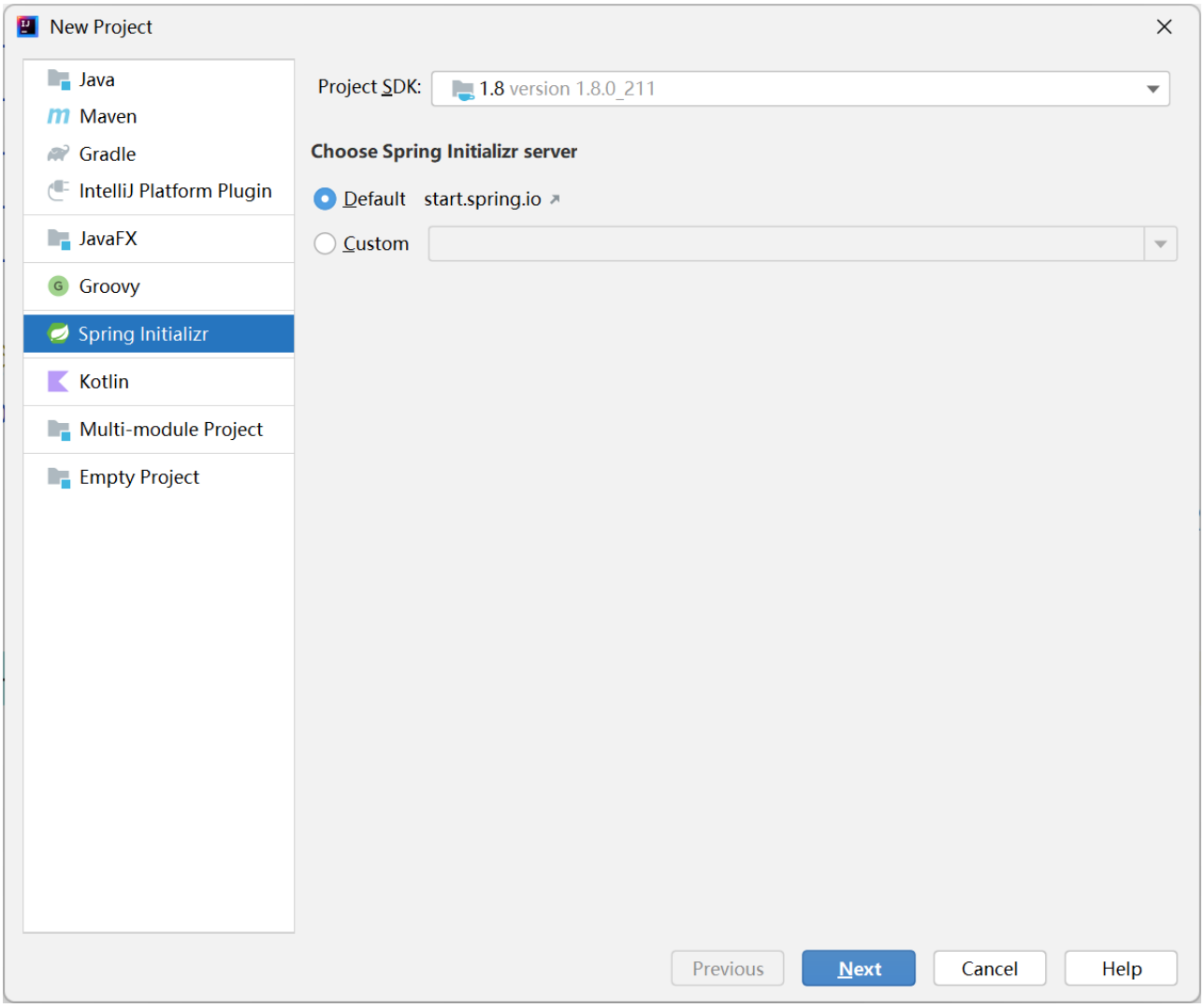
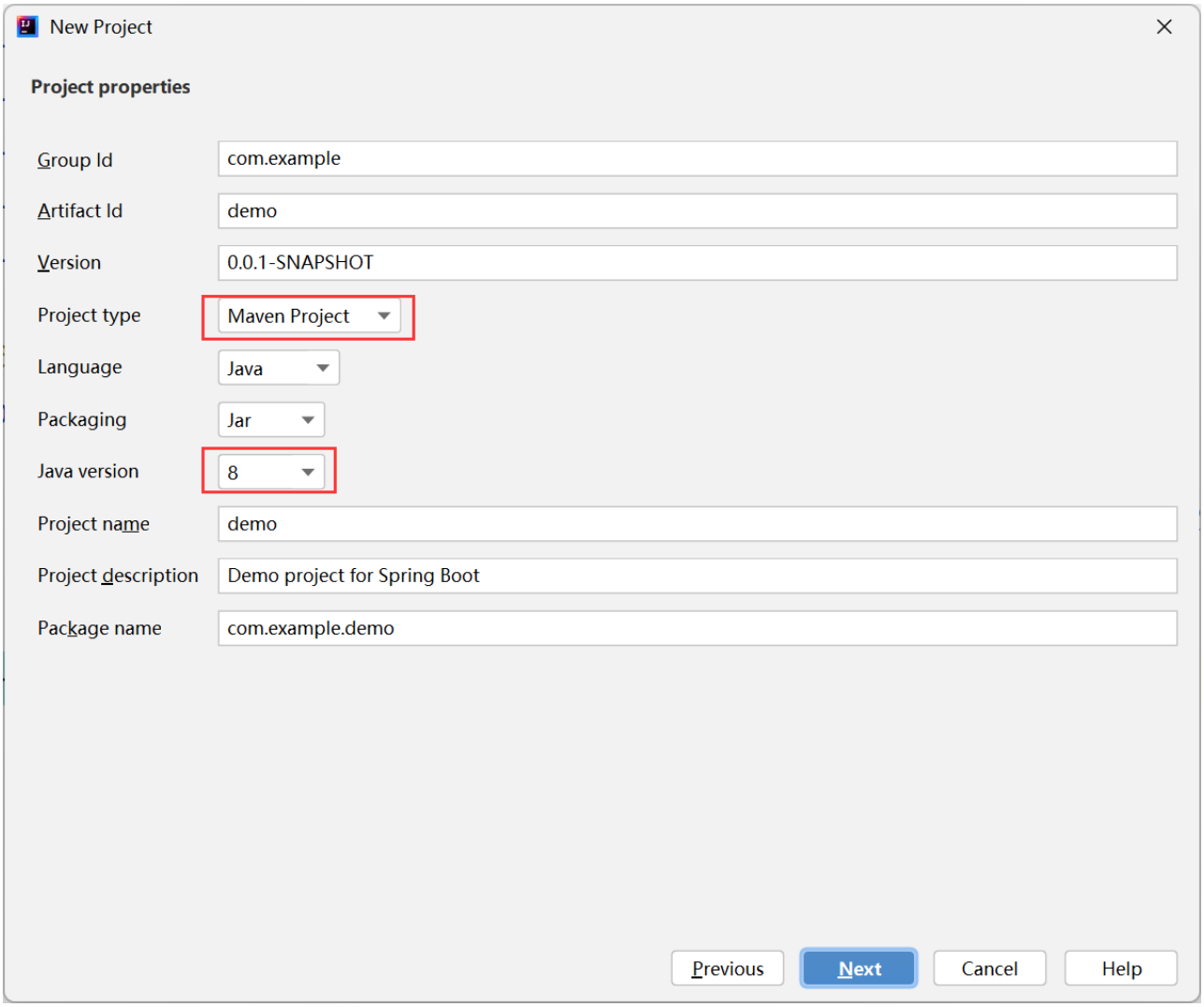
Spring Session 持久化

1.存储到 MySQL

将 Session 存储到 MySQL 数据库中，具体操作步骤如下。

1.1 创建 Spring Boot 项目（非必须）





1.2 添加依赖

以上 3 个依赖缺一不可，Maven 项目直接在 pom.xml 中添加以下代码：

```
<dependency>
      <groupId>org.springframework.session</groupId>
      <artifactId>spring-session-jdbc</artifactId>
  </dependency>
 <dependency>
      <groupId>org.mybatis.spring.boot</groupId>
      <artifactId>mybatis-spring-boot-starter</artifactId>
      <version>2.2.2</version>
  </dependency>
  <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <scope>runtime</scope>
  </dependency>
```

1.3 设置配置文件

```
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/java10?characterEncoding=utf8&useSSL=true
spring.datasource.username=root
spring.datasource.password=12345678
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.session.store-type=jdbc
spring.session.jdbc.initialize-schema=always
spring.session.jdbc.table-name=SPRING_SESSION
```

其中：

- spring.session.jdbc.initialize-schema：表示让程序自动创建 Session 存储的表结构，无需自己手动创建；
- spring.session.jdbc.table-name：表示设置 Session 存储表的名称。

1.4 存储和读取代码

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

@RestController
public class UserController {
    private static final String SESS_USER_KEY = "SESS_USER_KEY";

    // 存储 session
    @RequestMapping("/login")
    public String login(HttpSession httpSession) {
        httpSession.setAttribute(SESS_USER_KEY, "zhangsan");
        return "Session set success.";
    }

    // 查询 session
    @RequestMapping("/user")
    public String getUser(HttpServletRequest request) {
        String user = "NULL";
        HttpSession session = request.getSession(false);
        if (session != null) {
            Object userObj = session.getAttribute(SESS_USER_KEY);
            if (userObj != null) {
                user = userObj.toString();
            }
        }
        return user;
    }
}
```

注意事项

session 存储需要借助 cookie，默认情况下 cookie 是会话级别的，想要实现非会话级别的 cookie 和 session 保持，就需要在配置文件中设置 cookie 的过期时间：

```
server.servlet.session.cookie.max-age=180
```

2.存储到 Redis

2.1 添加依赖

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.session</groupId>
    <artifactId>spring-session-data-redis</artifactId>
</dependency>
```

2.2 修改配置

```
spring.session.store-type=redis
server.servlet.session.timeout=1800
spring.session.redis.flush-mode=on_save
spring.session.redis.namespace=spring:session
spring.redis.host=82.157.14.10
spring.redis.password=
spring.redis.port=6379
```

2.3 存储和读取代码

```
// https://docs.spring.io/spring-session/reference/guides/boot-redis.html
@RestController
public class UserController {
    private static final String SESS_USER_KEY = "SESS_USER_KEY";

    @RequestMapping("/login")
    public String login(HttpSession httpSession) {
        httpSession.setAttribute(SESS_USER_KEY, "wangwu");
        return "Session set success.";
    }

    @RequestMapping("/user")
    public String getUser(HttpServletRequest request) {
        String user = "NULL";
        HttpSession session = request.getSession(false);
        if (session != null) {
            Object userObj = session.getAttribute(SESS_USER_KEY);
            if (userObj != null) {
                user = userObj.toString();
            }
        }
        return user;
    }
}
```