连接的IP地址为

```
1 public static final String BASE_URL = "http://39.105.43.3:8080/";
```

apk下载链接

HaveFun

User相关

```
package com.hebtu.havefun.controller;
 3
    import com.alibaba.fastjson.JSON;
    import com.hebtu.havefun.service.UserService;
 4
    import org.springframework.web.bind.annotation.RequestMapping;
 6
    import org.springframework.web.bind.annotation.RestController;
 8
    import javax.annotation.Resource;
 9
    import java.util.List;
10
    /**
11
12
    * @author PengHuAnZhi
13
     * @createTime 2020/11/19 11:15
    * @projectName HaveFun
14
15
     * @className UserController.java
16
     * @description TODO
17
18
    @RestController
    @RequestMapping("user")
19
20
    public class UserController {
21
22
        @Resource
23
        UserService userService;
```

判断是否注册

```
1
 2
         * @param phoneNum 电话号码
         * @return 返回"true"或者"false"
 3
         * @description 判断是否注册
 5
         */
        @RequestMapping("/judgeRegistered")
 6
 7
        public String judgeRegistered(String phoneNum) {
8
            if (phoneNum == null) {
                System.out.println("judgeRegistered Error");
 9
10
                return "ErrorParameter";
11
            if (userService.judgeRegistered(phoneNum)) {
12
13
                return "false";
14
            } else {
15
                return "true";
```

```
16 | }
17 | }
```

注册

```
/**
 2
         * @param phoneNum 手机号码
 3
         * @param password 密码
 4
         * @return 返回"true"或者"false"
 5
         * @description 注册
 6
         */
 7
        @RequestMapping("/register")
 8
        public String register(String phoneNum, String password) {
 9
            if (phoneNum == null || password == null) {
                System.out.println("register Error");
10
                return "ErrorParameter";
11
12
            }
            if (userService.register(phoneNum, password)) {
13
                return "true";
14
15
            } else {
                return "false";
16
17
            }
        }
18
```

景景

```
1
 2
         * @param phoneNum 手机号
 3
         * @param password 密码
 4
         * @return 返回的是user的对象Json串
 5
         * @description 登录
 6
        @RequestMapping("/login")
 7
        public String login(String phoneNum, String password) {
 8
 9
            if (phoneNum == null || password == null) {
10
                System.out.println("login Error");
11
                return "ErrorParameter";
12
            }
            String user = userService.login(phoneNum, password);
13
14
            return "".equals(user) ? "" : user;
        }
15
```

修改密码

```
1
2
         * @param phoneNum 电话号码
3
         * @param password 密码
        * @return 返回"true"或"false"
4
5
         * @description 修改密码
6
         */
        @RequestMapping("/modifyPassword")
7
8
        public String modifyPassword(String phoneNum, String password) {
9
            if (phoneNum == null || password == null) {
                System.out.println("modifyPassword Error");
10
```

```
11
                 return "ErrorParameter";
12
            }
            if (userService.modifyPassword(phoneNum, password)) {
13
14
                 return "true";
15
            } else {
                 return "false";
16
17
            }
18
        }
```

报名活动

```
1
        * @param activityId 活动id
 2
 3
        * @param id 用户id,注意不是getUserId,是getId
        * @return 返回报名成功"true",如果超出活动人数上限,返回"enough",已经报名
 4
    了"exists"
 5
        * @description 报名活动
        */
 6
 7
       @RequestMapping("/enrollActivity")
8
        public String enrollActivity(Integer activityId, Integer id) {
 9
           if (activityId == null || id == null) {
10
               System.out.println("enrollActivity Error");
               return "ErrorParameter";
11
           }
12
13
           return userService.enrollActivity(activityId, id);
14
        }
```

取消报名活动

```
/**
1
 2
        * @param activityId 活动id
        * @param id 用户id,注意不是getUserId,是getId
 3
        * @return 返回取消报名成功"true"或者失败"false"
 4
        * @description 取消报名活动
 6
        */
 7
        @RequestMapping("/cancelEnrollActivity")
8
        public String cancelEnrollActivity(Integer activityId, Integer id) {
9
            if (activityId == null || id == null) {
               System.out.println("enrollActivity Error");
10
11
                return "ErrorParameter";
12
            }
            return "success".equals(userService.cancelEnrollActivity(activityId,
13
    id)) ? "true" : "false";
14
        }
```

收藏或者取消收藏

```
@RequestMapping("/changeCollectActivity")
 8
        public String changeCollectActivity(Integer activityId, Integer id,
    String collect) {
10
            if (activityId == null || id == null || collect == null) {
11
                System.out.println("changeCollectActivity Error");
                return "ErrorParameter";
12
13
14
            boolean tag = Boolean.parseBoolean(collect);
            boolean flag = userService.changeCollectActivity(activityId, id,
15
    tag);
            return flag ? "true" : "false";
16
17
        }
```

修改个性签名

```
1
        /**
 2
         * @param id
                                    用户id,注意不是getUserId,是getIds
 3
         * @param personalSignature 新的个性签名
         * @return 返回"true"或者"false"
 4
 5
         * @description 修改个性签名
         */
 6
 7
        @RequestMapping("/modifyPersonalSignature")
 8
        public String modifyPersonalSignature(Integer id, String
    personalSignature) {
9
            if (id == null || personalSignature == null) {
                System.out.println("modifyPersonalSignature Error");
10
11
                return "ErrorParameter";
12
            }
            return userService.modifyPersonalSignature(id, personalSignature) ?
13
    "true" : "false";
        }
14
```

获取消息列表, 分页显示

```
1
        /**
 2
         * @param sender   发送者id,注意不是getUserId,是getId
 3
         * @param receiver 接收者id,注意不是getUserId,是getId
 4
         * @param pageNum 当前页码
 5
         * @param pageSize 页面大小
 6
         * @return 返回List<Messages>集合Json,没有消息则为empty
 7
         * @description 获取消息列表,分页显示
 8
 9
        @RequestMapping("/getMsg")
10
        public String getMsg(Integer sender, Integer receiver, Integer pageNum,
    Integer pageSize) {
            if (sender == null || receiver == null || pageNum == null ||
11
    pageSize == null) {
                System.out.println("getMsg Error");
12
13
                return "ErrorParameter";
14
            }
15
            List<Messages> messagesList = userService.getMsg(sender, receiver,
    pageNum, pageSize);
16
            return messagesList != null ? JSON.toJSONString(messagesList) :
    "empty";
17
        }
```

判断是否关注

```
1
 2
         * @param followId
                            当前用户的id,即关注者id,注意不是getUserId,是getId
 3
         * @param followedId 对方id,即被关注者id,注意不是getUserId,是getId
 4
         * @return 返回是否关注 "true"或者"false"
 5
         * @description 判断是否关注
        */
 6
 7
        @RequestMapping("/judgeFollow")
 8
        public String judgeFollow(Integer followId, Integer followedId) {
9
            if (followId == null || followedId == null) {
                System.out.println("judgeFollow Error");
10
11
                return "ErrorParameter";
12
            }
13
            return userService.judgeFollow(followId, followedId) ? "true" :
    "false";
14
        }
```

关注用户

```
1
2
                            当前用户的id,即关注者id,注意不是getUserId,是getId
        * @param followId
3
        * @param followedId 对方id,即被关注者id,注意不是getUserId,是getId
        * @return 返回关注成功还是失败true或者false
4
5
        * @description 关注用户
6
       @RequestMapping("/followUser")
7
8
       public String followUser(Integer followId, Integer followedId) {
9
           if (followId == null || followedId == null) {
10
               System.out.println("followUser Error");
               return "ErrorParameter";
11
12
           }
13
           return userService.followUser(followId, followedId) ? "true" :
    "false":
       }
14
```

取消关注用户

```
1
        /**
2
                            当前用户的id,即关注者id,注意不是getUserId,是getId
        * @param followId
3
        * @param followedId 对方id,即被关注者id,注意不是getUserId,是getId
4
        * @return 返回取消关注成功还是失败true或者false
5
        * @description 取消关注用户
        */
6
7
       @RequestMapping("/unFollowUser")
       public String unFollowUser(Integer followId, Integer followedId) {
8
9
           if (followId == null || followedId == null) {
               System.out.println("unFollowUser Error");
10
               return "ErrorParameter";
11
12
           }
           return userService.unFollowUser(followId, followedId) ? "true" :
13
    "false";
14
       }
```

获取粉丝数量

```
1
         * @param id 当前用户的id
 2
 3
         * @return 返回粉丝数量
         * @description 获取用户被多少人关注,即粉丝数量
 4
 5
         */
        @RequestMapping("/getFollowedCount")
 6
 7
        public String getFollowedCount(Integer id) {
 8
            if (id == null) {
9
                System.out.println("getFollowedCount Error");
                return "ErrorParameter";
10
11
            }
12
            Long count = userService.getFollowedCount(id);
13
            return count + "";
14
        }
```

获取关注人的数量

```
1
        /**
 2
         * @param id 当前用户的id
 3
         * @return 返回关注人数
 4
         * @description 获取用户关注了多少人
 5
        @RequestMapping("/getFollowCount")
 6
 7
        public String getFollowCount(Integer id) {
8
            if (id == null) {
9
                System.out.println("getFollowCount Error");
10
                return "ErrorParameter";
11
            }
12
            Long count = userService.getFollowCount(id);
13
            return count + "";
14
        }
```

获取用户信息

```
/**
 1
 2
            * @param id 用户的id
 3
            * @return 返回user对象的JSON串,没找到对应的用户则是"false"
            * @description 根据id获取用户的信息,用于点击某个用户头像进入个人主页
 4
        @RequestMapping("/getUser")
 6
 7
        public String getUser(Integer id) {
8
           if (id == null) {
9
               System.out.println("getUser Error");
10
               return "ErrorParameter";
11
           }
12
           User user = userService.getUser(id);
13
            return user != null ? JSON.toJSONString(user) : "false";
14
        }
```

获取当前用户的关注列表

```
1
 2
         * @param id 当前用户的id
 3
         * @return 返回一个用户列表的Json串,如果为空则返回empty
 4
         * @description 获取当前用户的关注列表
 5
        @RequestMapping("/getFollowUsers")
 6
 7
        public String getFollowUsers(Integer id) {
8
            if (id == null) {
9
               System.out.println("getFollowUsers Error");
10
                return "ErrorParameter";
11
           }
12
            List<User> userList = userService.getFollowUsers(id);
13
            return userList.size() != 0 ? JSON.toJSONString(userList) : "empty";
        }
14
```

获取当前用户的粉丝列表

```
/**
 1
 2
         * @param id 当前用户的id
 3
         * @return 返回一个用户列表的Json串,如果为空则返回empty
         * @description 获取当前用户的粉丝列表
 4
 5
        @RequestMapping("/getFollowedUsers")
 6
 7
        public String getFollowedUsers(Integer id) {
8
            if (id == null) {
                System.out.println("getFollowedUsers Error");
 9
10
                return "ErrorParameter";
11
           List<User> userList = userService.getFollowedUsers(id);
12
13
            return userList.size() != 0 ? JSON.toJSONString(userList) : "empty";
14
        }
```

更改用户头像

```
1
        * @param headPortrait 使用OkHttp3传输图片,类型为MultipartFile,名称为
 2
    headPortrait
 3
         * @param id
                              用户id
         * @return 返回修改成功还是失败"true"或者"false"
 4
 5
        * @description 更改用户头像
 6
        */
 7
        @RequestMapping("/modifyUserHeadPortrait")
        public String modifyUserHeadPortrait(MultipartFile headPortrait, Integer
 8
    id) {
9
            if (headPortrait == null || id == null) {
                System.out.println("modifyUserHeadPortrait Error");
10
                return "ErrorParameter";
11
12
            }
            return userService.modifyUserHeadPortrait(headPortrait, id) ? "true"
13
    : "false";
14
       }
```

更改用户昵称

```
1
 2
         * @param userName 用户新昵称,请求参数名称为userName
                     用户id
 3
         * @param id
        * @return 返回修改成功还是失败"true"或者"false"
4
 5
        * @description 更改用户昵称
 6
        */
 7
        @RequestMapping("/modifyUserName")
8
        public String modifyUserName(String userName, Integer id) {
9
           if (userName == null || id == null) {
10
               System.out.println("modifyUserName Error");
               return "ErrorParameter";
11
12
           }
13
           return userService.modifyUserName(userName, id) ? "true" : "false";
14
       }
```

更改用户性别

```
1
        /**
 2
         * @param sex 用户性别字段,1或者0
 3
         * @param id 用户id
         * @return 返回修改成功还是失败"true"或者"false"
 4
 5
        * @description 修改用户性别
        */
 6
        @RequestMapping("/modifyUserSex")
 7
8
        public String modifyUserSex(Integer sex, Integer id) {
            if (sex == null || id == null) {
9
10
                System.out.println("modifyUserName Error");
                return "ErrorParameter";
11
12
            }
13
            return userService.modifyUserSex(sex, id) ? "true" : "false";
14
        }
```

身份证号认证

```
1
        /**
 2
        * @param id
                                用户id
 3
         * @param residentIdCard 身份证号
 4
         * @param realName 真实姓名
         * @return 返回是否认证成功, "true"或者"false"
 5
 6
         * @description 根据用户id设置用户身份证号
 7
        */
 8
        @RequestMapping("/idCardAuthentication")
        public String idCardAuthentication(Integer id, String residentIdCard,
    String realName) {
10
            if (id == null || residentIdCard == null || realName == null) {
                System.out.println("idCardAuthentication Error");
11
                return "ErrorParameter";
12
13
            }
            return "true".equals(userService.idCardAuthentication(id,
14
    residentIdCard, realName)) ? "true" : "false";
15
        }
16
    }
```

Activity相关

```
package com.hebtu.havefun.controller;
 2
    import com.alibaba.fastjson.JSON;
 3
 4
    import com.google.gson.Gson;
    import com.hebtu.havefun.entity.activity.Activity;
 5
    import com.hebtu.havefun.entity.activity.ActivityDetail;
    import com.hebtu.havefun.s ervice.ActivityService;
    import org.springframework.web.bind.annotation.RequestMapping;
 8
 9
    import org.springframework.web.bind.annotation.RestController;
10
    import javax.annotation.Resource;
11
    import java.util.List;
12
13
14
    /**
15
    * @author PengHuAnZhi
16
    * @createTime 2020/11/21 15:23
    * @projectName HaveFun
17
    * @className ActivityController.java
18
19
    * @description TODO
20
    */
21 @RestController
    @RequestMapping("activity")
    public class ActivityController {
23
24
25
        @Resource
26
        ActivityService activityService;
```

获取轮播图在服务器上的地址

```
/**

* @return 返回的是一个json串,解析出来是一个list<String>,字符串分别是图片在服务器上的地址

* @description 获取轮播图在服务器上的地址

*/

@RequestMapping("/getRotationChartPictures")
public String getRotationChartPictures() {
    return new Gson().toJson(activityService.getRotationChartPictures());
}
```

获取活动列表

```
1
2
        * @param activityKind 热门或者近期,1或者0
3
                             页码
        * @param pageNum
        * @param pageSize 页大小
4
        * @return List<Activity>集合的JSON串,没有活动则是"empty"
5
6
        * @description 获取活动列表
7
8
       @RequestMapping("/getActivityList")
       public String getActivityList(Integer activityKind, Integer pageNum,
    Integer pageSize) {
10
           if (activityKind == null || pageNum == null || pageSize == null) {
11
               System.out.println("getActivityList Error");
               return "ErrorParameter";
12
13
           }
```

```
String activities = activityService.getActivityList(activityKind, pageNum, pageSize);
return "empty".equals(activities) ? "" : activities;
}
```

获取活动详细信息

```
1
 2
         * @param activityId 活动的id
         * @return 返回ActivityDetail的json串,如果没有找到对应的活动详细信息则返回空串""
 3
         * @description 获取活动详细信息
 5
 6
        @RequestMapping("/getActivityDetail")
 7
        public String getActivityDetail(Integer activityId) {
 8
            if (activityId == null) {
 9
                System.out.println("getActivityDetail Error");
10
                return "ErrorParameter";
11
            }
12
            String activityDetail =
    activityService.getActivityDetail(activityId);
            return activityDetail != null ? activityDetail : "";
13
14
        }
```

判断是否收藏

```
1
 2
         * @param id
                            用户id,注意不是getUserId,是getId
 3
         * @param activityId 活动的id
 4
         * @return 返回true或者false的字符串
 5
         * @description 判断是否收藏
         */
 6
 7
        @RequestMapping("/judgeCollected")
        public String judgeCollected(Integer id, Integer activityId) {
 8
 9
            if (id == null || activityId == null) {
10
                System.out.println("judgeCollected Error");
                return "ErrorParameter";
11
            }
12
13
            boolean flag = activityService.judgeCollected(id, activityId);
            return flag ? "true" : "false";
14
15
        }
```

获取收藏的活动列表

```
1
2
        * @param id
                        用户id,注意不是getUserId,是getId
3
        * @param pageNum 页码
4
        * @param pageSize 页大小
5
        * @return 返回List<UserCollectActivity>集合,如果没有就返回"empty"
        * @description 获取收藏的活动列表
6
7
        */
8
       @RequestMapping("/getCollectedActivities")
       public String getCollectedActivities(Integer id, Integer pageNum,
    Integer pageSize) {
10
           if (id == null || pageNum == null || pageSize == null) {
```

```
System.out.println("getCollectedActivities Error");
return "ErrorParameter";
}
return activityService.getCollectedActivities(id, pageNum, pageSize);
}
```

添加活动

```
/**
 1
 2
        * @param files
                                 客户端传递若干MultipartFile文件,要求请求的参数名
    字都
 3
                                  是file(就是要重复),这边接收就是自动加入
    List<MultipartFile>中
4
        * @param activityDetailJson 客户端将封装好的ActivityDetail类转换为Json串发送
    过来,
5
                                  注意ActivityDetail里面的Activity类也得把数据都
   封装好,
6
                                  添加操作,不需要设置id值,因为数据库id是自增的
 7
        * @return 返回一个添加成功或者失败"true"或者"false"
        * @description 添加活动,接受一个ActivityDetail类的Json串数据,且参数名称为
    activityJson
        */
9
10
       @RequestMapping("/addActivity")
11
       public String addActivity(@RequestParam("file") List<MultipartFile>
    files, String activityDetailJson) {
12
          if (activityDetailJson == null) {
              System.out.println("addActivity Error");
13
14
              return "ErrorParameter";
15
16
           return activityService.addActivity(files, activityDetailJson)?
    "true" : "false";
17
       }
```

获取报名的活动列表

```
1
 2
        * @param id
                     用户id,注意不是getUserId,是getId
 3
         * @param pageNum 页码
 4
        * @param pageSize 页大小
 5
        * @return 返回List<UserCollectActivity>集合,如果没有就返回"empty"
        * @description 获取报名的活动列表
 6
 7
        */
        @RequestMapping("/getEnterActivities")
 8
9
        public String getEnterActivities(Integer id, Integer pageNum, Integer
    pageSize) {
10
           if (id == null || pageNum == null || pageSize == null) {
               System.out.println("getEnterActivities Error");
11
                return "ErrorParameter";
12
13
            return activityService.getEnterActivities(id, pageNum, pageSize);
14
15
        }
```

获取发布的活动列表

```
2
         * @param id
                     用户id,注意不是getUserId,是getId
 3
         * @param pageNum 页码
 4
        * @param pageSize 页大小
 5
        * @return 返回List<UserCollectActivity>集合,如果没有就返回"empty"
 6
         * @description 获取发布的活动列表
 7
        */
 8
        @RequestMapping("/getPublishActivities")
        public String getPublishActivities(Integer id, Integer pageNum, Integer
    pageSize) {
           if (id == null || pageNum == null || pageSize == null) {
10
               System.out.println("getPublishActivities Error");
11
12
                return "ErrorParameter";
13
           }
14
            return activityService.getPublishActivities(id, pageNum, pageSize);
        }
15
```

根据若干条件筛选活动

```
1
 2
         * @description 根据若干条件筛选活动,每个参数都给我传过来,如果用户没有做出对应的选
    择, String类型传empty, Integer类型传-1
 3
        * @param typeName 小类活动的名称
        * @param lowCost 价格区间的低区间
 4
 5
        * @param highCost 价格区间的高区间
        * @param howManyDays 近多少天
 6
        * @param city 市
        * @param county ⊠
8
9
        * @param pageNum 页码
10
        * @param pageSize 页大小
11
        * @return 返回一个List<Activity>集合的JSON串,如果集合为空,返回字符串"empty"
12
        */
13
        @RequestMapping("/screenActivities")
        public String screenActivities(String typeName, Integer lowCost, Integer
14
    highCost, Integer howManyDays, String city, String county, Integer pageNum,
    Integer pageSize) {
15
            if (howManyDays == null || pageNum == null || pageSize == null ||
    typeName == null || lowCost == null || highCost == null || city == null ||
    county == null) {
16
               System.out.println("screenActivities Error");
               return "ErrorParameter";
17
18
           }
           String activityList = activityService.screenActivities(howManyDays,
19
    typeName, lowCost, highCost, city, county, pageNum, pageSize);
20
            return !"empty".equals(activityList) ?
    JSON.toJSONString(activityList) : "empty";
21
       }
```

判断用户是否报名活动

```
1/**2* @param id用户id3* @param activityId 活动id4* @return 返回已报名"true",未报名"false"5* @description 根据用户id和活动id查询用户是否报名这个活动
```

```
*/
6
 7
        @RequestMapping("/judgeEnterActivity")
 8
        public String judgeEnterActivity(Integer id, Integer activityId) {
 9
            if (id == null || activityId == null) {
                System.out.println("judgeEnterActivity Error");
10
11
                return "ErrorParameter";
12
            }
            return activityService.judgeEnterActivity(id, activityId);
13
        }
14
```

修改活动信息

```
/**
 1
 2
         * @param activityDetailJson 修改后的ActivityDetail对象的JSON串
         * @return 返回修改成功"true"
 3
 4
         * @description 修改活动信息
 5
         */
        @RequestMapping("/modifyActivity")
 6
 7
        public String modifyActivity(String activityDetailJson) {
8
            if (activityDetailJson == null) {
 9
                System.out.println("modifyActivity Error");
10
                return "ErrorParameter";
            }
11
            return activityService.modifyActivity(activityDetailJson);
12
13
        }
```

删除活动

```
1
 2
         * @param activityId 活动id
 3
         * @return 返回删除成功"true"或者"false"
 4
         * @description 删除活动,数据库状态字段置0
 5
         */
        @RequestMapping("/deleteActivity")
 6
 7
        public String deleteActivity(Integer activityId) {
            if (activityId == null) {
8
9
                System.out.println("deleteActivity Error");
10
                return "ErrorParameter";
11
            return "true".equals(activityService.deleteActivity(activityId)) ?
12
    "true" : "false";
13
        }
14
    }
```