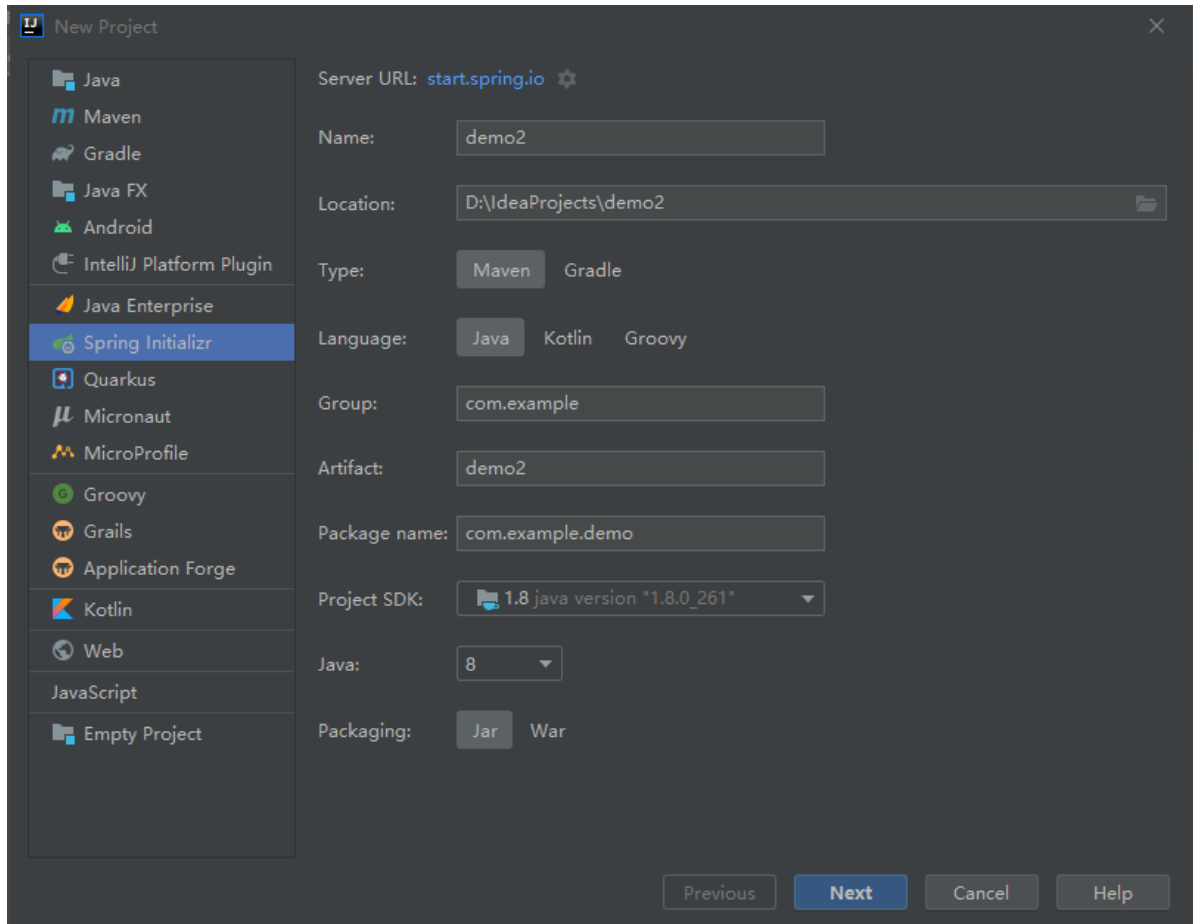


# MyBatis-Plus教程

---

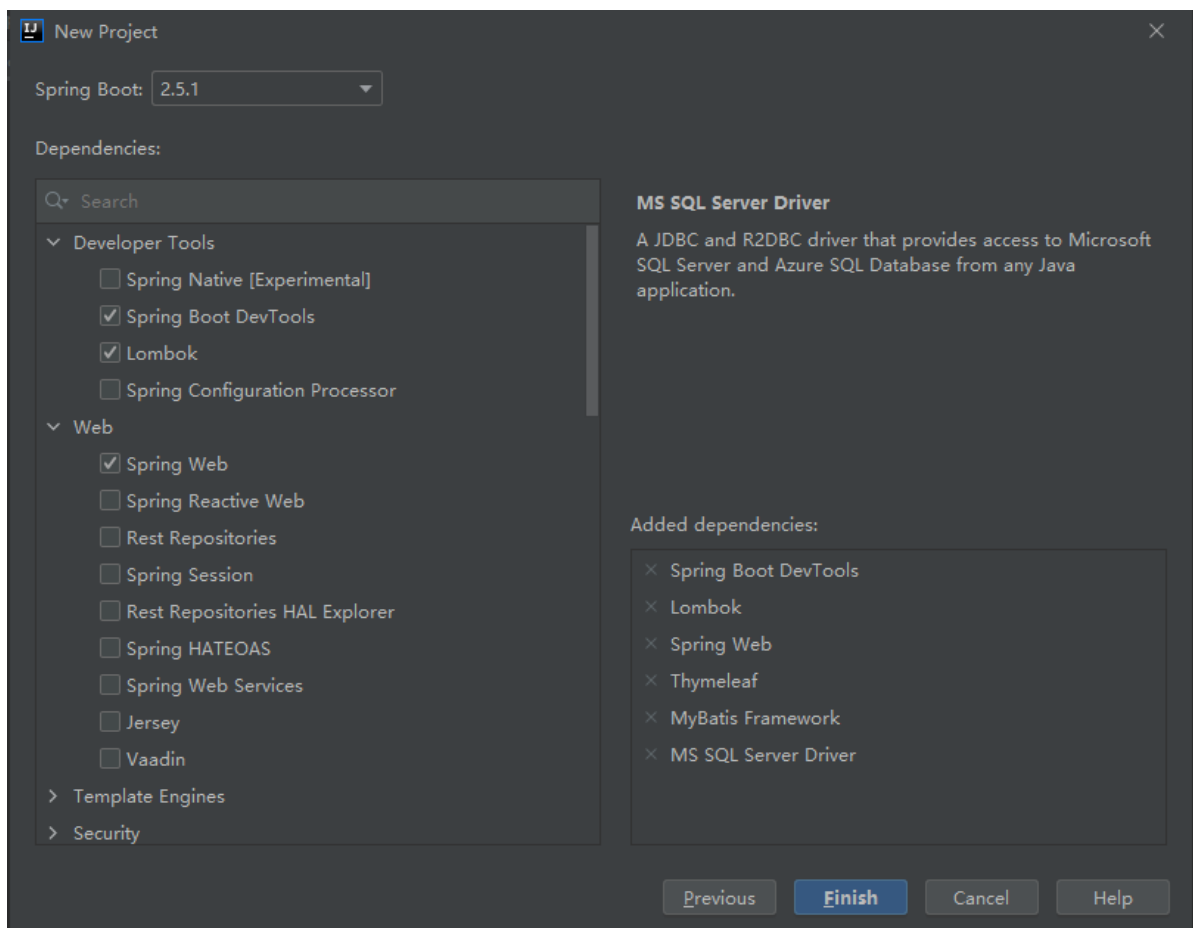
## 一、新建Spring Boot项目

---



## 二、添加依赖

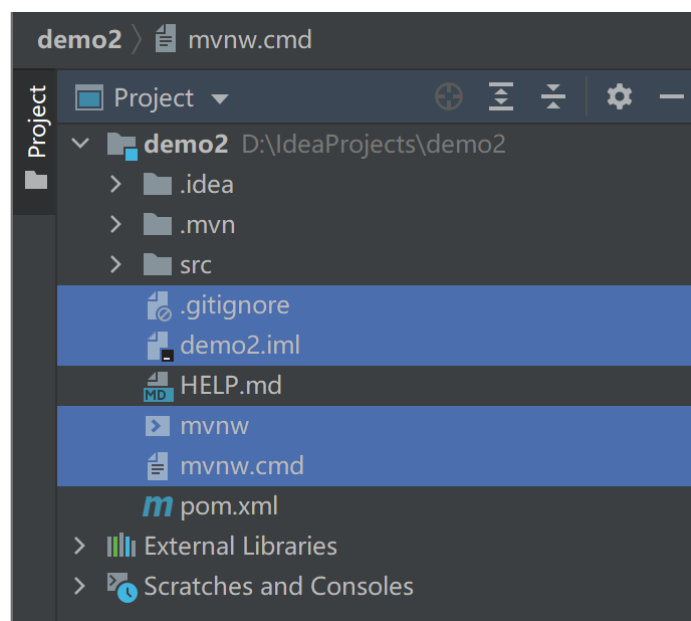
---



添加mybatis-plus依赖:

```
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-boot-starter</artifactId>
  <version>3.4.2</version>
</dependency>
```

删除多余文件:



### 三、 application.yml

```
server:
  port: 80

spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/address_book?serverTimezone=Asia/Shanghai
    username: root
    password: root
  thymeleaf:
    cache: false

mybatis:
  mapper-locations: classpath:mapper/**/*.xml
  configuration:
    map-underscore-to-camel-case: true
```

## 四、实体类

## 五、DemoApplication

```
package com.example.demo2;

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

@SpringBootApplication
public class Demo2Application {

    public static void main(String[] args) {
        SpringApplication.run(Demo2Application.class, args);
    }

}
```

## 六、测试应用

```
package com.example.demo2;

import com.example.demo2.entity.Person;
import com.example.demo2.mapper.PersonMapper;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import java.util.List;

@SpringBootTest
class Demo2ApplicationTests {

    @Autowired
    PersonMapper personMapper;

    @Test
    void contextLoads() {
```

```

        List<Person> persons = personMapper.selectList(null);
        persons.forEach(System.out::println);
    }
}

```

## 七、MyBatis的两种用法

MyBatis-Plus: 自定义SQL

### 1、注解用法

PersonMapper.java

```

package com.example.demo2.mapper;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import com.example.demo2.entity.Person;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import java.util.List;

@Mapper
public interface PersonMapper extends BaseMapper<Person> {
    @Select("SELECT * FROM person WHERE person_name LIKE '%三%'")
    List<Person> selectPerson();
}

```

### 2、XML用法

PersonMapper.java

```

package com.example.demo2.mapper;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import com.example.demo2.entity.Person;
import org.apache.ibatis.annotations.Mapper;
import java.util.List;

@Mapper
public interface PersonMapper extends BaseMapper<Person> {
    List<Person> selectPerson();
}

```

PersonMapper.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.demo2.mapper.PersonMapper">
    <select id="selectPerson" resultType="com.example.demo2.entity.Person">
        SELECT * FROM person WHERE person_name LIKE '%三%'
    </select>
</mapper>

```

## 八、业务类

PersonService.java

```
package com.example.demo2.service;

import com.example.demo2.entity.Person;

import java.util.List;

public interface PersonService {
    List<Person> selectPerson();
}
```

PersonServiceImpl.java

```
package com.example.demo2.service;

import com.example.demo2.entity.Person;
import com.example.demo2.mapper.PersonMapper;
import org.springframework.stereotype.Service;

import java.util.List;
@Service
public class PersonServiceImpl implements PersonService {
    final
    PersonMapper personMapper;

    public PersonServiceImpl(PersonMapper personMapper) {
        this.personMapper = personMapper;
    }

    @Override
    public List<Person> selectPerson() {
        return personMapper.selectPerson();
    }
}
```

测试:

```
package com.example.demo2;

import com.example.demo2.entity.Person;
import com.example.demo2.mapper.PersonMapper;
import com.example.demo2.service.PersonService;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import java.util.List;

@SpringBootTest
class Demo2ApplicationTests {
    @Autowired
    PersonService personService;

    @Test
    void contextLoads() {
    }
}
```

```

@Test
void contextLoads() {
    List<Person> persons = personService.selectPerson();
    persons.forEach(System.out::println);
}
}

```

## 九、控制器

PersonController.java

```

package com.example.demo2.controller;

import com.example.demo2.entity.Person;
import com.example.demo2.service.PersonService;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import java.util.List;

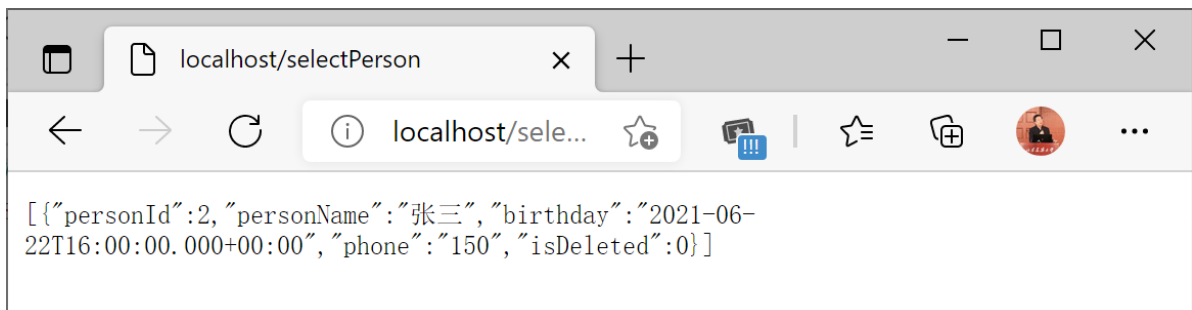
@Controller
public class PersonController {
    final
    PersonService personService;

    public PersonController(PersonService personService) {
        this.personService = personService;
    }

    @RequestMapping("/selectPerson")
    @ResponseBody
    public List<Person> selectPerson(){
        return personService.selectPerson();
    };
}

```

在浏览器输入网址: [localhost/selectPerson](http://localhost/selectPerson)



访问网页，修改控制器如下：

```

package com.example.demo2.controller;

import com.example.demo2.service.PersonService;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;

```

```

import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class PersonController {
    final
    PersonService personService;

    public PersonController(PersonService personService) {
        this.personService = personService;
    }

    @RequestMapping("/selectPerson")
    public String selectPerson(Model model){
        model.addAttribute("persons", personService.selectPerson());
        return "person";
    }
}

```

## 十、网页person.html

```

<!DOCTYPE html>
<html lang="zh" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>通讯录</title>
    <style>
        table {
            border-collapse: collapse;
            margin: auto;
        }
        th,
        td {
            border: chocolate solid 1px;
            padding: 5px;
        }
    </style>
</head>
<body>
    <table>
        <tr>
            <th>姓名</th>
            <th>生日</th>
            <th>电话</th>
        </tr>
        <tr th:each="person: ${persons}">
            <td th:text="${person.personName}"></td>
            <td th:text="${#dates.format(person.birthday, 'yyyy年MM月dd日')}">
</td>
            <td th:text="${person.phone}"></td>
        </tr>
    </table>
</body>
</html>

```

通讯录		
localhost/selectPerson		
姓名	生日	电话
张三	2021年06月23日	150