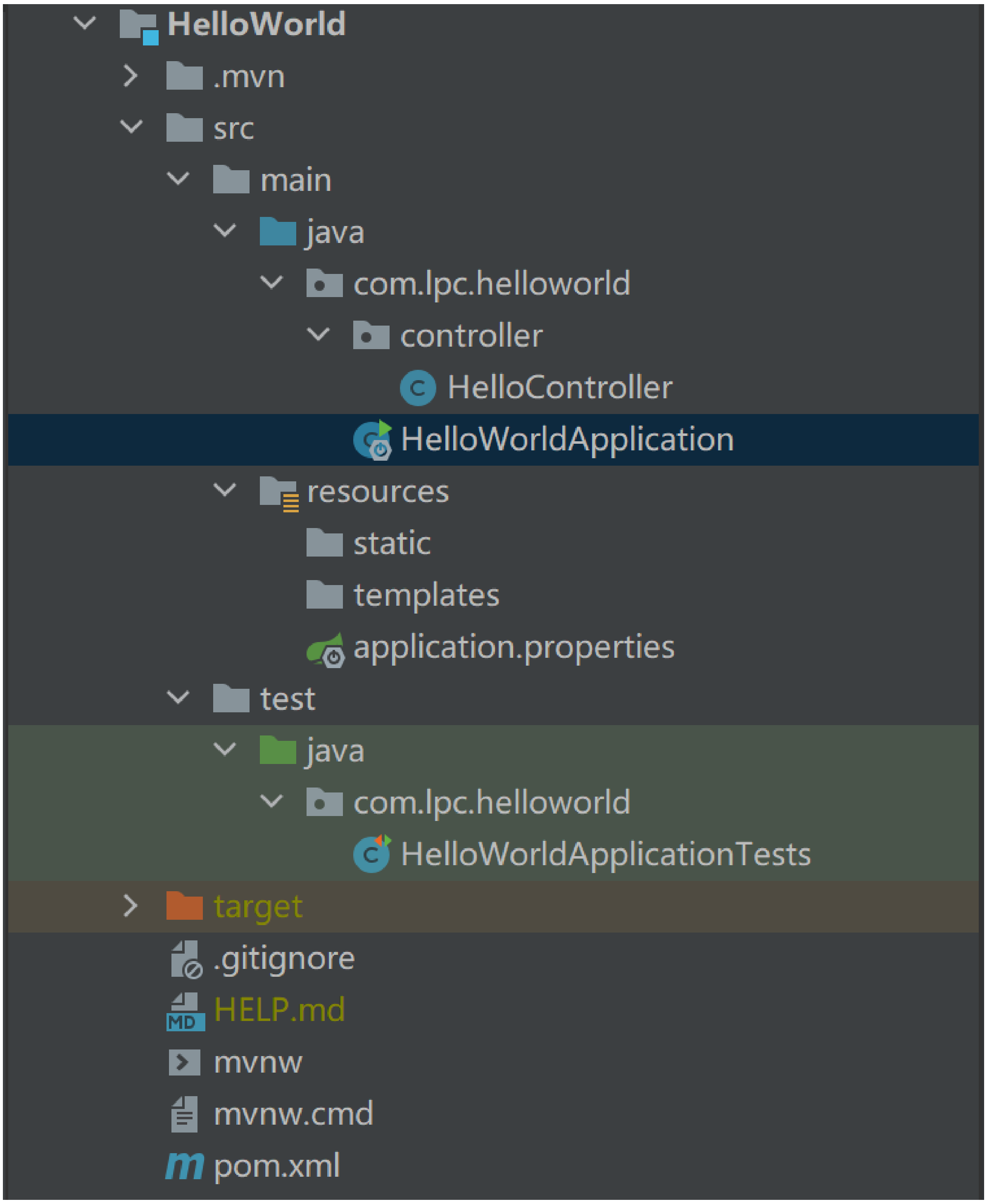
一、Hello World

1、目录结构



```
HelloWorld
    .mvn
    src
        main
            java
                com.lpc.helloworld
                    controller
                        HelloController
                    HelloWorldApplication
            resources
                static
                templates
                application.properties
        test
            java
                com.lpc.helloworld
                    HelloWorldApplicationTests
    target
    .gitignore
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

2、代码

- HelloController代码

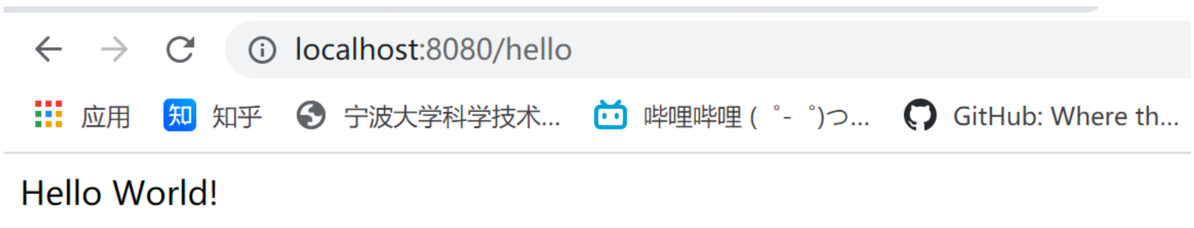
```java
package com.lpc.helloworld.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * @author byu_rself
 * @create 2021/9/7 21:58
 */

@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello(){
        return "Hello World!";
    }

}
```

- HelloWorldApplication代码

```java
package com.lpc.helloworld;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class HelloworldApplication {

    public static void main(String[] args) {
        SpringApplication.run(HelloworldApplication.class, args);
    }

}
```

3、运行效果

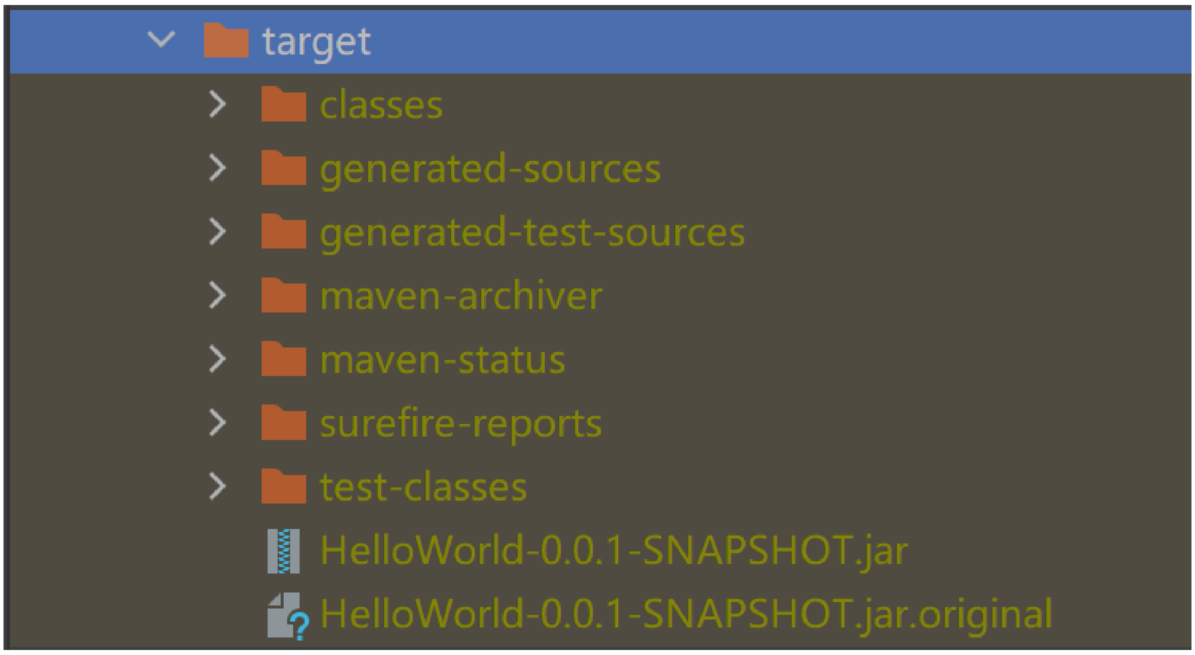
>浏览器内运行



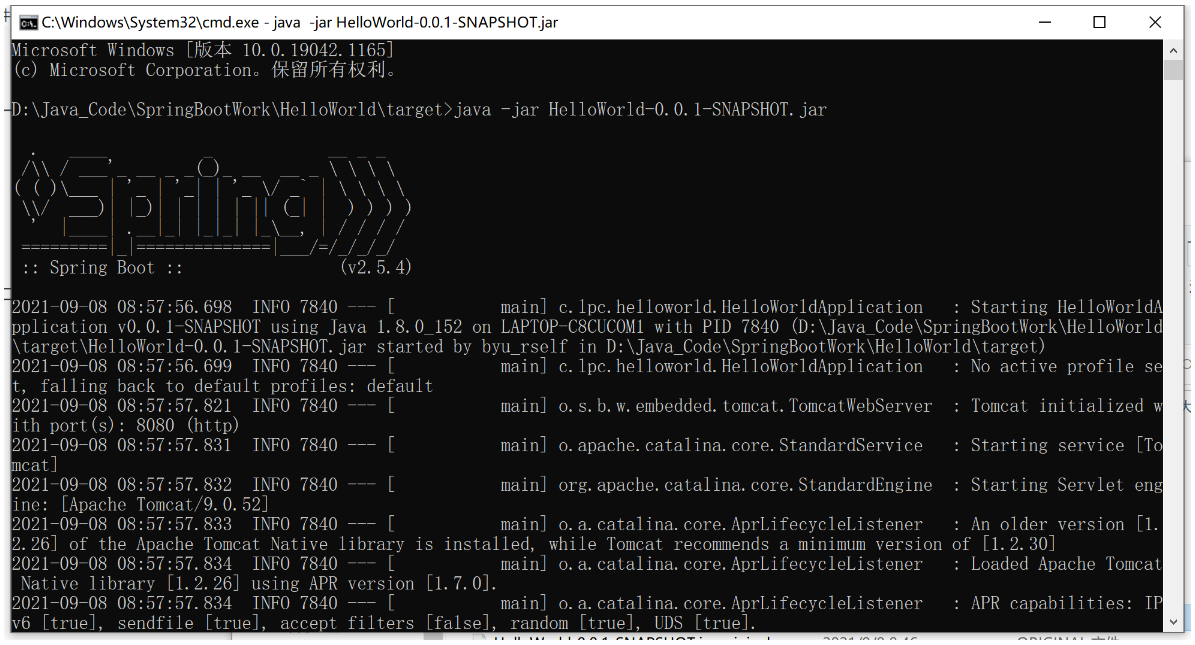
Hello World!

>cmd命令行中运行

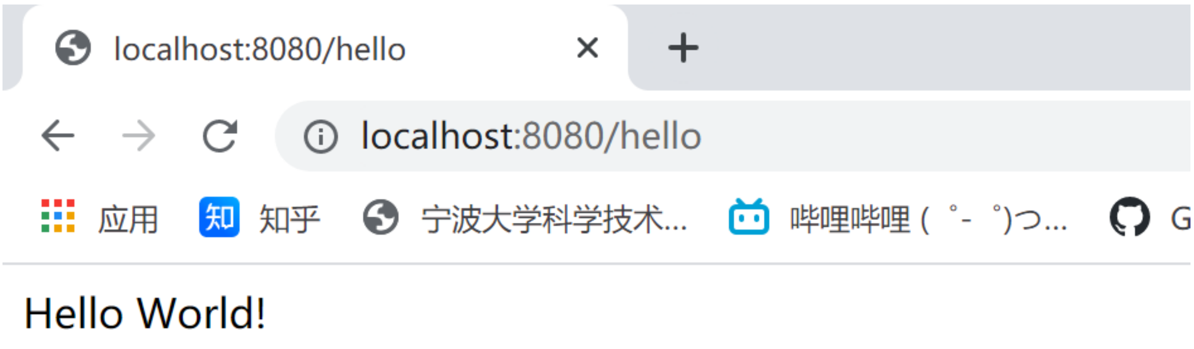
对代码打包


```
target
  classes
  generated-sources
  generated-test-sources
  maven-archiver
  maven-status
  surefire-reports
  test-classes
  HelloWorld-0.0.1-SNAPSHOT.jar
  HelloWorld-0.0.1-SNAPSHOT.jar.original
```

在cmd中输入指令


C:\Windows\System32\cmd.exe - java -jar HelloWorld-0.0.1-SNAPSHOT.jar

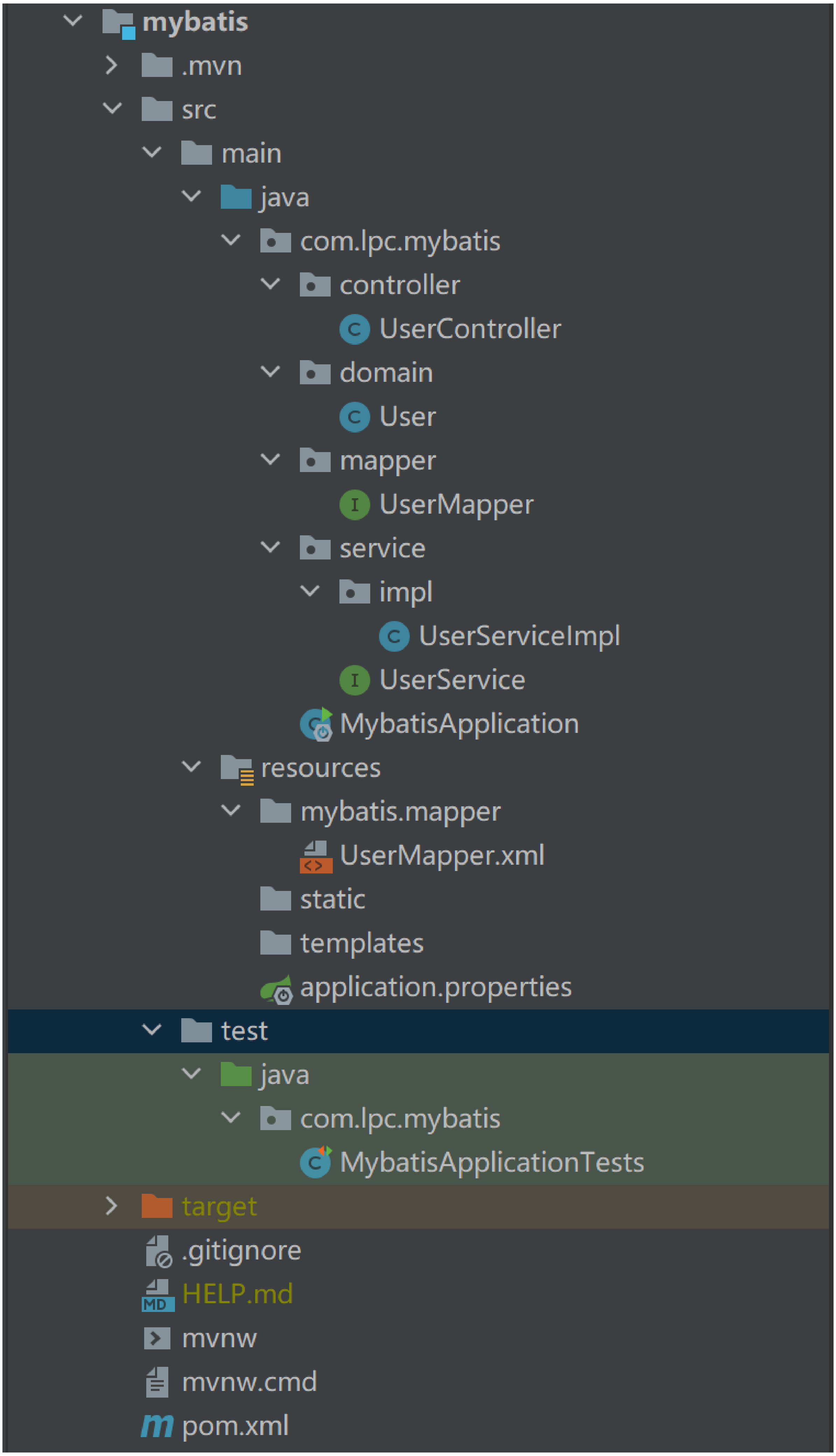
```
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

D:\Java_Code\SpringBootWork\HelloWorld\target>java -jar HelloWorld-0.0.1-SNAPSHOT.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.4)

2021-09-08 08:57:56.698  INFO 7840 --- [           main] c.lpc.helloworld.HelloWorldApplication   : Starting HelloWorldA
pplication v0.0.1-SNAPSHOT using Java 1.8.0_152 on LAPTOP-C8CUCOM1 with PID 7840 (D:\Java_Code\SpringBootWork\HelloWorld
\target\HelloWorld-0.0.1-SNAPSHOT.jar started by byu_rself in D:\Java_Code\SpringBootWork\HelloWorld\target)
2021-09-08 08:57:56.699  INFO 7840 --- [           main] c.lpc.helloworld.HelloWorldApplication   : No active profile se
t, falling back to default profiles: default
2021-09-08 08:57:57.821  INFO 7840 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized w
ith port(s): 8080 (http)
2021-09-08 08:57:57.831  INFO 7840 --- [           main] o.apache.catalina.core.StandardService   : Starting service [To
mcat]
2021-09-08 08:57:57.832  INFO 7840 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet eng
ine: [Apache Tomcat/9.0.52]
2021-09-08 08:57:57.833  INFO 7840 --- [           main] o.a.catalina.core.AprLifecycleListener   : An older version [1.
2.26] of the Apache Tomcat Native library is installed, while Tomcat recommends a minimum version of [1.2.30]
2021-09-08 08:57:57.834  INFO 7840 --- [           main] o.a.catalina.core.AprLifecycleListener   : Loaded Apache Tomcat
 Native library [1.2.26] using APR version [1.7.0].
2021-09-08 08:57:57.834  INFO 7840 --- [           main] o.a.catalina.core.AprLifecycleListener   : APR capabilities: IP
v6 [true], sendfile [true], accept filters [false], random [true], UDS [true].
```



Hello World!

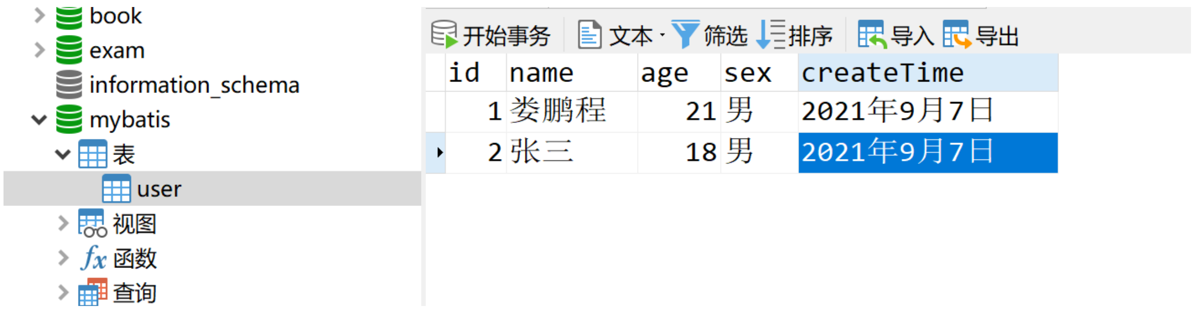
二、Mybatis

1、目录结构



```
mybatis
  .mvn
  src
    main
      java
        com.lpc.mybatis
          controller
            UserController
          domain
            User
          mapper
            UserMapper
          service
            impl
              UserServiceImpl
            UserService
          MybatisApplication
      resources
        mybatis.mapper
          UserMapper.xml
        static
        templates
        application.properties
    test
      java
        com.lpc.mybatis
          MybatisApplicationTests
  target
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

2、初始数据库



id	name	age	sex	createTime
1	娄鹏程	21	男	2021年9月7日
2	张三	18	男	2021年9月7日

3、代码

- User代码

```java
package com.lpc.mybatis.domain;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * @author byu_rself
 * @create 2021/9/7 20:46
 */

@Data
@NoArgsConstructor
@AllArgsConstructor
public class User {

    private Integer id;
    private String name;
    private Integer age;
    private String sex;
    private String createTime;

}
```

- application.properties代码

```properties
spring.datasource.username=root
spring.datasource.password=123456
spring.datasource.url=jdbc:mysql://localhost:3306/mybatis?
serverTimezone=UTC&useUnicode=true&characterEncoding=utf-8
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# 整合mybatis
mybatis.type-aliases-package=com.lpc.mybatis.domain
mybatis.mapper-locations=classpath:mybatis/mapper/*.xml
```

- 在test包中测试mybatis是否配置成功

```java
package com.lpc.mybatis;

```

```java
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import javax.sql.DataSource;
import java.sql.SQLException;

@SpringBootTest
class MybatisApplicationTests {

    @Autowired
    DataSource dataSource;

    @Test
    void contextLoads() throws SQLException {
        System.out.println(dataSource.getClass());
        System.out.println(dataSource.getConnection());
    }

}
```

- UserMapper代码

```java
package com.lpc.mybatis.mapper;

import com.lpc.mybatis.domain.User;
import org.apache.ibatis.annotations.Mapper;
import org.springframework.stereotype.Repository;

import java.util.List;

/**
 * @author byu_rself
 * @create 2021/9/7 20:50
 */

@Mapper
@Repository
public interface UserMapper {

    List<User> selectAllUser();

    User queryUserById(Integer id);

    int addUser(User user);

    int updateUserById(User user);

    int deleteUserById(int id);

}
```

- UserMapper.xml代码

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
```

```
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.lpc.mybatis.mapper.UserMapper">

    <select id="selectAllUser" resultType="User">
        select * from user
    </select>

    <select id="queryUserById" resultType="User">
        select * from user where id = #{id}
    </select>

    <insert id="addUser" parameterType="User">
        insert into user (id,name,age,sex,createTime) values (#{id},#
{name},#{age},#{sex},#{createTime})
    </insert>

    <update id="updateUserById" parameterType="User">
        update user set name = #{name},age = #{age},sex=#{sex},createTime=#
{createTime} where id = #{id}
    </update>

    <delete id="deleteUserById" parameterType="Integer">
        delete from user where id = #{id}
    </delete>

</mapper>
```

- UserService代码

```java
package com.lpc.mybatis.service;

import com.lpc.mybatis.domain.User;

import java.util.List;

/**
 * @author byu_rself
 * @create 2021/9/7 21:44
 */

public interface UserService {

    List<User> selectAllUser();

    User queryUserById(Integer id);

    int addUser(User user);

    int updateUserById(User user);

    int deleteUserById(int id);

}
```

- UserServiceImpl代码

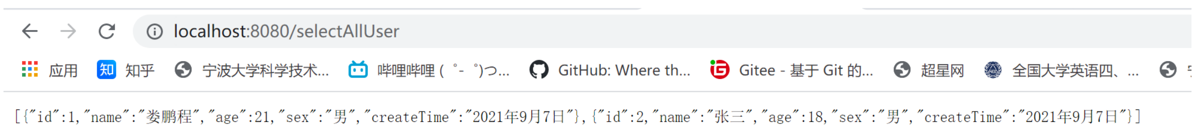
```java
package com.lpc.mybatis.service.impl;

import com.lpc.mybatis.domain.User;
import com.lpc.mybatis.mapper.UserMapper;
import com.lpc.mybatis.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

/**
 * @author byu_rself
 * @create 2021/9/7 21:46
 */
@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserMapper UserMapper;

    @Override
    public List<User> selectAllUser() {
        List<User> userList = UserMapper.selectAllUser();
        return userList;
    }

    @Override
    public User queryUserById(Integer id) {
        return UserMapper.queryUserById(id);
    }

    @Override
    public int addUser(User user) {
        return UserMapper.addUser(user);
    }

    @Override
    public int updateUserById(User user) {
        return UserMapper.updateUserById(user);
    }

    @Override
    public int deleteUserById(int id) {
        return UserMapper.deleteUserById(id);
    }
}
```

- UserController代码

```java
package com.lpc.mybatis.controller;

import com.lpc.mybatis.domain.User;
import com.lpc.mybatis.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import java.util.List;

/**
 * @author byu_rself
 * @create 2021/9/7 21:09
 */

@RestController
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping("/selectAllUser")
    public List<User> selectAllUser(){
        List<User> userList = userService.selectAllUser();
        for (User user : userList) {
            System.out.println(user);
        }
        return userList;
    }

    @GetMapping("/queryUserById")
    public User queryUserById(){
        User user = userService.queryUserById(1);
        return user;
    }

    @GetMapping("/addUser")
    public String addUser(){
        userService.addUser(new User(null,"张三",18,"男","2021年9月7日"));
        return "添加成功";
    }

    @GetMapping("/updateUser")
    public String updateUser(){
        userService.updateUserById(new User(1,"Lpc",18,"男","2021年9月9日"));
        return "修改成功";
    }

    @GetMapping("/deleteUserById")
    public String deleteUserById(){
        userService.deleteUserById(2);
        return "删除成功";
    }

}
```

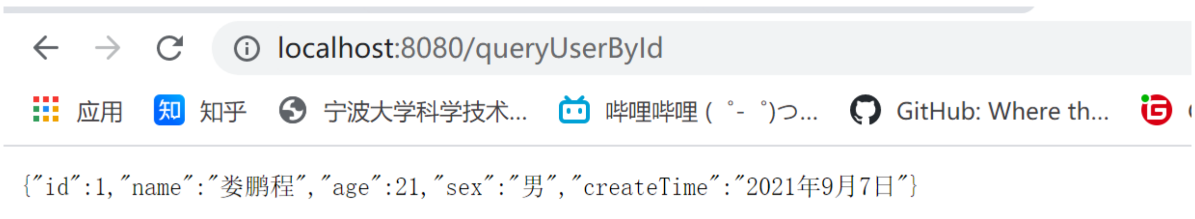
4、运行效果

>查询所有用户



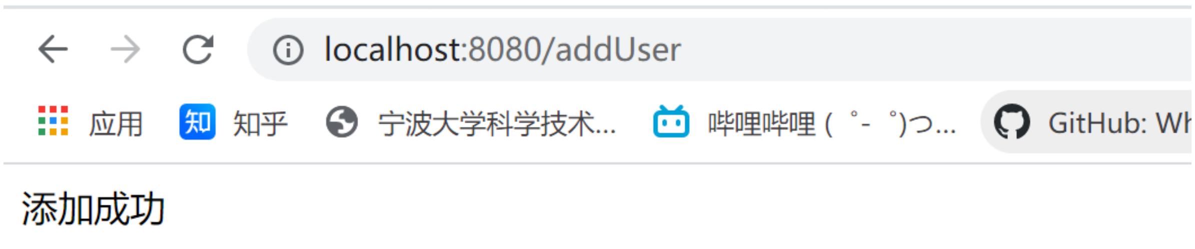
```
[{"id":1,"name":"娄鹏程","age":21,"sex":"男","createTime":"2021年9月7日"},{"id":2,"name":"张三","age":18,"sex":"男","createTime":"2021年9月7日"}]
```

>根据id查询用户（此时id=1）



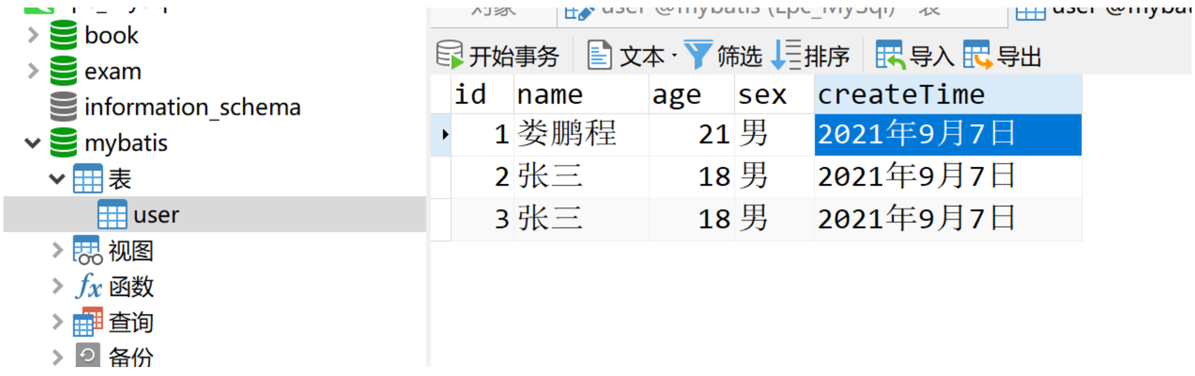
```
{"id":1,"name":"娄鹏程","age":21,"sex":"男","createTime":"2021年9月7日"}
```

>添加用户



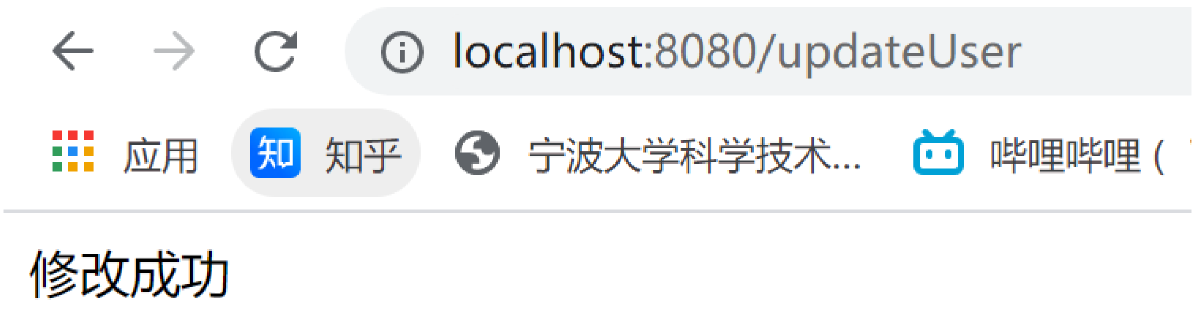
添加成功

此时数据库中新增一条记录



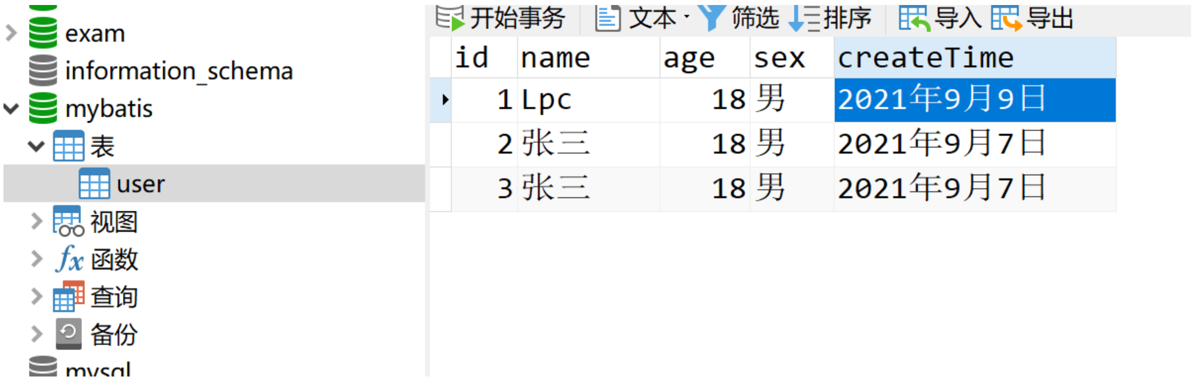
id	name	age	sex	createTime
1	娄鹏程	21	男	2021年9月7日
2	张三	18	男	2021年9月7日
3	张三	18	男	2021年9月7日

>修改用户



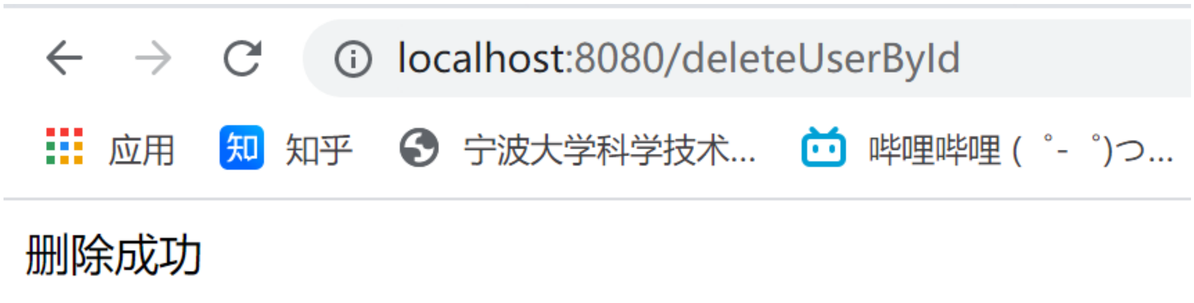
修改成功

此时数据库第一条记录被修改



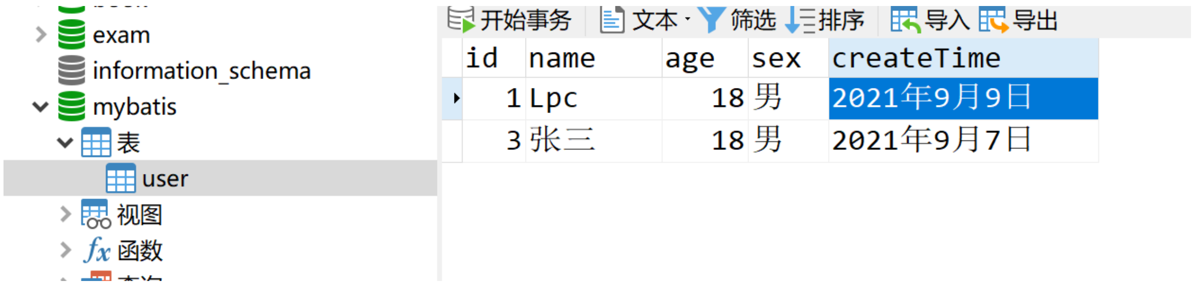
id	name	age	sex	createTime
1	Lpc	18	男	2021年9月9日
2	张三	18	男	2021年9月7日
3	张三	18	男	2021年9月7日

>删除用户



删除成功

此时数据库第二条记录被删除



id	name	age	sex	createTime
1	Lpc	18	男	2021年9月9日
3	张三	18	男	2021年9月7日