**WEEK-5**



**Exercise-1**: **Create a Spring Web Project using Maven**

**pom.xml**

<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

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>spring-data-jpa-demo</artifactId>

  <version>1.0.0</version>

  <packaging>jar</packaging>

  <parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>3.2.5</version>

    <relativePath/>

  </parent>

  <properties>

    <java.version>21</java.version>

  </properties>

  <dependencies>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-data-jpa</artifactId>

  </dependency>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-web</artifactId>

  </dependency>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-security</artifactId>

  </dependency>

  <dependency>

    <groupId>com.h2database</groupId>

    <artifactId>h2</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-test</artifactId>

    <scope>test</scope>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-api</artifactId>

    <version>0.11.5</version>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-impl</artifactId>

    <version>0.11.5</version>

    <scope>runtime</scope>

  </dependency>

  <dependency>

    <groupId>io.jsonwebtoken</groupId>

    <artifactId>jjwt-jackson</artifactId>

    <version>0.11.5</version>

    <scope>runtime</scope>

  </dependency>

</dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-maven-plugin</artifactId>

      </plugin>

    </plugins>

  </build>

</project>

**SpringLearnApplication.java**

package com.example.spring\_data\_ jpa\_demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

    public static void main(String[] args) {

        SpringApplication.run(SpringLearnApplication.class, args);

        }

}

**SecurityConfig.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

        http

            .csrf(csrf -> csrf.disable())

            .authorizeHttpRequests(auth -> auth

                .anyRequest().permitAll()

            );

        return http.build();

    }

}

**AccountController.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.web.bind.annotation.\*;

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

    @GetMapping("/{number}")

    public Map<String, Object> getAccountDetails(@PathVariable String number) {

        return Map.of(

                "number", number,

                "type", "savings",

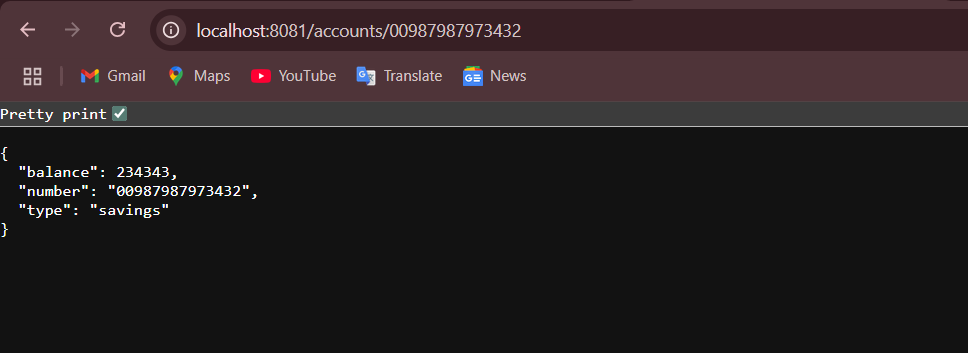
                "balance", 234343

        );

    }

}

**OUTPUT-1 :**



**LoanController.java**

package com.example.spring\_data\_jpa\_demo;

import org.springframework.web.bind.annotation.\*;

import java.util.Map;

@RestController

@RequestMapping("/loans")

public class LoanController {

    @GetMapping("/{number}")

    public Map<String, Object> getLoanDetails(@PathVariable String number) {

        return Map.of(

                "number", number,

                "type", "car",

                "loan", 400000,

                "emi", 3258,

                "tenure", 18

        );

    }

}

**OUTPUT-2 :**

