Api 接口说明

1、登录

url：ip:8080/login/signIn

参数：username,password

返回：0表示成功，1表示失败

注意：没有实现注册，用户信息需要在数据库里添加，并且密码需要加密

加密可以使用接口：

url：ip:8080/getPassword

参数：password

返回：加密后的密码

user/admin服务 (login required,role user or admin requried)

2、发布预约

url：ip:8080/user/add/reserve

参数：gymid,fieldid,date,starttime,endtime

返回：string表示发布状态

3、删除预约

url：ip:8080/user/delete/reserve

参数：reserveid

返回：string表示删除状态

4、查询用户所有预约

url：ip:8080/user/query/reserve

返回：json文件

5、查询所有通知

url：ip:8080/user/query/announce

返回：json文件

6、查询可用场地

url：ip:8080/user/query/field
参数：date,starttime,endtime
返回：json文件

7、查询用户信息
url：ip:8080/user/query/user
返回：json文件

teacher/admin服务 (login required,role teacher or admin requried)

8、添加课程
url：ip:8080/teacher/add/course
参数：coursename,weekday,starttime,endtime,gymid
返回：string表示发布状态

9、删除课程
url：ip:8080/teacher/delete/course
参数：courseid
返回：string表示删除状态

10、查询自己发布的所有课程
url：ip:8080/teacher/query/course
返回：json文件

11、查询可用场馆
url：ip:8080/teacher/query/gym
参数：weekday,starttime,endtime
返回：json文件

student/admin服务 (login required,role student or admin requried)

12、查询所有课程
url：ip:8080/student/query/allcourse
返回：json文件

13、查询我的课程
url：ip:8080/student/query/mycourse
返回：json文件

14、加入课程：
url：ip:8080/student/add/course
参数：courseid
返回：string表示加入状态
注意：不要重复加入同一个课程

15、删除课程
url：ip:8080/student/delete/course
参数：courseid
返回：string表示删除状态

admin服务 (login required,role admin requried)

16、添加通知
url：ip:8080/admin/add/announce
参数：time,content
返回：string表示添加状态

java代码注意

在**application.properties**中：
spring.datasource.url=jdbc:mysql://localhost:3306/project?useUnicode=true&characterEncoding=utf8中project为数据库名
需要执行一下mysql语句：

```
create databse project
use project
```

spring.datasource.password中需要填写数据库密码

数据库定义

```sql
create table role
(
    role_id int(11) auto_increment,
    name char(20),
    primary key(role_id)
);

create table user
(
    user_id int(11) auto_increment,
    username char(50),
    password char(100),
    name char(50),
    depr char(30),
    role_id int(11),
    primary key (user_id)
);

alter table users modify name char(50) character set gbk;
alter table user add constraint fk0 foreign key(role_id) references role(role_id);

create table gym
(
    gym_id int(11) auto_increment,
    name char(50),
    starttime char(15),
    endtime char(15),
    primary key(gym_id)
);

alter table gym modify name char(50) character set gbk;

create table field
(
    gym_id int(11),
    field_id int(11),
    name char(50),
    primary key (gym_id,field_id)
);

alter table field modify name char(50) character set gbk;
alter table field add constraint fk1 foreign key(gym_id) references gym(gym_id);

create table reserve
(
    reserve_id int(11) auto_increment,
    user_id int(11),
    gym_id int(11),
    field_id int(11),
    date char(20),
    starttime char(15),
    endtime char(15),
    primary key(reserve_id)
);

alter table reserve add constraint fk2 foreign key(user_id) references user(user_id);
alter table reserve add constraint fk3 foreign key(gym_id,field_id) references field(gym_id,field_id);

create table course
(
    course_id int(11) auto_increment,
    weekday int(11),
    name char(50),
    srarttime char(15),
    endtime char(15),
    gym_id int(11),
    primary key(course_id)
);

alter table course modify name char(50) character set gbk;
alter table course add constraint fk4 foreign key(gym_id) references gym(gym_id);

create table take
(
    user_id int(11),
    course_id int(11),
    primary key(user_id,course_id)
);
```

```sql

alter table take add constraint fk5 foreign key(user_id) references user(user_id);
alter table take add constraint fk6 foreign key(course_id) references course(course_id);

create table teach
(
    user_id int(11),
    course_id int(11),
    primary key(user_id,course_id)
);

alter table teach add constraint fk7 foreign key(user_id) references user(user_id);
alter table teach add constraint fk8 foreign key(course_id) references course(course_id);

CREATE TABLE `SPRING_SESSION` (
  `SESSION_ID` char(36) NOT NULL DEFAULT '',
  `CREATION_TIME` bigint(20) NOT NULL,
  `LAST_ACCESS_TIME` bigint(20) NOT NULL,
  `MAX_INACTIVE_INTERVAL` int(11) NOT NULL,
  `PRINCIPAL_NAME` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`SESSION_ID`) USING BTREE,
  KEY `SPRING_SESSION_IX1` (`LAST_ACCESS_TIME`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `SPRING_SESSION_ATTRIBUTES` (
  `SESSION_ID` char(36) NOT NULL DEFAULT '',
  `ATTRIBUTE_NAME` varchar(100) NOT NULL DEFAULT '',
  `ATTRIBUTE_BYTES` blob,
  PRIMARY KEY (`SESSION_ID`,`ATTRIBUTE_NAME`),
  KEY `SPRING_SESSION_ATTRIBUTES_IX1` (`SESSION_ID`) USING BTREE,
  CONSTRAINT `SPRING_SESSION_ATTRIBUTES_ibfk_1` FOREIGN KEY (`SESSION_ID`) REFERENCES `SPRING_SESSION` (`SESSION_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

insert into role(name) values ("ROLE_ADMIN"),("ROLE_STUDENT"),("ROLE_TEACHER"),("ROLE_TOURIST");

insert into gym(name,starttime,endtime) values ("中北大学生活动中心羽毛球馆","08:00","22:00"),("中北体育馆羽毛球馆","07:00","22:00");

insert into field(gym_id,field_id,name) values (1,1,"1号场地"),(1,2,"2号场地"),(1,3,"3号场地"),(1,4,"4号场地"),(1,5,"5号场地"),(1,6,"6号场地"),(1,7,"7号场地"),(1,8,"8号场地");

insert into field(gym_id,field_id,name) values (2,1,"1号场地"),(2,2,"2号场地"),(2,3,"3号场地"),(2,4,"4号场地"),(2,5,"5号场地"),(2,6,"6号场地"),(2,7,"7号场地"),(2,8,"8号场地"),(2,9,"9号场地"),(2,10,"10号场地"),(2,11,"11号场地"),(2,12,"12号场地");
```