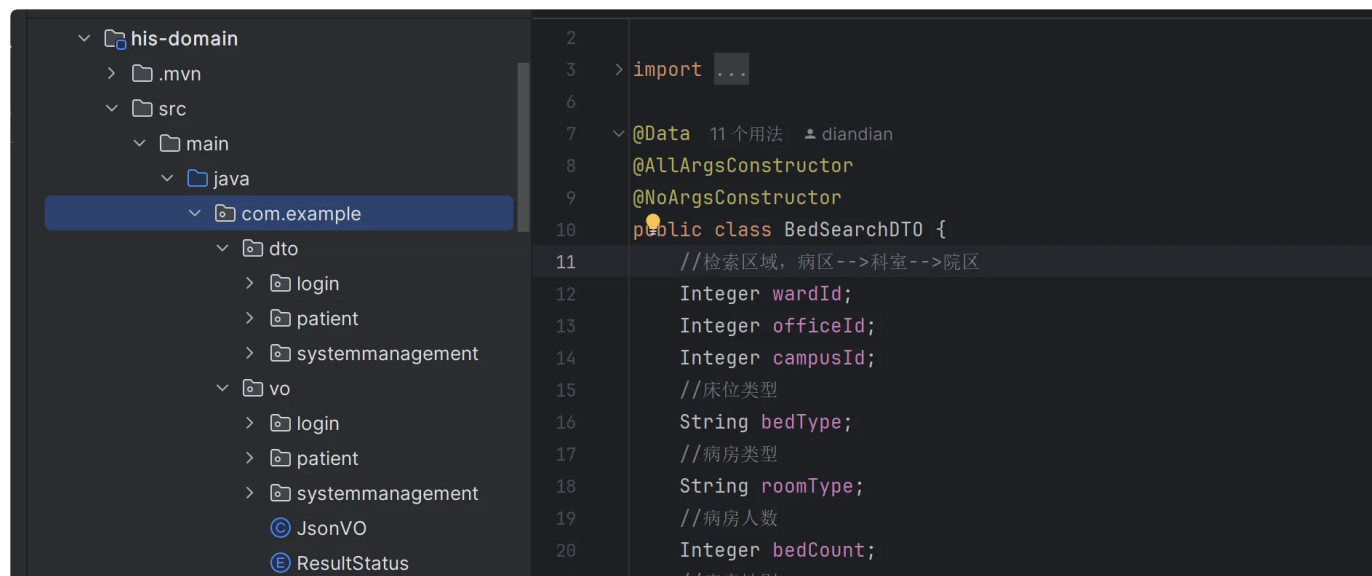


后端开发注意事项

关于dto、vo的定义都要在his-domain这个模块中进行，之后想要调用，都直接引入这个模块的依赖



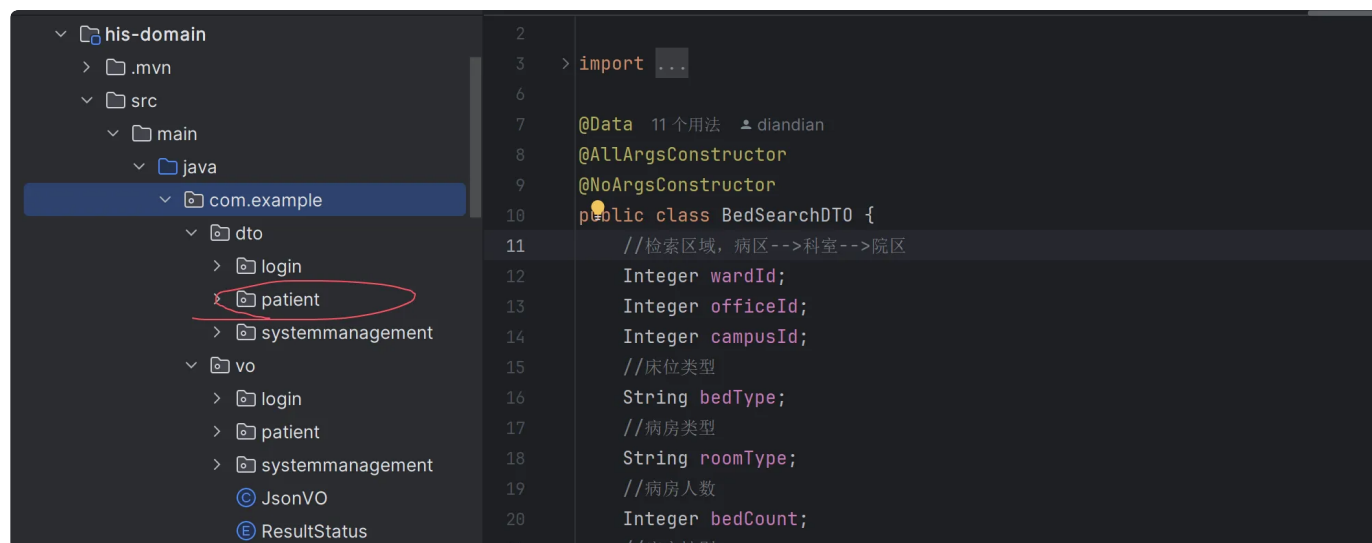
The screenshot shows an IDE with a project structure on the left and a Java class definition on the right. The project structure is as follows:

- his-domain
 - .mvn
 - src
 - main
 - java
 - com.example
 - dto
 - login
 - patient
 - systemmanagement
 - vo
 - login
 - patient
 - systemmanagement
 - JsonVO
 - ResultStatus

The Java class definition on the right is as follows:

```
2
3 > import ...
6
7 @Data 11 个用法 diandian
8 @AllArgsConstructor
9 @NoArgsConstructor
10 public class BedSearchDTO {
11     //检索区域, 病区-->科室-->院区
12     Integer wardId;
13     Integer officeId;
14     Integer campusId;
15     //床位类型
16     String bedType;
17     //病房类型
18     String roomType;
19     //病房人数
20     Integer bedCount;
21     //病房性别
```

根据自己的模块定义一个包，在里面创建dto或vo



The screenshot shows an IDE with a project structure on the left and a Java class definition on the right. The project structure is as follows:

- his-domain
 - .mvn
 - src
 - main
 - java
 - com.example
 - dto
 - login
 - patient
 - systemmanagement
 - vo
 - login
 - patient
 - systemmanagement
 - JsonVO
 - ResultStatus

The Java class definition on the right is as follows:

```
2
3 > import ...
6
7 @Data 11 个用法 diandian
8 @AllArgsConstructor
9 @NoArgsConstructor
10 public class BedSearchDTO {
11     //检索区域, 病区-->科室-->院区
12     Integer wardId;
13     Integer officeId;
14     Integer campusId;
15     //床位类型
16     String bedType;
17     //病房类型
18     String roomType;
19     //病房人数
20     Integer bedCount;
21     //病房性别
```

所有的接收值都要通过dto来接收，返回值则是通过vo返回，将自己定义vo传入到JsonVO的方法中返回

```
@PostMapping("/msg/add")
public JsonVO<Object> AddUser(@RequestBody UsersAddDTO usersAddDTO) {
    UsersAddVO vo=iUsersService.insertUser(usersAddDTO);
    if(vo!=null){
        return JsonVO.success(vo);
    }else{
        return JsonVO.fail(vo);
    }
}
```