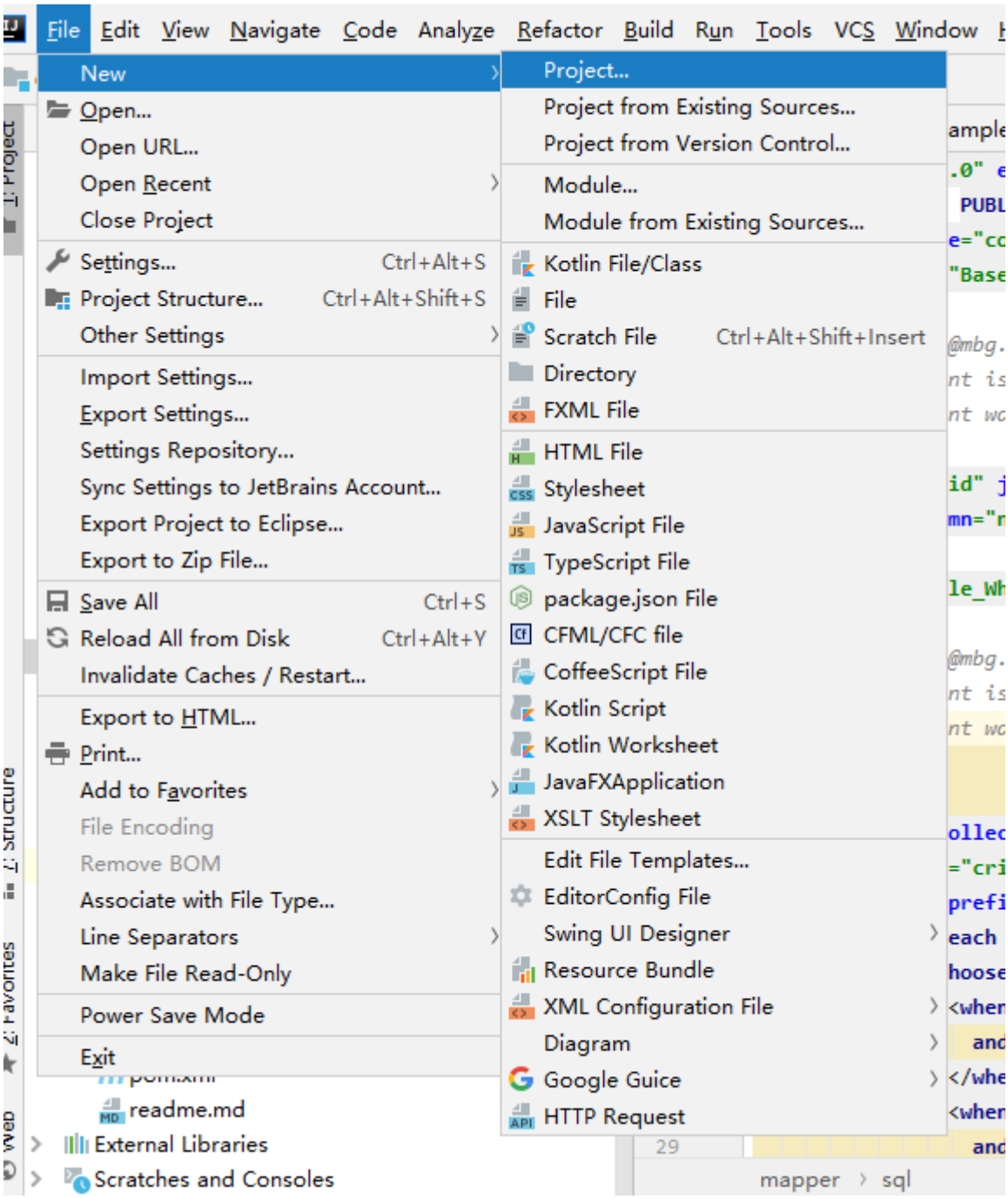
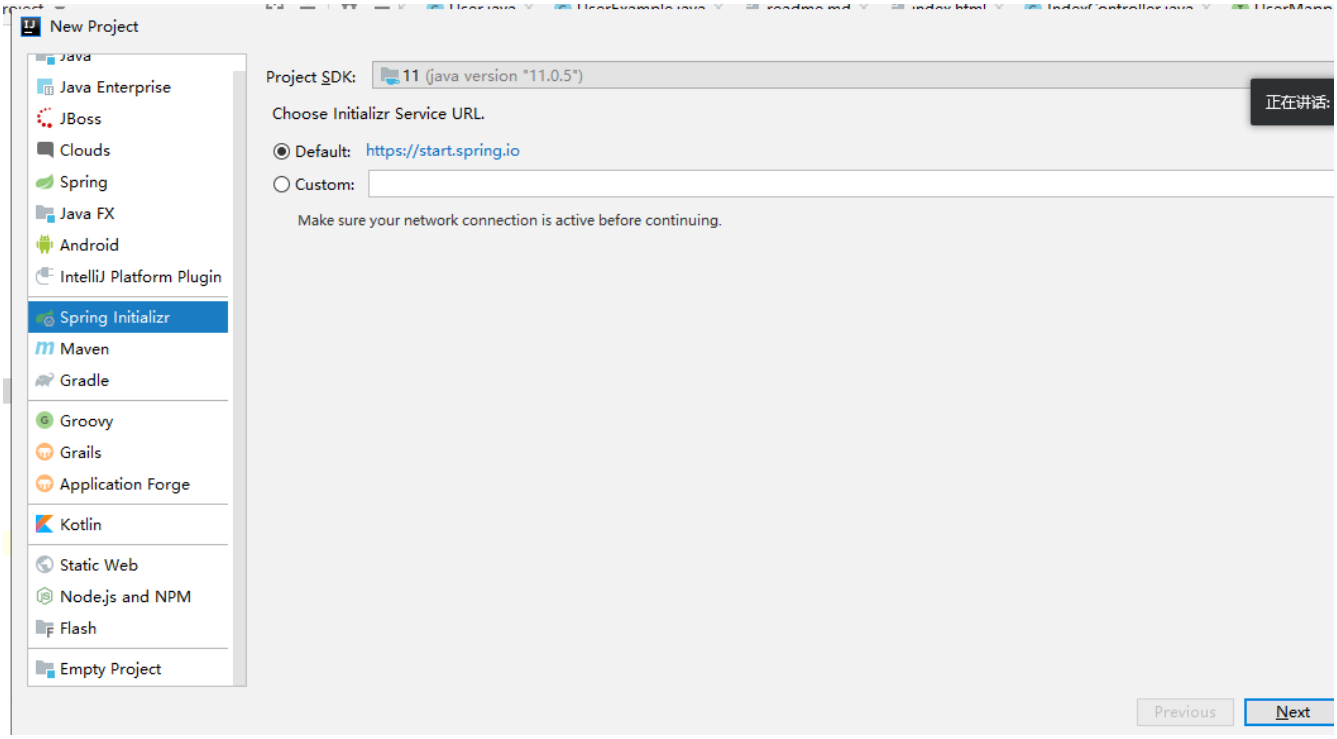
创建并运行spring boot项目

一、创建spring boot项目

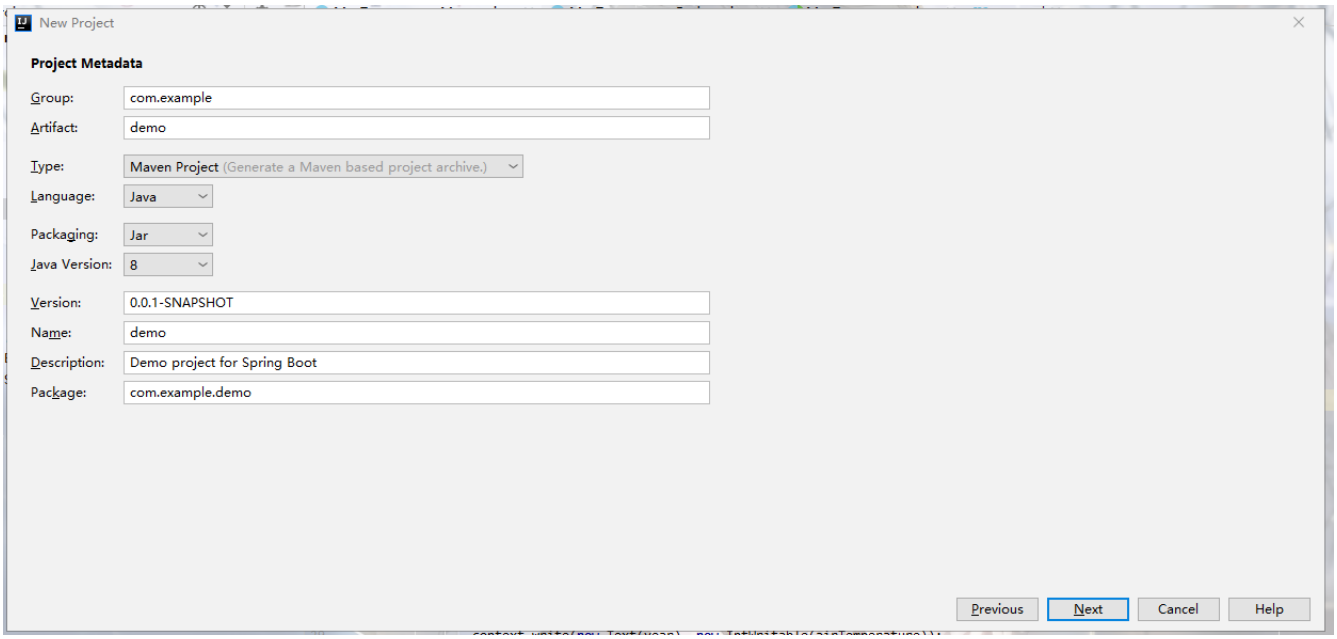
1、新建



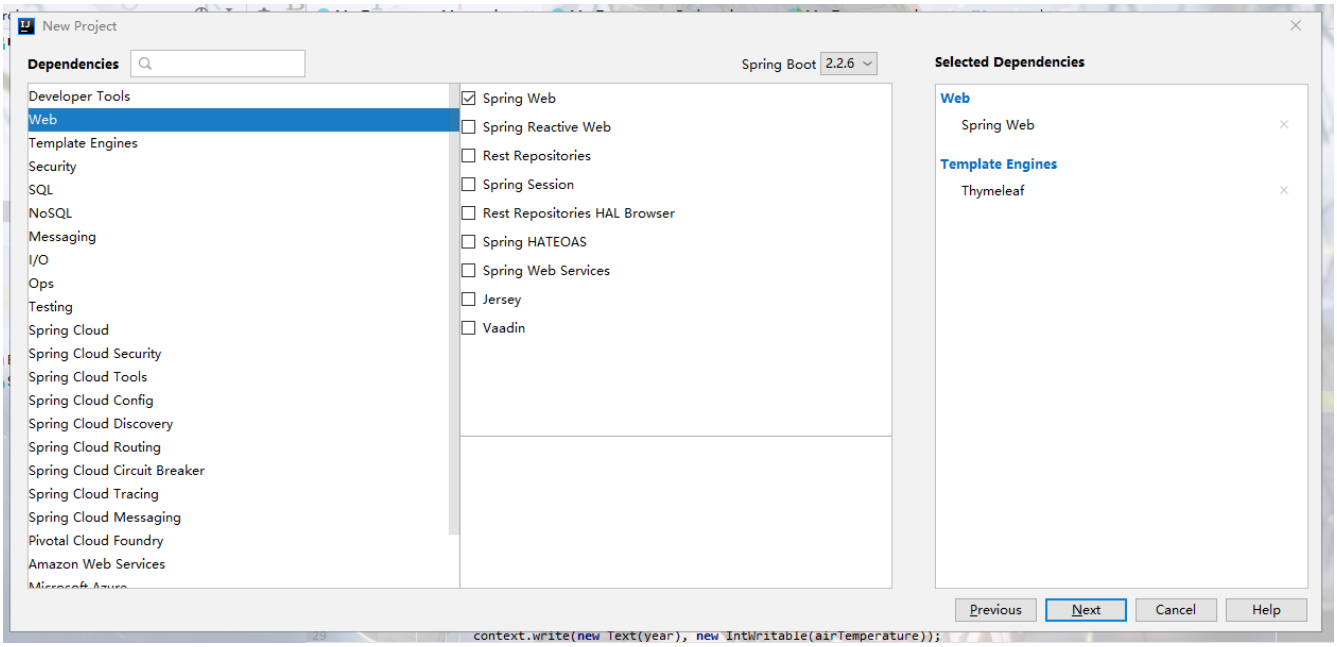
2、通过 spring initializr创建



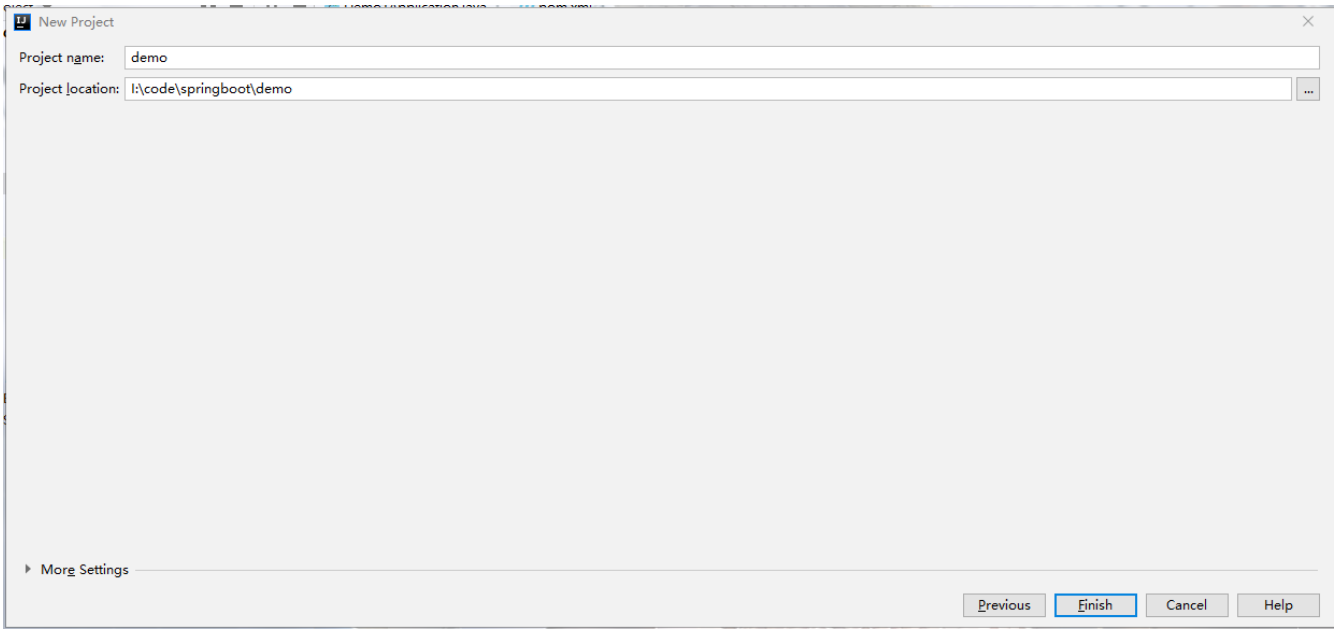
3、填写项目相关信息



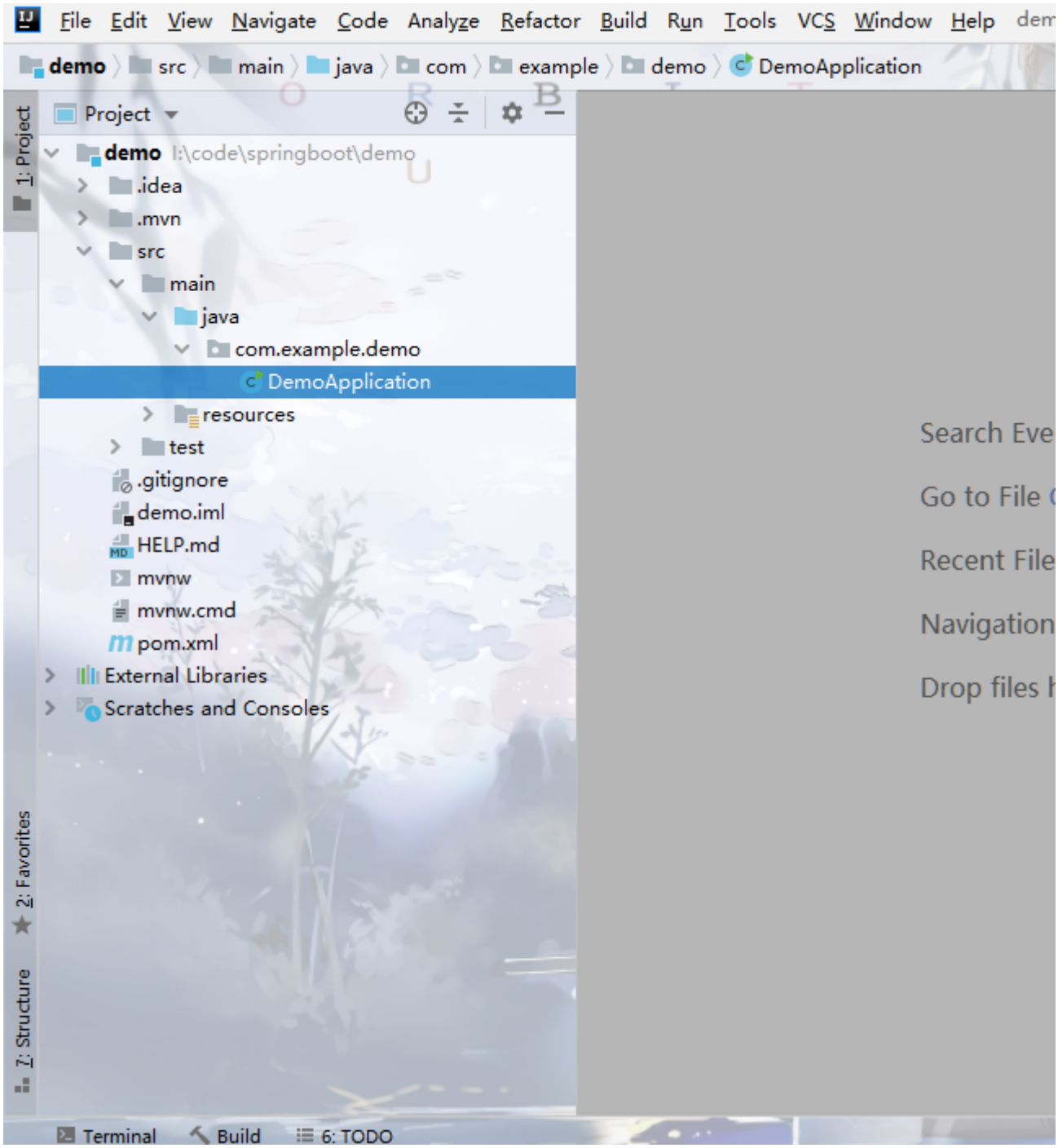
4、选择项目的相关组件（组件可以在后期的开发中自己添加）



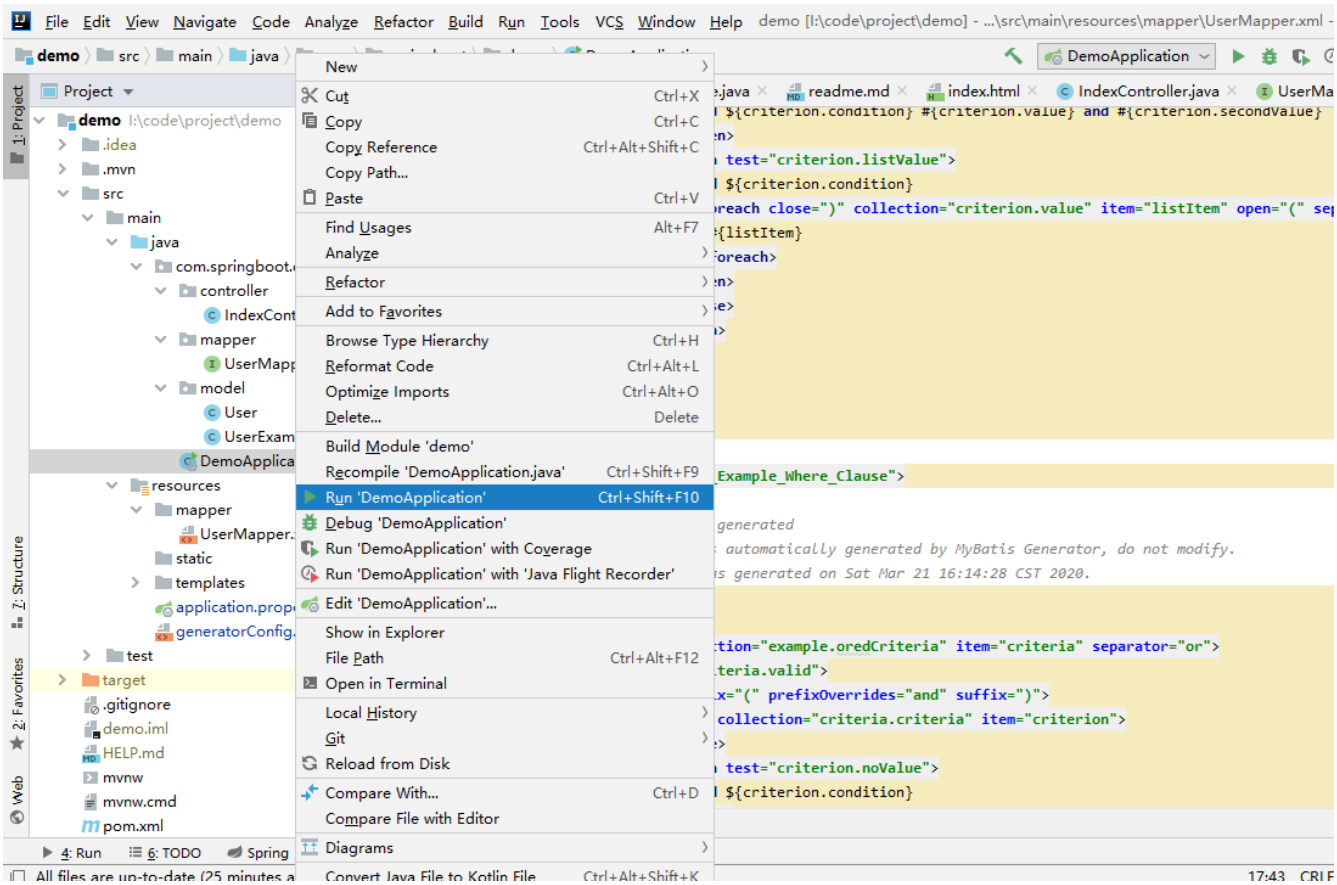
5、填写项目保存路径



6、创建项目成功



7、等待maven拉取相关依赖后吗，点击运行



二、配置项目

1、配置数据库

项目的配置文件是**application.properties**

```
#服务器运行端口
server.port=8090

#配置mysql驱动
spring.datasource.url=jdbc:mysql://127.0.0.1/demo?characterEncoding=UTF-8&serverTimezone=UTC
#数据库用户名
spring.datasource.username=monty
```

```
#数据库密码
spring.datasource.password=root
#数据库的驱动类
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect

#mybatis
mybatis.configuration.map-underscore-to-camel-case=true
mybatis.type-aliases-package=com.monty.community.model
mybatis.mapper-locations=classpath:mapper/*.xml
```

2、配置mybatis-generator

（前置条件：需要在数据库中创建数据库并且创建表）

在**pom.xml**文件中配置

1、添加依赖和plugin

```
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.19</version>
        </dependency>
        <!--        配置mybatis-->
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>2.1.1</version>
        </dependency>

        <plugin>
            <groupId>org.mybatis.generator</groupId>
            <artifactId>mybatis-generator-maven-plugin</artifactId>
            <version>1.3.7</version>
            <dependencies>
                <dependency>
                    <groupId>mysql</groupId>
                    <artifactId>mysql-connector-java</artifactId>
                    <version>${mysql.version}</version>
                </dependency>
            </dependencies>
        </plugin>
```

2、在resources文件夹下新建GeneratorConfig.xml用来配置mybatis-genenrator

```
#设置jdbc连接相关数据
  <jdbcConnection driverClass="com.mysql.cj.jdbc.Driver"
                        connectionURL="jdbc:mysql://127.0.0.1/demo"
                        userId="monty"
                        password="root">
            <!--MySQL 8.x 需要指定服务器的时区-->
            <property name="serverTimezone" value="UTC"/>
            <!--MySQL 不支持 schema 或者 catalog 所以需要添加这个-->
            <!--参考 : http://www.mybatis.org/generator/usage/mysql.html-->
```

```
            <property name="nullCatalogMeansCurrent" value="true"/>
            <!-- MySQL8默认启用 SSL ,不关闭会有警告-->
            <property name="useSSL" value="false"/>
        </jdbcConnection>

#指定生成的model类存放文件夹
        <javaModelGenerator targetPackage="com.springboot.demo.model"
targetProject="src\main\java">
            <property name="enableSubPackages" value="true" />
            <property name="trimStrings" value="true" />
        </javaModelGenerator>

#指定生成的mapper类的存放位置
        <sqlMapGenerator targetPackage="mapper"  targetProject="src\main\resources">
            <property name="enableSubPackages" value="true" />
        </sqlMapGenerator>
#指定生成的.xml文件的存放位置
        <javaClientGenerator type="XMLMAPPER" targetPackage="com.springboot.demo.mapper"
 targetProject="src\main\java">
            <property name="enableSubPackages" value="true" />
        </javaClientGenerator>
#指定model中的类和数据库中的表的对应关系
        <table   tableName="user" domainObjectName="User" >
        </table>
    </context>
</generatorConfiguration>
```

3、运行命令（命令行运行）

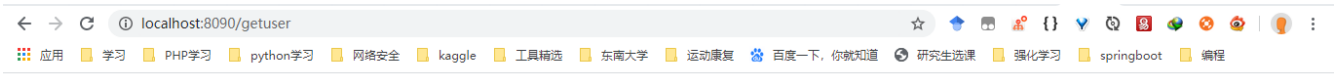
```
mvn -Dmybatis.generator.overwrite=true mybatis-generator:generate
```

三、前后端分离编程

1、在src目录下controller包，然后新建HelloController类

```
@Controller
public class IndexController {
    @Autowired
    private UserMapper userMapper;
    @GetMapping("/")
    public String index(){
        return "index";
    }
//前后端分离，使用ResponseBody注解，使用UserMapper从数据库中user表获取信息，将数据传递到页面
    @ResponseBody
    @RequestMapping(value = "/getuser")
    public List<User> getUser(){
        UserExample userExample=new UserExample();
        List<User> users=userMapper.selectByExample(userExample);
        return users;
    }
}
```

2、运行项目，访问接口



[{"id":1,"name":"y"},{"id":2,"name":"s"},{"id":3,"name":"wsdf"}]