

# 协作：开发规范

---

## 一、目录

- 1、目录结构规范
- 2、目录命名规范

## 二、基础

### 1、编码规范

命名风格

常量命名

路径规范

git提交规范

### 2、数据库规范

建表规范

公共字段

## 三、流程

### 1、代码生成

### 2、功能开发

后台

前台

### 3、功能更新

## 一、目录

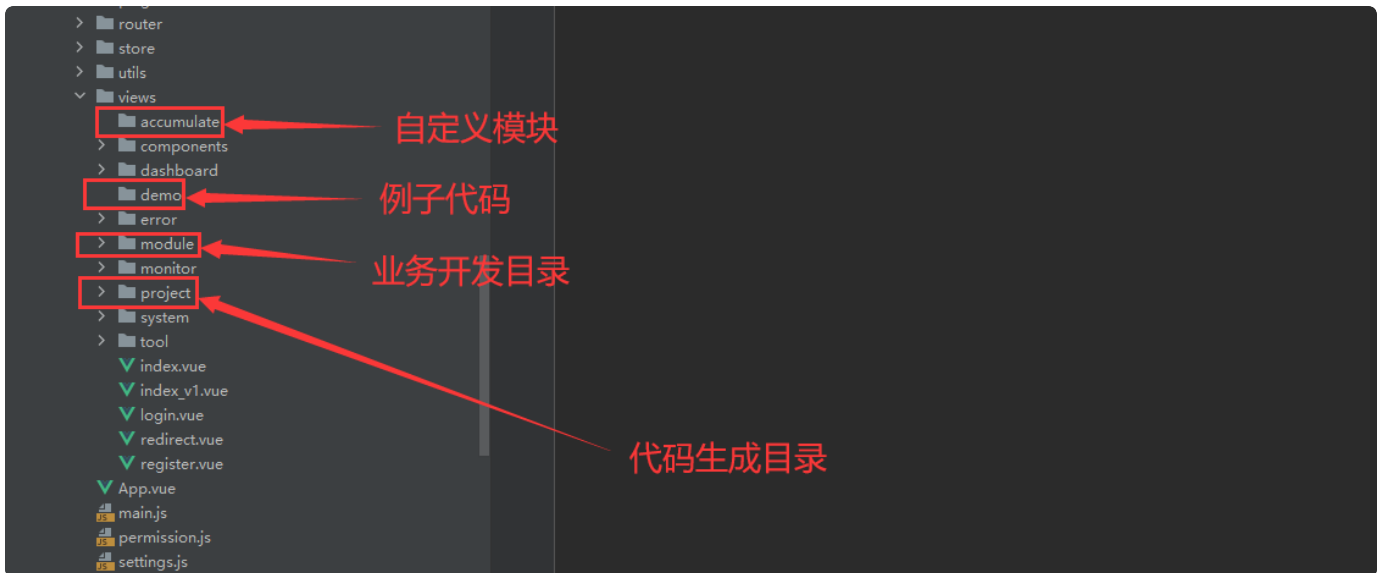
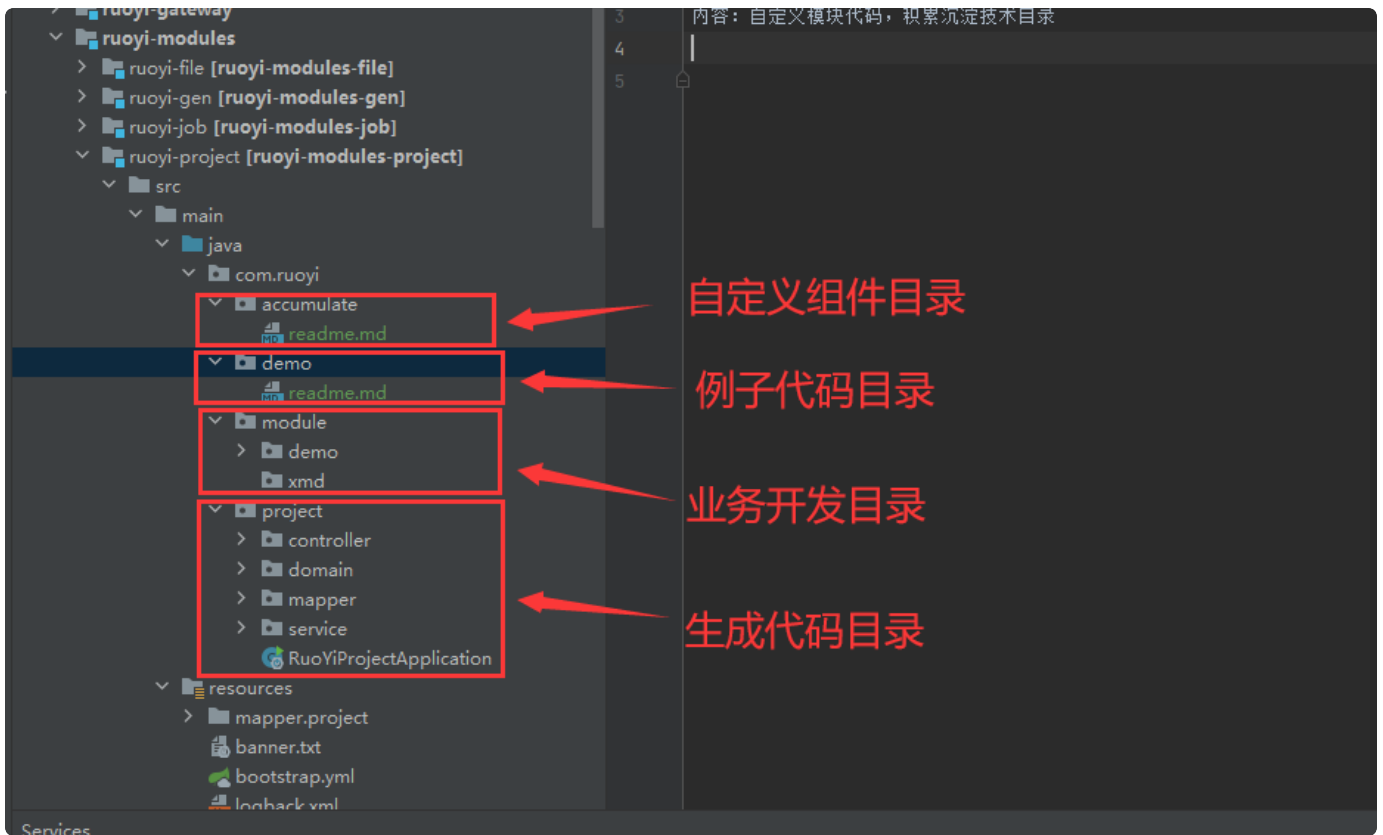
### 1、目录结构规范

accumulate：自定义组件目录

demo：例子代码目录

module：业务开发目录

project：代码生成目录



## 2、目录命名规范

模块名：ruoyi-project

开发目录：module

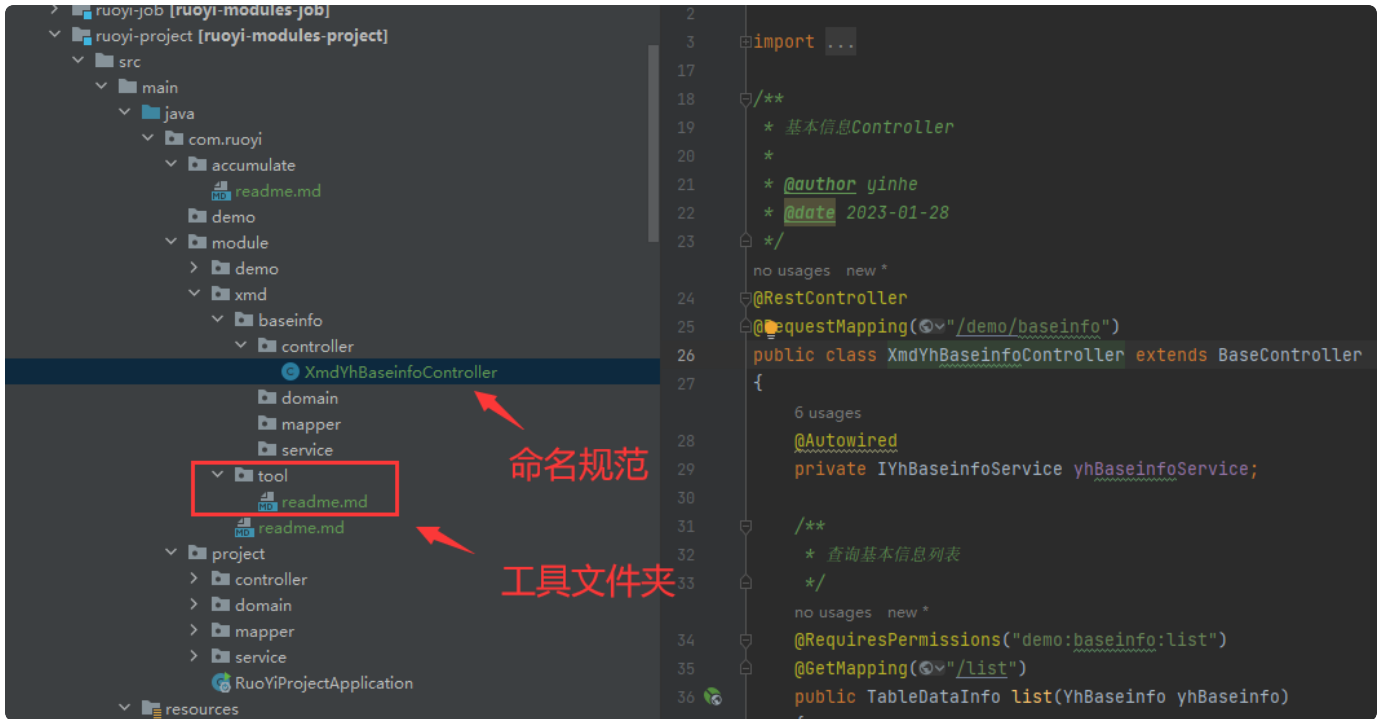
个人文件夹：一般为个人姓名拼音，例如：demo、xinmindong、niuyong .....

功能名称：一般为表名，例如：baseinfo .....

代码逻辑：controller、domain、mapper、service .....

文件命名规则：Demo（个人名称拼音）YhBaseinfo（表名）Controller（代码逻辑名）

其他说明：其他工具类建立在个人名称文件夹下 tool 文件夹内，命名规范为：个人名称拼音+工具名称



## 二、基础

### 1、编码规范

#### 命名风格

强制类名使用UpperCamelCase风格，必须遵守驼峰形式

强制方法名，参数名，成员变量，局部变量都统一使用lowerCamelCase风格，必须遵守驼峰形式

强制常量命名全部大写，单词间用下划线隔开

强制包名统一使用小写，点分隔符之前有且仅有一个英语单词

强制代码中命名严禁使用拼音与英文混合，更加不能直接使用中文的方式

强制代码中命名均不能以下划线或美元符号开始以及结束

## 常量命名

强制不允许任何魔法值直接出现在代码中(即未定义的常量)

## 路径规范

路径名称为：个人标识/表名称/功能名称；（例如：xmd/baseinfo/list）

权限名称同路径名称（功能必须权限限制）：个人标识/表名称/功能名称；（例如：xmd:baseinfo:list）

## git提交规范

提交内容填写：姓名：张三；内容：基础信息功能开发；类型：个人文件/公共文件-xxxxx表相关；

# 2、数据库规范

## 建表规范

强制表名，字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个下划线出现数字，数据库字段名的修改代价大，因为无法进行预发布

强制表名不使用复数次

强制禁用保留字

表命名最好是：标识\_业务名称 或 标识\_业务名称\_表作用 （例如：yh\_baseinfo、yh\_dict\_data/yh\_dict\_type）

库名与应用名称尽量保存一直多个单次用下划线分隔

## 公共字段

id 自增主键

create\_by 创建者

create\_time 创建时间

update\_by 更新者

update\_time 更新时间

remark 备注

user\_id 用户ID

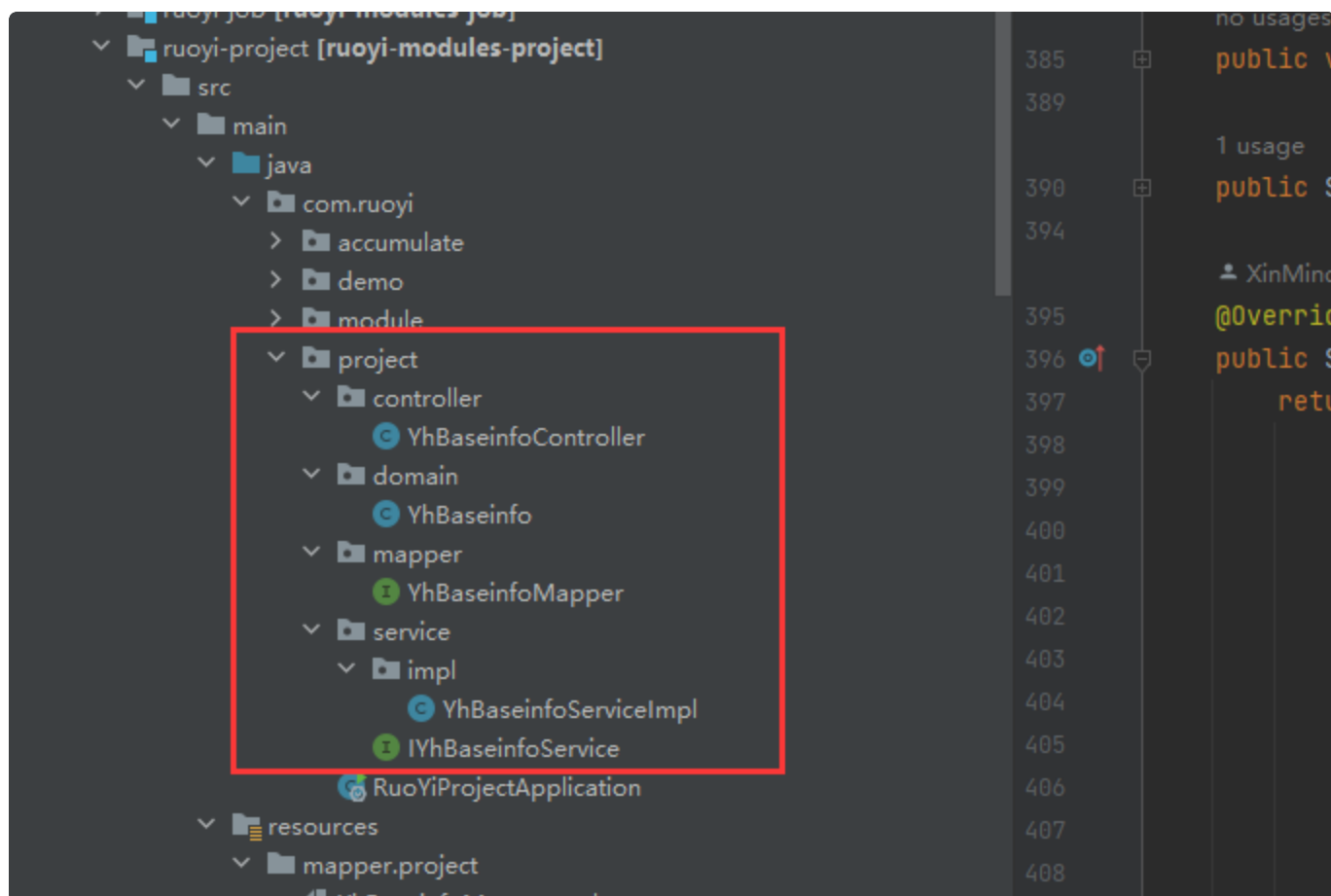
dept\_id 部门ID

## 三、流程

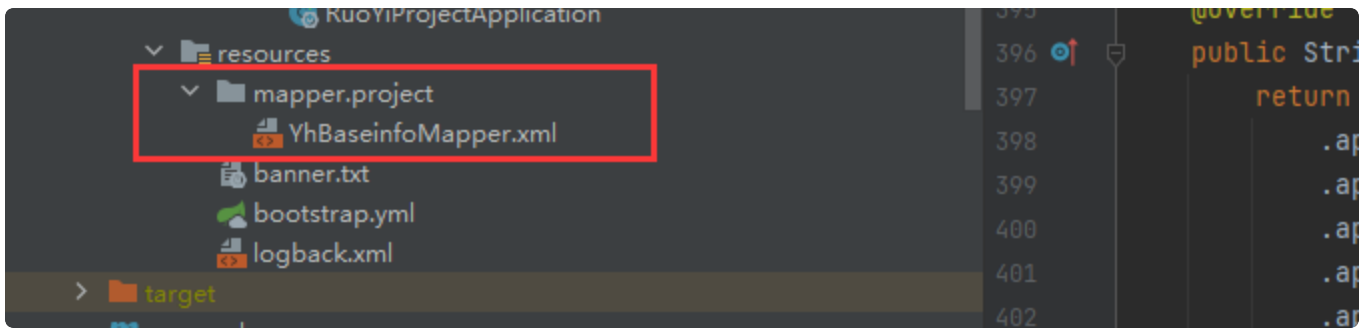
### 1、代码生成

生成文件包括：

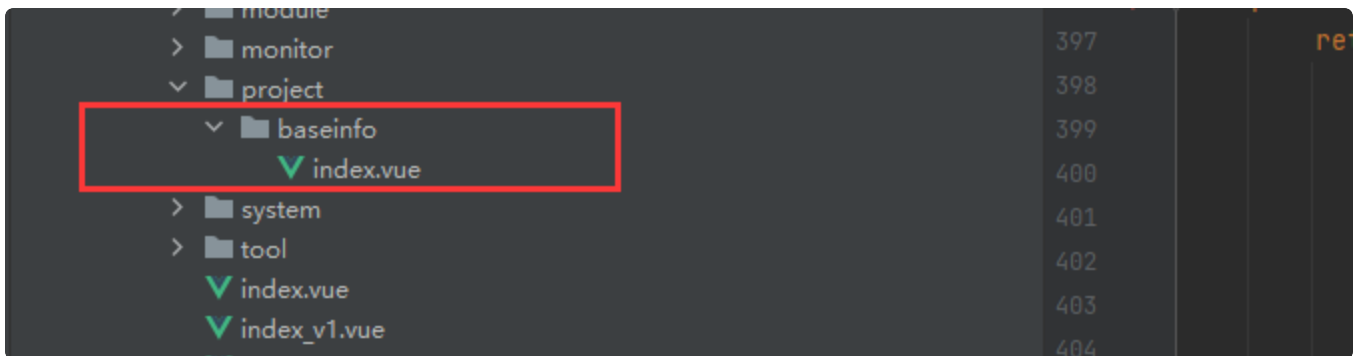
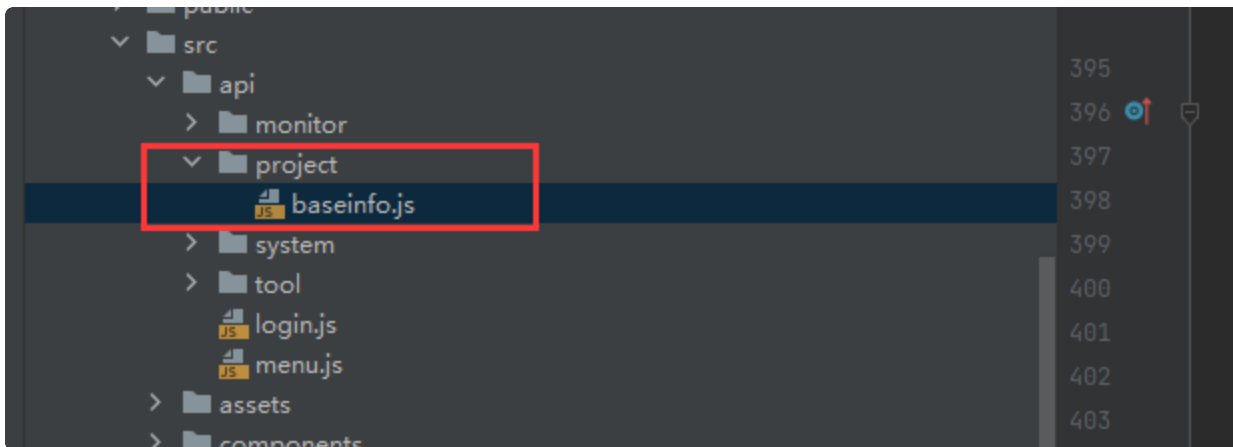
java： controller/domain/mapper/service(impl+service)



xml： mapper



前台：js/vue



## 2、功能开发

### 后台

XxxxController 文件可复制到个人的业务开发目录中，进行修改，业务逻辑写在 controller 具体方法中  
如遇业务开发新方法，需修改生成文件中的：xml/mapper/impl/service，以及 controller

### 前台

index.vue 文件可复制到个人的业务开发目录中，进行修改，样式可以参考前台模板

如遇业务开发新方法，修改生成文件中的：js 文件，以及 vue 文件

### 3、功能更新

如遇数据库字段变更情况，需手动修改：xml、domain 文件，可直接替换 index.vue，其他文件不可变更