负能量厕所创业孵化项目-commons-core基础搭建-v2.0

文档作者: 张初刚

创建日期： 06.20 2019

更新日期：

文档编码：

当前版本： 2.0

拷贝编号：V1.0

文档控制

**变更记录**

| 日期 | 作者 | 版本 | 变更参考 |
| --- | --- | --- | --- |
|  |  |  |  |
| 2019-06-20 | 张初刚 | 1.0 | 原始文档 |
| 2019-06-28 | 张初刚 | 2.0 | 变更 |
|  |  |  |  |

**审核**

| 姓名 | 职位 |
| --- | --- |
|  |  |
|  |  |
|  |  |

**分发**

| 拷贝编号 | 姓名 | 位置/岗位 |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**备注:**

出于文档管理的目的，如果您收到了本文档的电子版本，请打印出来并在封面的相应位置写上您的名字。

出于文档管理的目的，如果您收到了本文档的纸介质版本，请在封面写上您的名字。

目录

[1、基础功能 4](#_Toc12622031)

[2、核心内容 4](#_Toc12622032)

[3、获取方式 5](#_Toc12622033)

[4、在其他微服务中引入 6](#_Toc12622034)

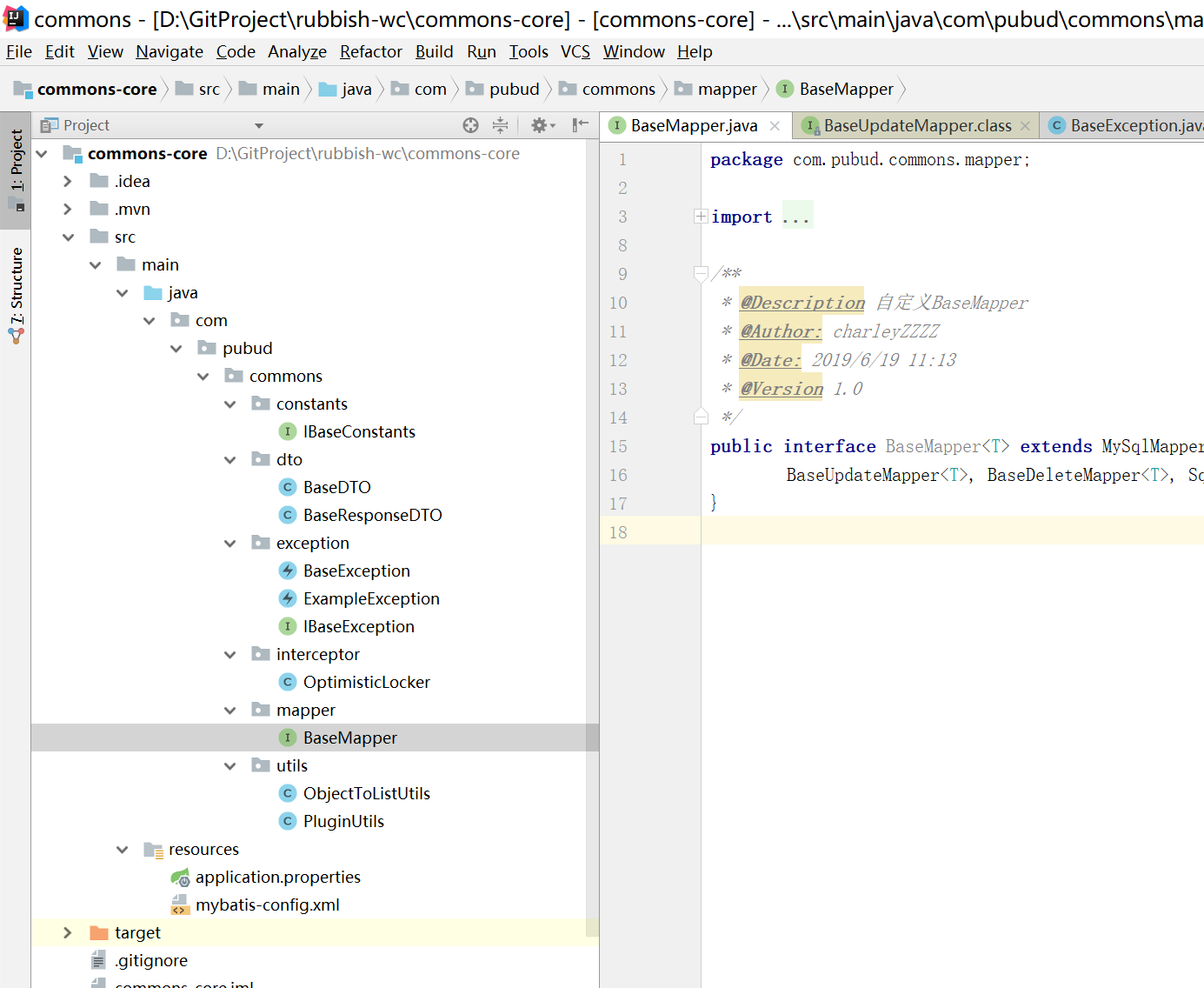
[5、业务微服务示例 6](#_Toc12622035)

[6、0628 补充异常捕捉处理 7](#_Toc12622036)

## 1、基础功能

**为了代码健壮性，我们将提取代码公共部分作为jar包供其他微服务使用**

**目前将其命名为 commons-core，包名com.pubud.commons**

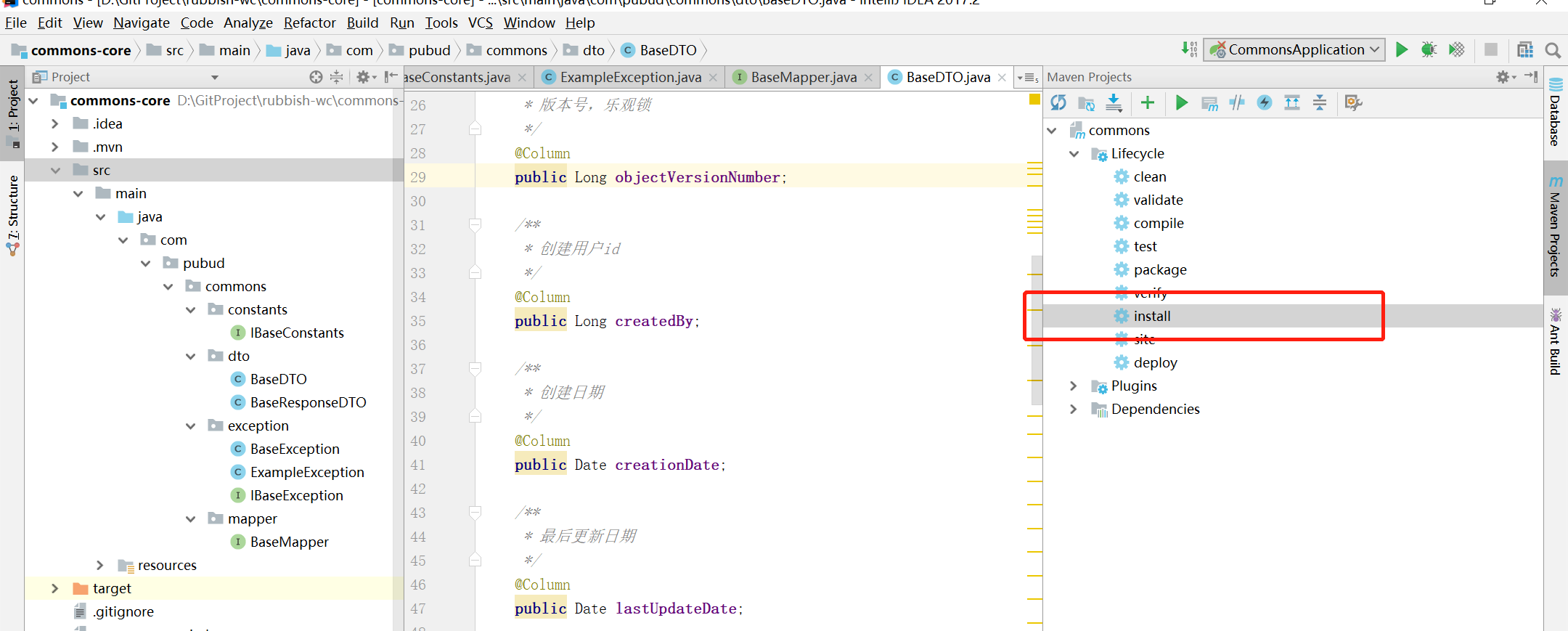


## 2、核心内容

* 常量接口 IBaseConstants, 包含基本常亮
* 基础DTO ：BaseDTO，内容包含 基础字段creationDate 、createdBy 、lastUpdateDate、lastUpdatedBy以及用于乐观锁控制的objectVersionNumber，建表时必须将这些字段保留（可用建表工具，我这有），乐观锁通过自定义mybatis拦截器已实现
* 基础响应DTO：BaseResponseDTO，封装了Controller 对外返回数据格式，支持code、message、List<Object>
* 异常通用类和接口，BaseException\IBaseException ，项目可以自定义Exception继承BaseException，异常捕捉类待补充，思路是自定义BaseController进行web层异常捕捉
* 持久化接口：BaseMapper，继承自tk.mybatis，使用中只要将自己的mapper继承自BaseMapper即可，支持基本增删改查功能，批量插入功能，批量更新删除得自己写，后面再补充

## 3、获取方式

1. git clone https://github.com/rubbish-wc/commons-core.git
2. maven install 本地仓库安装



## 4、在其他微服务中引入

<**dependency**>  
 <**groupId**>com.pubud</**groupId**>  
 <**artifactId**>commons-core</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
</**dependency**>

添加commons-core 功能，目前只提供这些功能

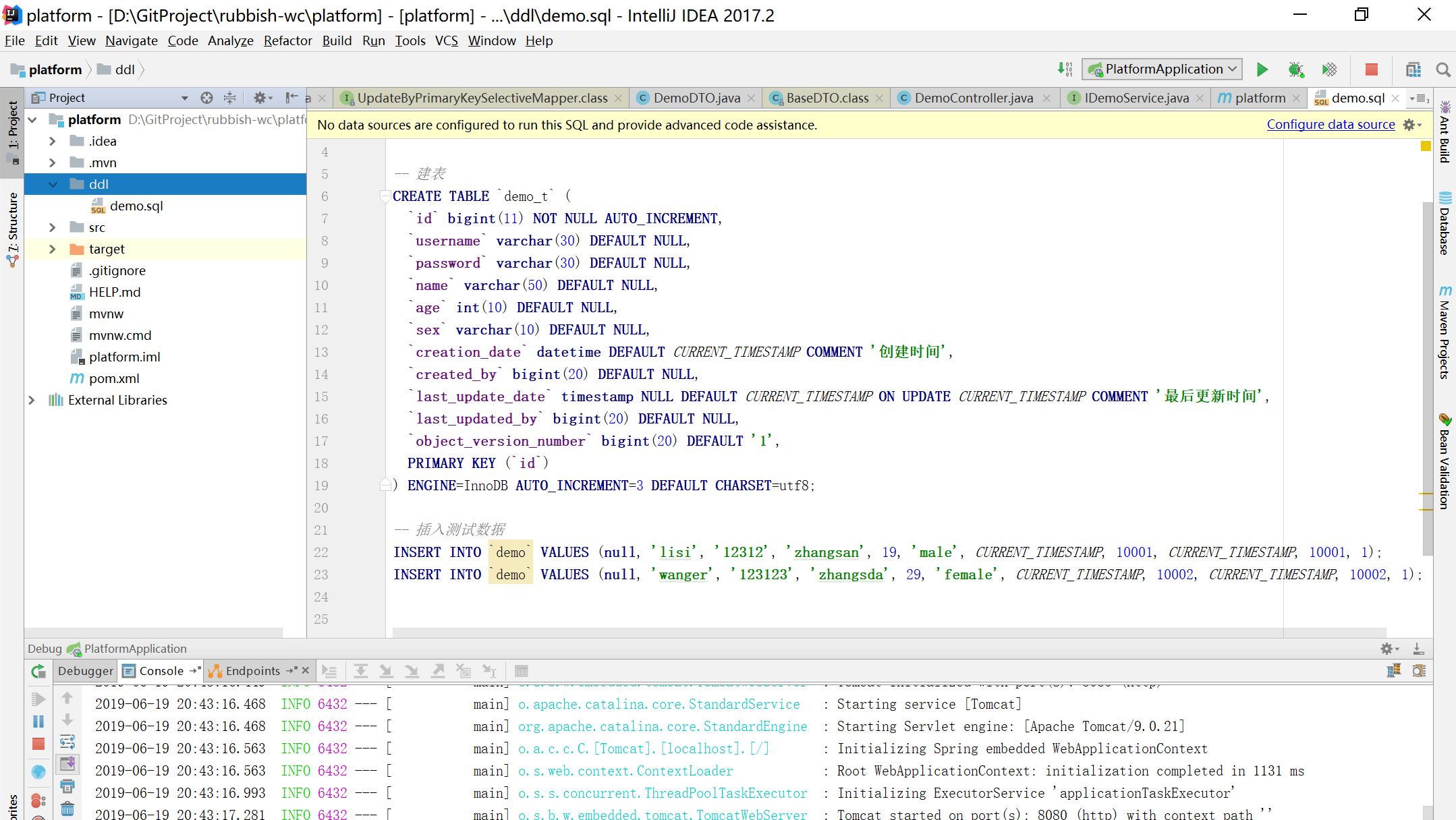
后期希望各位大佬向其添加新的功能，只要将代码提交，重新clean install 即可

## 5、业务微服务示例

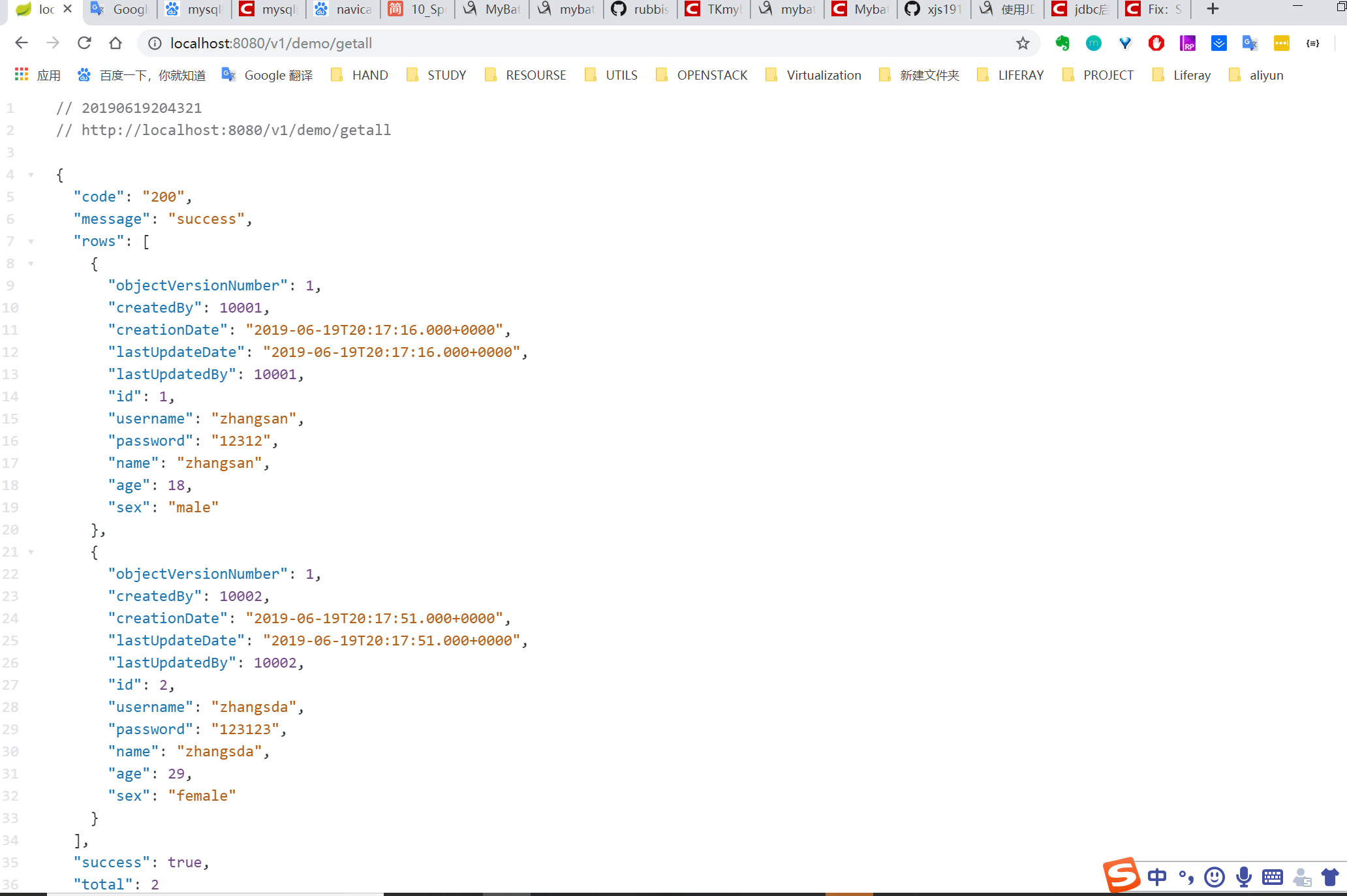
这里准备了一个 platform微服务，作为引入commons-core模块测试

Git地址：https://github.com/rubbish-wc/test\_platform.git

DDL脚本位置：￥{project}/ddl

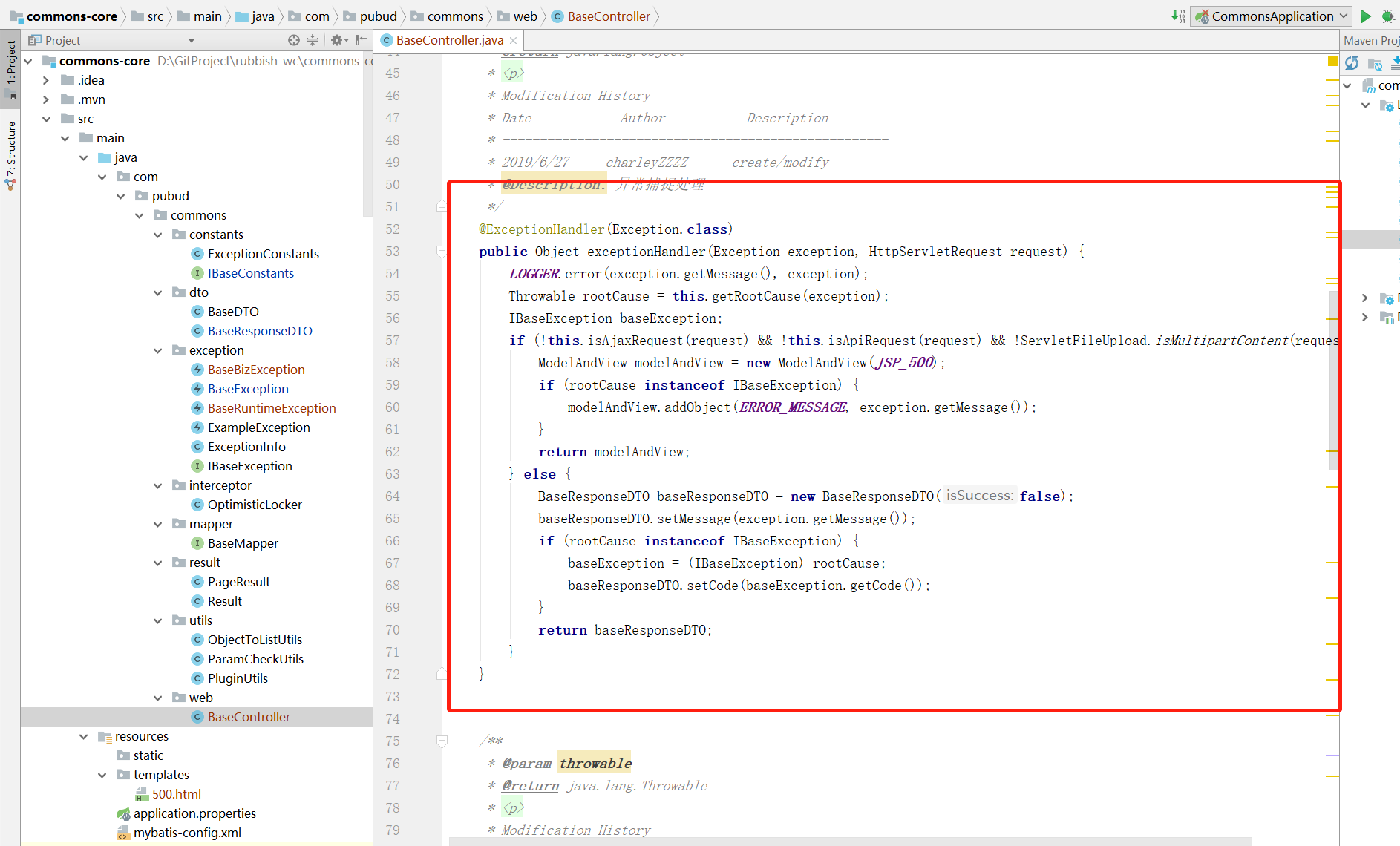


测试结果：



## 6、0628 补充异常捕捉处理

新增 BaseController 添加异常捕捉方法：



### 异常捕捉分类

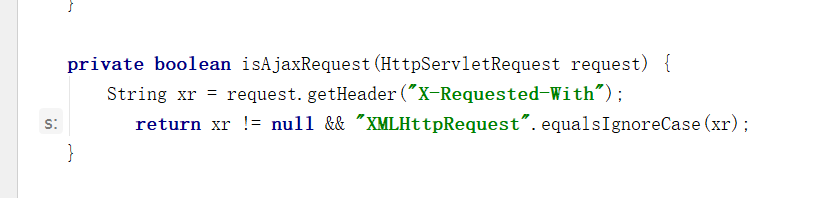
一类是自定义异常，即IBaseException及其子类，BaseException实现，子类分为BaseBizException(业务异常)，BaseRuntimeException（运行时异常），及其他异常（待补充），这部分异常可以获取 异常代码，和异常消息，又开发人员抛出。

另一类是其他异常，无法获取异常代码

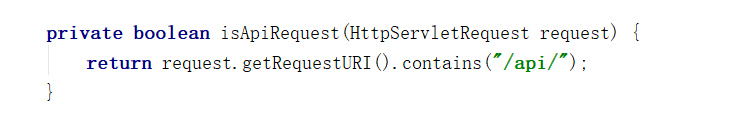
### 异常处理分类

第一类为：ajax，api，文件请求，返回 BaseResponseDTO ，判断依据如下

* + 判断ajax请求



* + 判断api请求



我们将约定所有api请求前缀为“/api”

* + 判断文件上传



第二类为非上述的类别，返回ModelAndView

### 使用方法

* Controller 继承 BaseController
* 业务层捕捉异常，抛IBaseException至Controller层，Controller方法作最后处理，抛出，由BaseController 异常捕捉方法捕捉，返回给客户端