**后端学习指导手册**

# 运行环境搭建

* Java JDK11环境搭建
* Java开发工具安装（比如：eclipse、idea等）
* Maven安装
* TortoiseGit安装（安装完毕后需要对eclipse或idea进行git相关配置）
* Mysql安装
* Mysql可视化工具安装及基础用法（推荐navicat premium）
* Postman安装

# Jfinaldemo项目运行

## 申请Gitlab账号

向Gitlab管理员申请账号（提供姓名、邮箱地址）

## 拉取项目

项目地址：<http://112.74.190.245:28080//training/backend.git>。

## 从master主分支创建新的分支

分支以本人姓名拼音小写命名。

*示例：张三，则分支名称为zhangsan*

## 运行项目

执行/src/main/java/com/shulv/study/jfinaldemo/Application.java中的main方法。

出现以下信息即为成功：

文本

描述已自动生成

## 访问接口

通过postman发送请求<http://127.0.0.1:8888/hello/>，返回如下信息即为成功：

图形用户界面, 文本, 应用程序

描述已自动生成

*通过postman发送请求<http://127.0.0.1:8888/hello/ex>，可查看抛出异常时返回信息。*

# Jfinaldemo连接数据库

## 创建数据库

在MySQL中创建数据库“jfinaldemo”，将项目中的jfinaldemo.sql文件导入数据库。

## 开启连接数据库配置

开放com.shulv.study.jfinaldemo.config.Bootstrap.java中configPlugin方法的注释：

文本, 信件

描述已自动生成

## 修改数据库连接信息

修改项目中/src/main/resources/config.dev.properties文件。

1. database.jdbcUrl：改为本地数据库地址
2. database.userName：改为本地数据库登录账号
3. database.password：改为本地数据库登录账号密码

## 运行项目

执行/src/main/java/com/shulv/study/jfinaldemo/Application.java中的main方法。

出现以下信息即为成功（红色字体信息表示建立数据库连接成功）：

文本

描述已自动生成

## 访问接口

通过postman发送请求http://127.0.0.1:8888/user/create，传入以下参数：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

返回如下信息即为成功：

图形用户界面, 文本, 应用程序

描述已自动生成

*通过postman发送请求<http://127.0.0.1:8888/user/find>，传入参数id（用户唯一标识），返回创建的用户信息。*

# 初步理解jfinaldemo项目

结合jfinal官方文档（https://jfinal.com/doc）理解项目，主要包含以下内容：

1. 熟悉项目开发基本规范
2. 熟悉项目基本启动配置
3. 掌握项目如何与数据库建立连接
4. 掌握数据库表如何与项目中的model类建立映射关系
5. 掌握接口如何暴露给前端，访问路由如何配置
6. 掌握如何接收前端数据
7. 掌握如何将数据返回给前端
8. 掌握基本的增删改查操作
9. 掌握如何抛出异常信息
10. 掌握统一返回数据格式的实现原理

# 熟悉Gitlab常用操作

* 拉取代码
* 更新代码
* 提交代码
* 发起代码合并请求
* 解决代码冲突

# 实践一

基于jfinaldemo项目本人的分支开发一个简单的用户管理模块，包含组织管理模块、用户管理模块、角色管理模块（不需要配置权限）。

## 组织管理模块

* 功能点描述

1. 新增组织
2. 更新组织
3. 删除组织
4. 分页查询组织信息
5. 查找指定ID组织信息

## 用户管理模块

* 功能点描述

1. 新增用户
2. 更新用户
3. 支持为用户配置组织、角色
4. 删除用户
5. 分页查询用户信息
6. 查找指定ID用户信息

## 角色管理模块

* 功能点描述

1. 新增角色
2. 更新角色
3. 删除角色，限制如果角色已分配给用户则抛出异常提醒“XXX”角色已分给XXX、XXX用户禁止删除
4. 分页查询角色信息
5. 查找指定ID角色信息

# 实践二

基于实践一开发用户认证模块。

## 添加用户登录名、密码

* 实现思路

1. 修改数据库及相关model，增加登录名、密码属性
2. 调整相关用户信息接口，增加相关的登录名、密码的持久化操作（注意：密码需要进行加密存储，如果能做加盐处理并实现更好）

## 颁发授信token票据

Token票据指的是通过用户认证后获取到的一串字符串，访问后端需要用户认证的接口时，需要携带此参数（推荐放在header中传输），并可通过token票据可解析出当前访问的用户身份信息。

* 实现思路

1. 数据库中增加token记录表，主要包含token标识（通过编码或加密处理后才能颁发给用户）、用户标识（对应用户身份信息）、颁发时间、有效期等
2. 增加颁发接口（又名：登录接口），所需参数为用户名、密码（如果能实现验证码登录更好），用户身份信息校验通过的则返回token，否则抛出异常信息登录失败（信息需指明失败原因）
3. token标识需按照指定规则（自行决定）生成的随机字符串（长度推荐32位），将其持久化到数据库中
4. 在返回时需要进行一次编码或加密处理，即存储在数据库的token标识和返回的token票据不同，但token票据是根据数据库中存储的标识按照一定编码或加密规则生成的

## 接口认证拦截

除颁发token票据接口，其余接口均增加用户认证拦截，仅允许携带有效token的用户可调用。

请参考jfinal的拦截器使用方法，要求通过在路由或接口上添加token校验拦截器实现。

# 实践三

基于实践二开发权限管理模块、数据统计模块。

## 权限管理模块

主要用于限制用户对接口的访问范围，权限信息主要包含名称、类型、编码等。

* 功能点描述

1. 新增权限，支持为权限定义类型，视图（VIEW）、接口（INTERFACE）
2. 权限分页/不分页查询
3. 支持为角色配置权限
4. 删除权限，限制接口权限和已分配至角色的视图权限禁止删除
5. 依据用户具有的权限限制可调用接口范围，实现思路：

* 通过用户认证拦截器将用户信息（主要是当前访问用户的权限）缓存在内存中（缓存方法推荐Threadlocal，请自行学习）
* 在需要权限才能调用的接口处（除登录和查询接口）添加相应接口所需权限限制，验证当前访问用户是否拥有其调用权（如果能够使用注解实现更好）
* 权限认证通过的才能够正常访问接口，否则抛出异常信息（指明用户没有哪种权限）

## 数据统计模块

简单统计各个组织（至少三个）中包含的用户有哪些，每个用户拥有多少个角色、多少个权限，返回数据（postman收到的data信息）以组织名称、用户姓名、角色数量、权限数量为一条记录。（组织名称会重复，属正常）

前提：需要了解MySQL存储过程如何编写以及jfinal如何调用存储过程。