1. 登录
   1. 登录
      1. 前台注册接口

**a.接口url:**

pub/admin/register.do

**b.接口说明:**

前台注册的接口

相关系统设置信息：注册用户名限制 : 注册密码限制 同一IP注册间隔限制 : 注册用户邮箱唯一性检查 :

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| register | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| account | String | | 账号 | | |
| password | String | | 密码 | | |
| ra | String | | 验证码（选填-视项目情况） | | |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "登陆成功", //信息

"flag": "1"， //标识

“token”:“12343487sjdfb”， //token值

"userId": "sdofhov" //用户id

}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置，判断该注册是否与系统设置匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 根据account查找人员主表是否存在相同账号，如果存在，则flag为0，msg为账号已存在
3. 从session中取出验证码，和上传验证码对比，不相等，则返回flag为0，msg为验证码错误
4. 如果以上3点均匹配，则
   * + 1. 生成一个产生一个UUId,再对这个UUID字符化UUID1
       2. 获得系统参数看是一次加密，还是二次加密，对密码进行加密处理
       3. 将相应数据存入人员主表内
       4. 返回flag为1，msg“为注册成功, UUID1”
     1. 前台登录接口

**a.接口url:**

pub/admin/userLogin.do

**b.接口说明:**

前台登陆检测的接口

相关系统设置信息：网络防火墙设置，记录登录日志 : 重新登录时间间隔 :。

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| Login1 | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| username | String | | 用户名 | | |
| password | String | | 用户密码 | | |
| ra | String | | 验证码（选填-视项目情况） | | |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "登陆成功", //信息

"flag": "1"， //标识

“token”:“12343487sjdfb”， //token值

"userId": "sdofhov" //用户id

}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置，判断该注册是否与(系统设置>系统参数配置>网络防火墙设置>前端用户配置)中相关项匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 根据account查找人员主表是否存在相同账号，如果不存在，则flag为0，msg为账号不存在
3. 从session中取出验证码，和上传验证码对比，不相等，则返回flag为0，msg为验证码错误（选）
4. 将上传的password与表中password进行比对，相符则返回flag为1，msg“操作成功, UUID1” 不相等则返回flag为0，msg为密码错误
   * 1. 后台登录接口

**a.接口url:**

pub/admin/adminLogin.do

**b.接口说明:**

后台登陆检测的接口

相关信息：后台管理员登录验证码 : 后台管理员登录认证码 : 后台登录次数限制 : 后台管理员记录登录日志 : 开启访问来源认证 : 开启后台来源认证 :以及防火墙设置中的后台管理员配置

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| Login1 | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| username | String | | 用户名 | | |
| password | String | | 用户密码 | | |
| ra | String | | 验证码（选填-视项目情况） | | |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "登陆成功", //信息

"flag": "1"， //标识

“token”:“12343487sjdfb”， //token值

"userId": "sdofhov" //用户id

}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置，判断该注册是否与系统设置匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 根据account查找人员主表是否存在相同账号，如果不存在，则flag为0，msg为账号不存在
3. 从session中取出验证码，和上传验证码对比，不相等，则返回flag为0，msg为验证码错误（选）
4. 将上传的password与表中password进行比对，相符则返回flag为1，msg“操作成功, UUID1”，不相等则返回flag为0，msg为密码错误
   * 1. 产生验证码接口

**a.接口url:**

pub/admin/randomcode.do

**b.接口说明:**

**向手机发送短信验证码**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| randomcode | | Void | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| Phone |  | | 手机号 | | |

**d.返回值示例与说明**

**e.功能实现说明**

1. 注册1个号码，负责发送短信
2. 验证手机号格式，并获取系统设置，查询验证码发送限制（如多少时间内可发送多少次验证码），超过限制则返回flag为0，msg。。。
3. 生成1个6位随机数字密码，调用注册的号码向上传的手机号发送短信，发送成功则将该验证码存入session，并返回flag：0.msg：验证码发送成功
   * 1. 修改密码

**a.接口url:**

pub/admin/modifyPwd.do

**b.接口说明:**

前台注册的接口

相关信息：验证注册IP、登录日志、防火墙设置、用户设置中的ip限制、时间限制、时间间隔、认证码限制等等。

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| register | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| account | String | | 账号 | | |
| password | String | | 密码 | | |
| ra | String | | 验证码（选填-视项目情况） | | |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "登陆成功", //信息

"flag": "1"， //标识

“token”:“12343487sjdfb”， //token值

"userId": "sdofhov" //用户id

}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置，判断该注册是否与系统设置匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 根据account查找人员主表是否存在相同账号，如果不存在，则flag为0，msg为账号不存在
3. 从session中取出验证码，和上传验证码对比，不相等，则返回flag为0，msg为验证码错误
4. 如果以上3点均匹配，则
   * + 1. 获得系统参数看是一次加密，还是二次加密，对密码进行加密处理，并将之存入password字段中
       2. 返回0000000000000000000000000000flag为1，msg“为密码修改成功, UUID1”