

项目文档

项目基本信息

项目名称: ctd (Connect to Database)

项目简介: 本项目使用 前后端分离B/S架构 实现简单的用户增删改查

技术栈:

- 后端: Java + Spring + Springboot + Mybatis + Mybatis-Plus
- 前端: Html+CSS+Vue
- 数据库: Mysql

ctd (项目根目录)

```
|- .idea (IDEA项目配置文件目录)
|- .mvn (Maven相关配置目录)
|- src (源代码主目录)
  |- main (主代码目录)
    |- java (Java源代码目录)
      |- com.michael.ctd (项目基础包)
        |- controller (控制器层: 处理HTTP请求, 返回响应)
          |- UsersController.java (用户模块的接口层, 定义接口路径、接收参数)
        |- dto (数据传输对象层: 封装接口入参 出参的结构, 解耦前后端数据)
          |- UsersDto.java (用户模块的DTO, 用于接口参数的接收 返回)
        |- entity (实体层: 映射数据库表的Java对象, 对应数据库字段)
          |- Users.java (用户模块的实体类, 与数据库users表一一对应)
        |- mapper (数据访问层: 定义数据库操作接口, 继承MP的BaseMapper)
          |- UsersMapper.java (用户模块的Mapper接口, 提供增删改查等数据库操作)
        |- response (响应封装层: 统一接口返回格式, 规范响应结构)
          |- UserResponse.java (用户模块的响应类, 封装返回数据、状态码等)
        |- service (业务逻辑层: 处理核心业务逻辑)
          |- UserService.java (用户模块的业务类, 调用Mapper实现业务逻辑)
        |- CtdApplication.java (Spring Boot启动类: 项目入口, 启动Spring容器)
  |- resources (资源文件目录)
  |- static (静态资源目录: 存放CSS·JS、图片等静态文件)
  |- application.yml (项目配置文件: 配置数据库、端口、日志等参数)
```

技术文档

基本信息

- 后端运行地址: localhost:8080 → Backend
- 前端运行地址: localhost:8888 → Frontend
- 数据库运行地址: localhost:3306 → SQL

运行Jar包

- Windows/linux:
 - 确保已经安装Java环境, Java21
 - 确保CMD/Shell中Java --v可以输出版本信息
 - 确保安装了Mysql, Jar包数据库信息为 (使用IDEA打开项目可自行配置)
 - 账号: root
 - 密码: Password
 - 在Jar包目录打开CMD/Shell

- 输入 `java -jar ctd-0.0.1-SNAPSHOT.jar`
- 输出停住无报错即运行成功

二开项目

- 确保已经安装Java环境，Java21
- 确保CMD/Shell中 `Java --v` 可以输出版本信息
- 使用IDEA打开项目根目录
- 让IDEA Maven自动根据Pom下载依赖
- 报错Timeout请换阿里源
- 无报错可以直接运行
- `/resources/application.yml` 为Springboot配置文件，此处配置数据库账号密码

数据库信息

- 数据库名: users
- 表名: users
- 主键: id (自增)
- 演示数据仅包含一条数据，Userid为1
- 字段:
 - "age": 年龄
 - "hobbies": 爱好
 - "school": 学校
 - "sex": 性别
 - "username": 用户名
- 初始化数据库:

```

/*创建数据库*/
CREATE DATABASE `users` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
/*! !80016 DEFAULT ENCRYPTION='N' */;
-- users.users definition

/*创建表*/
CREATE TABLE `users` (
  `id` int unsigned NOT NULL AUTO_INCREMENT,
  `username` text NOT NULL,
  `age` int unsigned NOT NULL,
  `school` text NOT NULL,
  `hobbies` text NOT NULL,
  `sex` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*向表插入数据*/
INSERT INTO users.users (username,age,school,hobbies,sex) VALUES
  ('Michael',17,'Shanwei Institute','Table Tennis','Male');

```

接口调用

返回值:

"status": 状态

"user_info": 用户信息

"age": 年龄
"hobbies": 爱好
"school": 学校
"sex": 性别
"username": 用户名
"message": 返回信息
"before": 修改前用户信息
"after": 修改后用户信息

删除用户：

- 请求地址: Backend/users/ <UserId>
- 请求方法: Delete
- 响应体:

```
{  
  
  "status": "success",  
  "message": "Deleted User , Username = xiaohong"  
  
}
```

查询用户：

- 请求地址: Backend/users/ <UserId>
- 请求方法: Get
- 响应体:

```
{  
  
  "status": "success",  
  
  "user_info": {  
    "age": 22,  
    "hobbies": "eat foods",  
    "school": "Beijing University",  
    "sex": "Female",  
    "username": "xiaohong"}  
  
}
```

新增用户：

- 请求地址: Backend/users
- 请求方法: Post
- 请求体:

```
{  
  
  "username": "xiaohong",  
  "age": 22,
```

```
"sex": "Female",
"school": "Beijing University",
"hobbies": "eat foods"

}
```

- 响应体:

```
{
  "message": "Created User , id = 11",
  "status": "success"
}
```

修改用户:

- 请求地址: Backend/users/ <UserId>
- 请求方法: Put
- 请求体:

```
{
  "username": "xiaohong",
  "age": 22,
  "sex": "Female",
  "school": "Beijing University",
  "hobbies": "eat foods"
}
```

- 响应体:

```
{
  "after": {
    "age": 17,
    "hobbies": "Table Tennis",
    "school": "Shanwei Institute",
    "sex": "Male",
    "username": "Michael"},

  "before": {
    "age": 22,
    "hobbies": "eat foods",
    "school": "Beijing University",
    "sex": "Female",
    "username": "xiaohong"},

  "id": 11,
  "status": "success"
}
```

