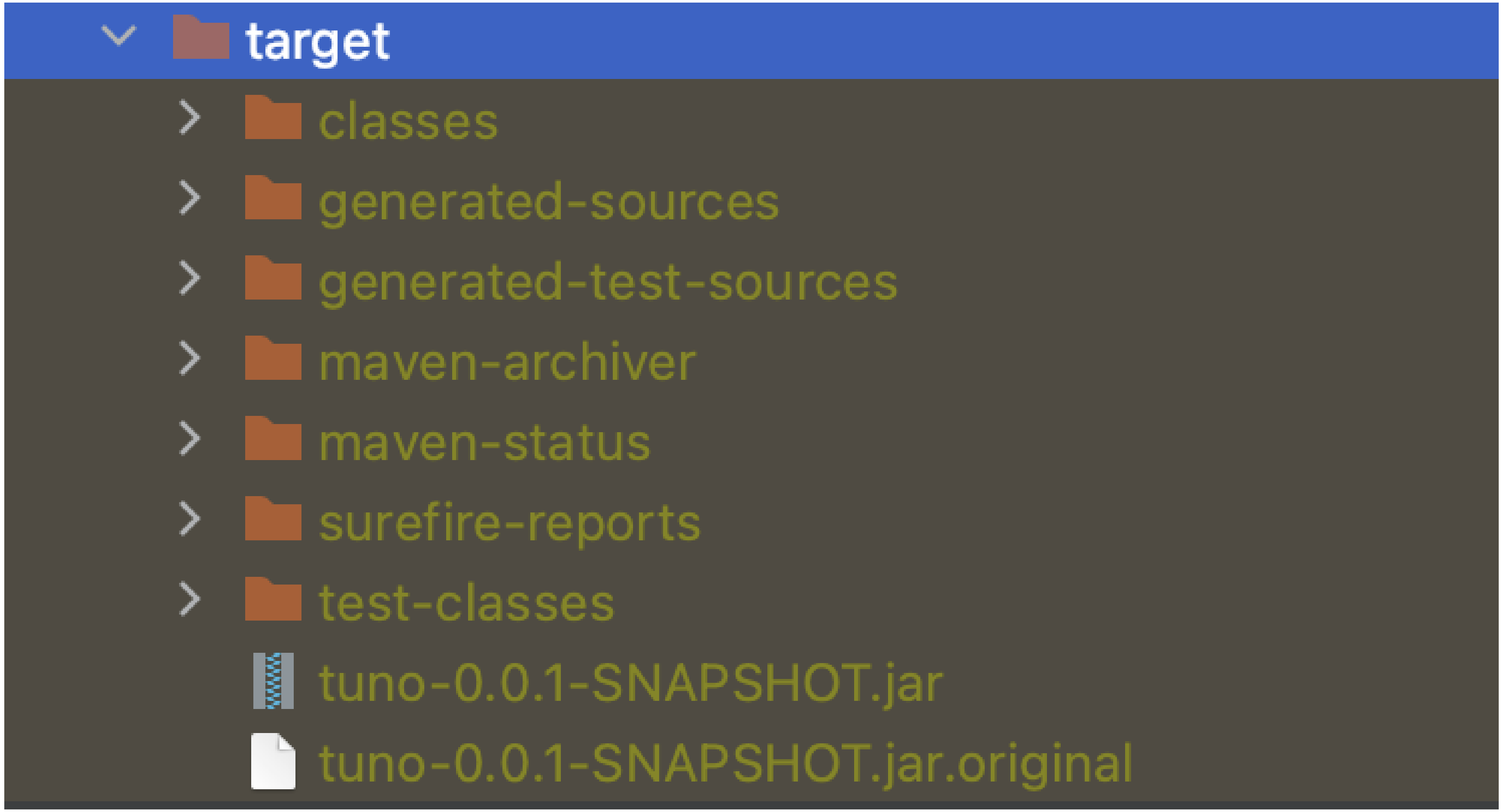
Clean



```
[INFO] Building tuno 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ tuno ---
[INFO] Deleting /Users/tuantran/Projects/tuno/target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.176 s
[INFO] Finished at: 2023-04-24T01:35:37+07:00
[INFO] ------------------------------------------------------------------------
```

Maven – Commands

Package

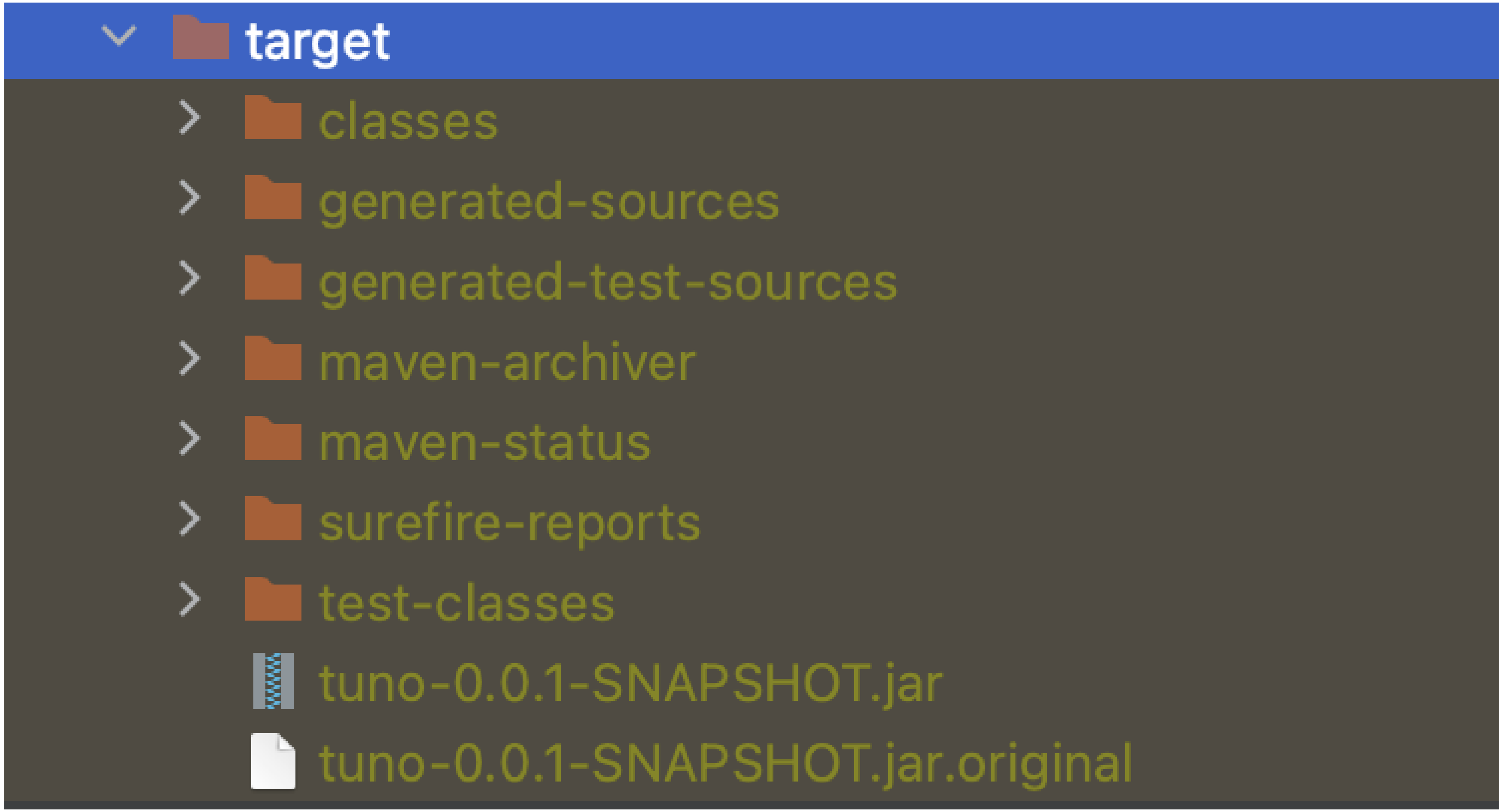


target
- classes
- generated-sources
- generated-test-sources
- maven-archiver
- maven-status
- surefire-reports
- test-classes
- tuno-0.0.1-SNAPSHOT.jar
- tuno-0.0.1-SNAPSHOT.jar.original

Maven – Commands

Resource Software Solution

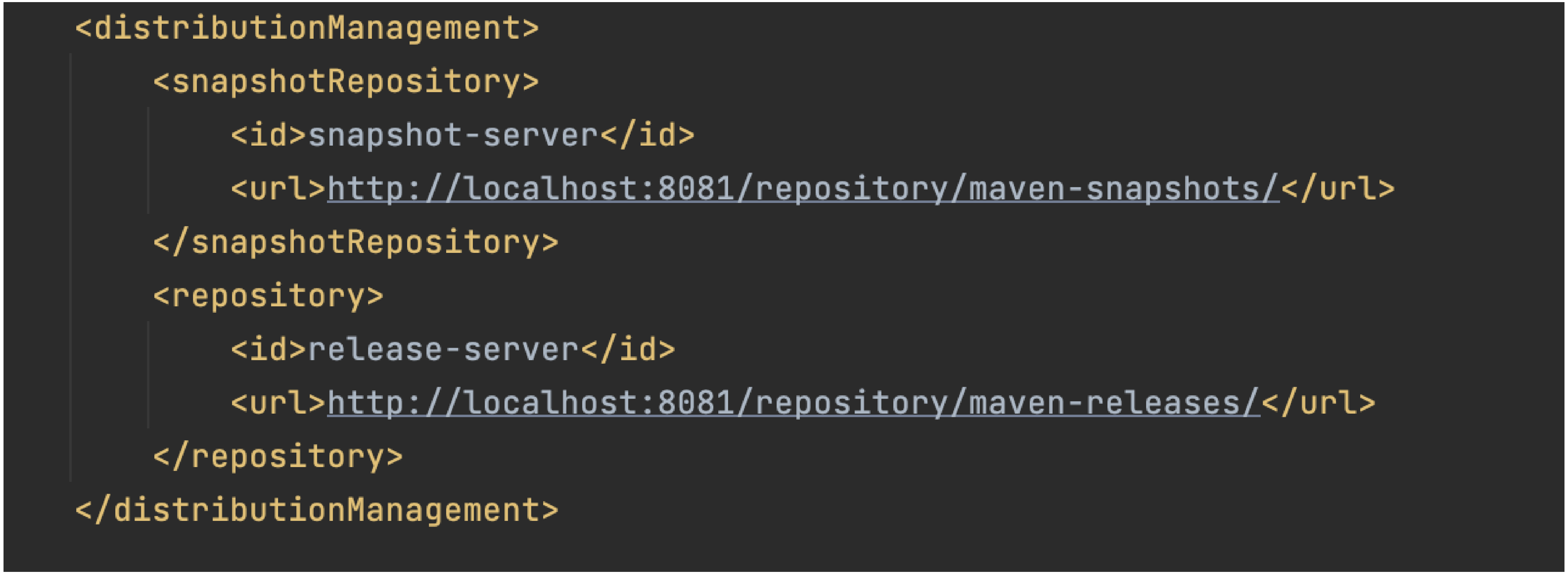
Install



```
target
  classes
  generated-sources
  generated-test-sources
  maven-archiver
  maven-status
  surefire-reports
  test-classes
  tuno-0.0.1-SNAPSHOT.jar
  tuno-0.0.1-SNAPSHOT.jar.original
```

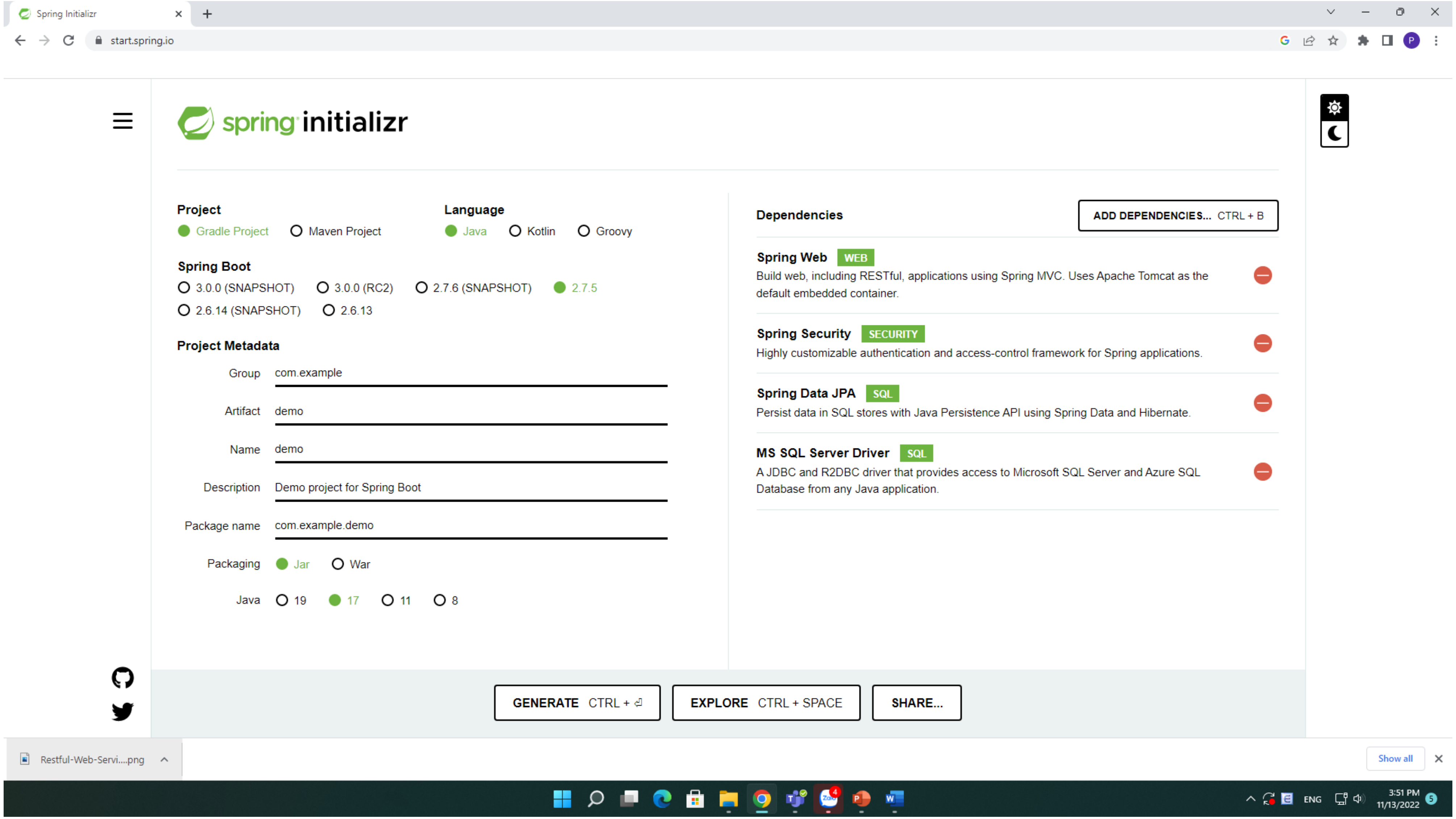
Maven – Commands

Deploy



```
<distributionManagement>
    <snapshotRepository>
        <id>snapshot-server</id>
        <url>http://localhost:8081/repository/maven-snapshots/</url>
    </snapshotRepository>
    <repository>
        <id>release-server</id>
        <url>http://localhost:8081/repository/maven-releases/</url>
    </repository>
</distributionManagement>
```

Spring Initializr



Project

● Gradle Project ○ Maven Project

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

○ 3.0.0 (SNAPSHOT) ○ 3.0.0 (RC2) ○ 2.7.6 (SNAPSHOT) ● 2.7.5
○ 2.6.14 (SNAPSHOT) ○ 2.6.13

Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	● Jar ○ War
Java	○ 19 ● 17 ○ 11 ○ 8

Dependencies ADD DEPENDENCIES... CTRL + B

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

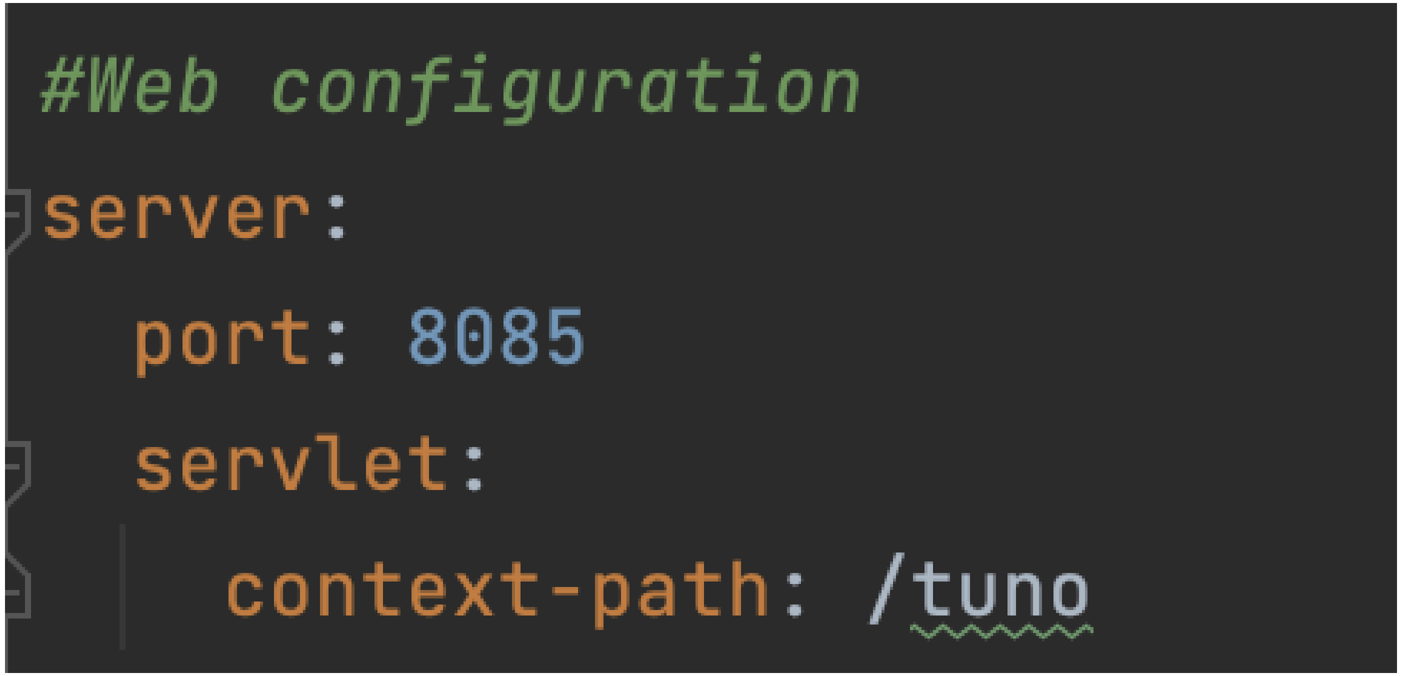
Spring Security SECURITY
Highly customizable authentication and access-control framework for Spring applications.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

MS SQL Server Driver SQL
A JDBC and R2DBC driver that provides access to Microsoft SQL Server and Azure SQL Database from any Java application.

Spring Boot Initialzr Demo

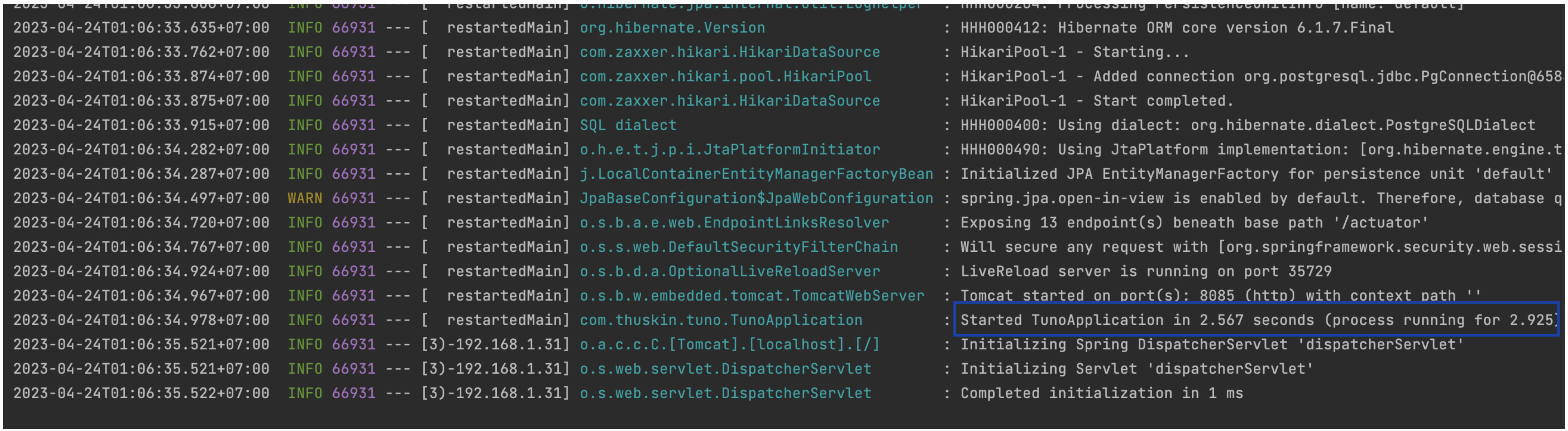
Configure web:



```
#Web configuration
server:
  port: 8085
  servlet:
    context-path: /tuno
```

Spring Boot Initialzr Demo

Run application:



```
2023-04-24T01:06:33.608+07:00  INFO 66931 --- [  restartedMain] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-04-24T01:06:33.635+07:00  INFO 66931 --- [  restartedMain] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.1.7.Final
2023-04-24T01:06:33.762+07:00  INFO 66931 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2023-04-24T01:06:33.874+07:00  INFO 66931 --- [  restartedMain] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection org.postgresql.jdbc.PgConnection@658
2023-04-24T01:06:33.875+07:00  INFO 66931 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2023-04-24T01:06:33.915+07:00  INFO 66931 --- [  restartedMain] SQL dialect                              : HHH000400: Using dialect: org.hibernate.dialect.PostgreSQLDialect
2023-04-24T01:06:34.282+07:00  INFO 66931 --- [  restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.t
2023-04-24T01:06:34.287+07:00  INFO 66931 --- [  restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-04-24T01:06:34.497+07:00  WARN 66931 --- [  restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database q
2023-04-24T01:06:34.720+07:00  INFO 66931 --- [  restartedMain] o.s.b.a.e.web.EndpointLinksResolver      : Exposing 13 endpoint(s) beneath base path '/actuator'
2023-04-24T01:06:34.767+07:00  INFO 66931 --- [  restartedMain] o.s.s.web.DefaultSecurityFilterChain     : Will secure any request with [org.springframework.security.web.sessi
2023-04-24T01:06:34.924+07:00  INFO 66931 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2023-04-24T01:06:34.967+07:00  INFO 66931 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8085 (http) with context path ''
2023-04-24T01:06:34.978+07:00  INFO 66931 --- [  restartedMain] com.thuskin.tuno.TunoApplication         : Started TunoApplication in 2.567 seconds (process running for 2.925)
2023-04-24T01:06:35.521+07:00  INFO 66931 --- [3)-192.168.1.31] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-04-24T01:06:35.521+07:00  INFO 66931 --- [3)-192.168.1.31] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2023-04-24T01:06:35.522+07:00  INFO 66931 --- [3)-192.168.1.31] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

*Keeping up those **inspiration** and the **enthusiasm** in the **learning path**. Let confidence to bring it into **your career path** for getting gain the **success** as your expectation.*

Thank you

Contact

- Name: R2S Academy

- Email: daotao@r2s.edu.vn

- Hotline/Zalo: 0919 365 363

- Website: https://r2s.edu.vn

- Fanpage: https://www.facebook.com/r2s.tuyendung

Questions and Answers