# 开发工具

## 软件列表

http://10.1.113.11:8080/pack/

下载下列软件，解压/安装到D盘Soft目录

|  |  |
| --- | --- |
| jdk1.8.0\_101.7z | 使用JDK1.8 |
| Git-2.11.0-64-bit.exe | 代码版本管理工具 |
| TortoiseGit-2.3.0.0-64bit.msi | Git图形化工具（IT小助手或热线安装） |
| IDEA2017C1.7z | 开发工具，替代eclipse |
| HeidiSQL\_9.4\_Portable.zip | MySQL数据库客户端 |
| gradle-3.0-bin.zip | 项目打包工具 |
| MobaXterm\_v94.7z | Linux连接/文件传输工具 |
| VSCode.7z | 文本编辑器(或使用UE) |
| restclient-3.6.1.zip | 接口测试工具（或自行安装PostMan） |

## 环境配置

### 环境变量

增加**用户环境变量**

|  |  |
| --- | --- |
| JAVA\_HOME | D:\Soft\jdk1.8.0\_101 |
| GRADLE\_HOME | D:\Soft\gradle-3.0 |
| GRADLE\_OPTS | -Dfile.encoding=UTF8 |
| Path | %GRADLE\_HOME%\bin;%JAVA\_HOME%\bin;D:\Soft\Git\cmd |

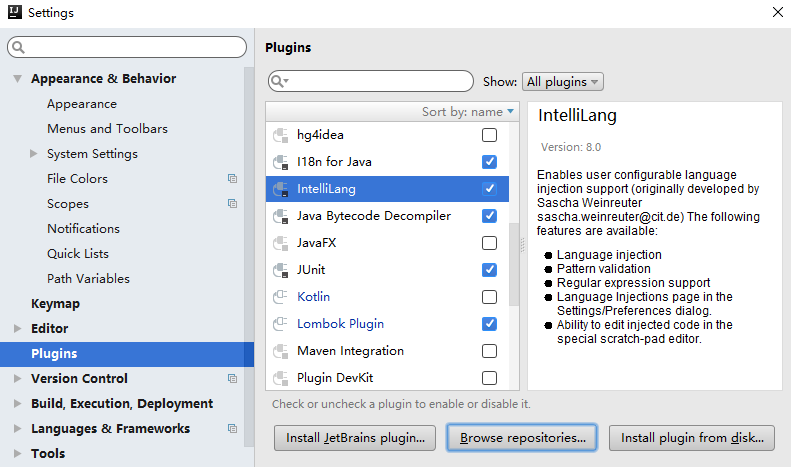
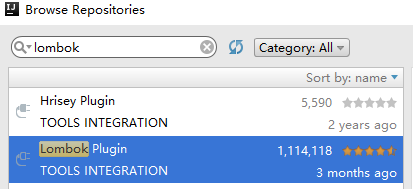
### Git用户名

打开命令行窗口，依次输入命令

git config --global user.name "**你的中文名字**"

git config --global user.email **[你的邮箱@banketech.com](mailto:你的邮箱@banketech.com)**

### 安装Lombok插件

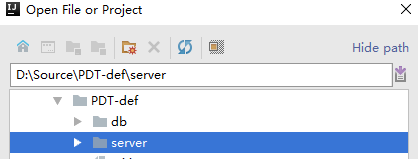
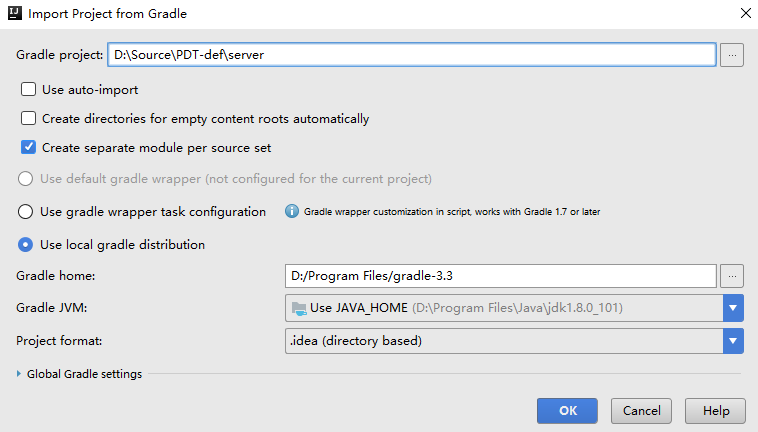
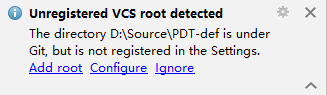
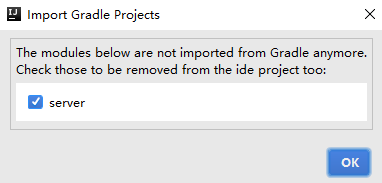
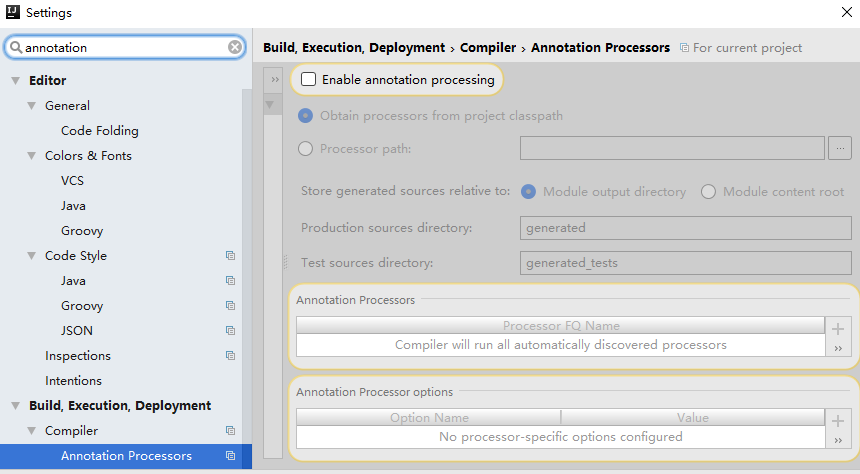
打开idea(bin/idea64.exe)，Settings… -> Plugins -> Browse repositories…  
  
搜索Lombok Plugin，点击Install安装  


# 开发流程

## 下载项目

1. 使用开机账号登录地址http://git.banketech.com:3000
2. 找到需要下载的微服务项目
3. D盘创建Source文件夹，在文件夹空白处按住shift右键，选择“在此处打开命令窗口”
4. 输入命令git clone –b 217xxxx <http://git.banketech.com:3000/banke/XXX-xxxxx.git>下载分支代码

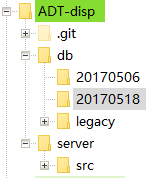
## 导入项目

1. 打开idea(bin/idea64.exe)
2. Open…
3. 选择项目下的**server**目录  
   
4. 选择本地Gradle目录，并配置JDK  
   
5. 右下角弹出git提示，选择Add root  
   
6. 弹出gradle项目导入，点oK  
   
7. 打开Settings… -> 搜索Annotation，勾选Enable annotation processing  
   

## 运行项目

1. 找到最外层的App.java
2. 点击右键， DEBUG 'App.main()'  
   

## 提交代码

1. 更新&提交代码：右上角按钮  
   
2. DB脚本文件，放在项目根目录db目录下，建立版本号文件夹存放  
   

## 打开多个工程

直接Open其他项目，每个项目都是独立窗口

## 项目打包

通过部署平台统一打包部署，禁止私自上传、修改测试环境文件及数据库

## 快速开发

|  |  |
| --- | --- |
| @Data | Class注解，不用写get/set |
| @Slf4j | Class注解，代码内可直接使用log.info等 |

## 配置文件

使用yml格式文件，语法详情自行搜索。**新项目请修改所有标记了TODO的字段**。

|  |  |
| --- | --- |
| bootstrap.yml | 基本配置文件，环境无关，禁止新增内容 |
| application-dev.yml  application-sit.yml  application-int.yml | 项目配置文件，区分开发/测试/生产环境，自定义配置放在这里 |

## 单元测试

# 应用部署

# 附1：开发规范

**请严格遵守，遇到特殊场景的单独提出。**

**微服务：**

* + 项目名称（微服务名）：系统名-服务名，系统名大写，模块名驼峰。例如：CIF-ClientInfo
  + Java包名：com.banke.系统名.服务名。例如：com.banke.cif.clientinfo
  + 接口名以/api/ 开头，驼峰规则，无后缀。例如： /api/client/createClient
  + API层只负责校验，业务逻辑、外部接口、数据库必须在service层
  + API层返回类型应使用ResponseInfo。对于成功的请求，返回ResponseInfo.success
  + DAO层禁止自己写SQL查询，简单查询参考附录2，复杂查询、分页参考附录3
  + 字段、类、方法驼峰命名，不使用下划线。其中类名首字母大写
  + 类、方法、字段注释，必须使用JavaDoc格式**/\*\* 内容 \*/**，写在注释对象上面，内容前后空格

**数据库：**

* + 表名小写下划线分割，长度不超过30。例如：client\_info, client\_address\_info
  + 基表用于表示分类且数量有限的编码表，除特殊情况外，统一放在base\_code表中
  + 字段名小写下划线分割，长度不超过。例如：client\_no, birth\_date
  + 所有表主键统一使用数字类型定义，必须包含以下字段(已在BasePO中定义)

id **BIGINT**(15) **ZEROFILL** **primary** **key** **AUTO\_INCREMENT**,

// 其他字段定义

created\_by **varchar**(30) **not** **null** **comment** '创建人',

created\_date **datetime** **not** **null** DEFAULT CURRENT\_TIMESTAMP **comment** '创建时间',

updated\_by **varchar**(30) **not** **null** **comment** '最后修改人',

updated\_date **datetime** **not** **null** DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP **comment** '最后修改时间'

* **禁止在脚本中使用drop、truncate等批量删除语句**
* JAVA-DB字段类型关系如下表，禁止使用其他类型

|  |  |  |
| --- | --- | --- |
| **JAVA** | **MySQL** | **说明** |
| String | VARCHAR |  |
| Integer | INT | 最大值21亿 |
| Long | BIGINT |  |
| Double | DOUBLE |  |
| BigDecimal | DECIMAL | 金额专用 |
| Boolean | TINYINT(1) |  |
| java.time.LocalDateTime | DATETIME | 包含日期时间 |
| java.time.LocalDate | DATE | 不含时间 |