# 医惠单点登录认证方案

## 一、 背景

随着医院信息化过程的推进，开始借助信息化手段规范业务流程，解决临床、科研、教学、管理等不同职能的问题，医院开始陆续上线各种各样的系统。

由此，为了加强对业务系统和办公室系统的安全控管，提高信息化安全管理水平，贯彻互联互通评审标准，我们采用以token令牌为基础，设计了一套支持CA（云key、硬key）的统一身份认证系统（门户管理系统），业务系统只通过token作为参数接受单点认证中心（SSO）的授权，实现各个业务系统的统一登录入口。

## 二、 范围

本标准规定了医惠统一身份认证系统（门户）的总体架构、认证方式、接入流程。

为实现统一登录，提高医护人员的工作效率，减少医护人员在各个系统之间频繁登录的时间，特提供单点登录接口，需要各

个系统积极配合并满足单点登录所需要求。

# 三、 单点登录具体要求如下

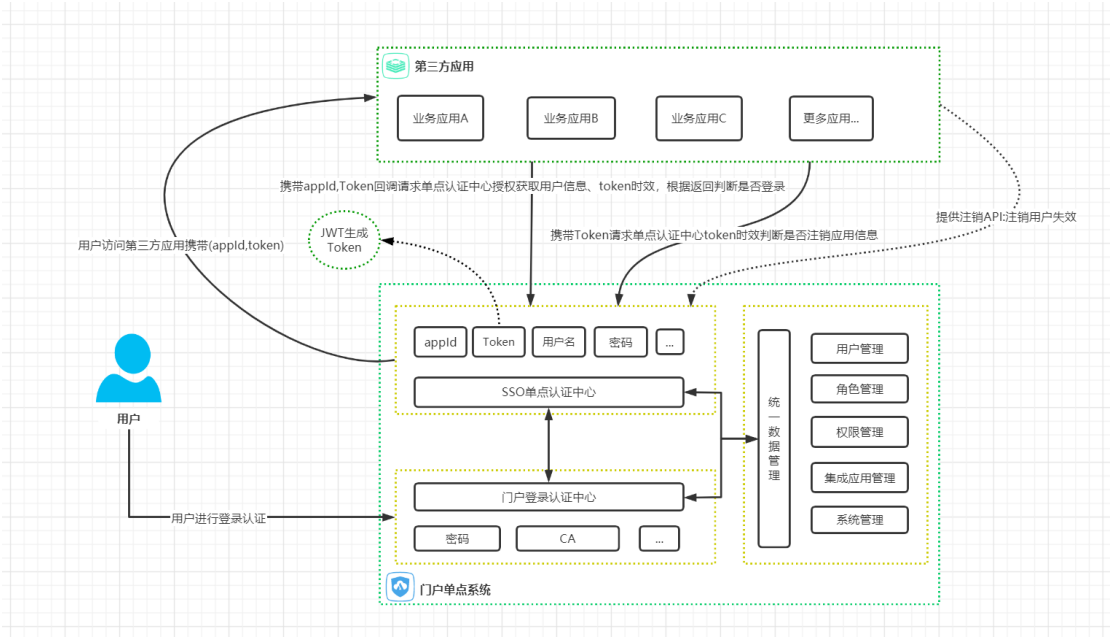
- 单点目前对于浏览器要求如下，外部跳转支持 Chrome、Firefox、IE、360，门户壳内跳转支持 Chrome、IE。
- BS 提供一个 URL 接收传入参数，参数为 appId、token 信息。
- CS 程序提供启动命令，可以接收参数，参数为 appId、token 信息。

# 四、 接入流程

## 资料准备

流程说明：

1. 待接入业务应用配置在门户单点系统，需配置内容参照表 1（应用申请单表格）。
2. 点击业务应用，会将在门户单点系统生成的 appid、token 通过参数方式传递给业务应用。
3. 业务应用在接收到 appid、token 需回调门户单点系统身份校验接口（userByPortal），通过返回值判断授权是否登录。




单点接入流程图 1

应用申请单位首先需要向医院信息中心提出申请，填写表 1 信息，成功之后由现场实施或者信息科根据表 1 信息在门户系统录入。

表 1　应用申请单表格

| | |
|---|---|
| 应用名称（必填） | 为中文名，应用名称要明确，需使用与应用功能相关的词汇，不能是日常通用性的描述词汇，不得出现测试 test 等宽泛字样； |
| 应用名称简称（必填） | 为中文名，应用名称的简称，方便应用在页面上显示； |
| 应用类别（必填） | 临床系统、财务系统、管理系统等 |
| 应用架构类型（必填） | BS、CS |
| 身份核验（门户提供） | 第三方业务系统的回调接口地址，用于验证用户有效性，并返回用户信息供第三方实现登录 |
| 应用访问 URL（必填） | 参照 BS 和 CS 测试方式。 |
| Token 有效验证 API(门户提供) | 用于验证用户登录状态，由第三方系统主动回调，判断的在线状态，若发现用户已经下线，则退出系统。 |
| 用户退出 API（门户提供） | 接入系统登出时，用户在第三方系统点击退出后，第三方系统要调用该接口进行用户的退出操作。 |
| 应用 Logo（必填） | 图片尺寸建议为 80px*80px |
| 应用描述（选填） | 简要描述应用功能，用于推荐宣传使用 |
| | |

# 门户单点登录方案

- BS 程序接入方案：

  提供URL以及参数appId、token，之后BS程序携带appId、token 回调授权接口获取获取用户信息实现本身登录进入系统首页（不再经过登录界面）；如果登录失败，直接返回到本身系统登录界面。

  测试方式：

| | |
|---|---|
| BS 的 url | http：//192.168.xxx.xxx:xxxx/xxxx |
| 接入门户的标识 | appId:xxxx |
| 门户认证参数 | token:xxxx |
| 请求方式 | post |

- CS 程序接入方案：

  提供 exe 启动命令以及参数 appId、token，之后 CS 程序携带 appId、token 回调授权接口获取获取用户信息实现本身登录进入系统首页（不再弹出系统登录界面框）；如果登录失败，直接返回到本身系统登录界面。

  测试方式：

| | |
|---|---|
| CS 的启动命令 | exe 目录 xxx.exe |
| 接入门户的标识 | appId:xxxx |
| 门户认证参数 | token:xxxx |
| 启动方式 | window 控制台窗口 |
| 示例 | exe 目录 xxx.exe appId:xxxx&token:xxxx |

## 身份校验方案

门户如接收到请求后，在请求头中获取 appId、token 串，验证接口检查 token 的真实有效性、时效性。如有效则同时返回用户的关键身份信息（如用户 ID、用户名、...）;若无效则返回错误信息，第三方系统返回登录界面并给予错误提示。

## Token 验证方案

第三方应用系统拿 token 请求验证接口，接口返回 token 时效，若有效，第三方应用继续有效操作，若无效，第三方系统返回登录界面并给予相应提示。

## 注销方案

1. 门户如接收到注销请求后，对本地认证登录认证标识做销毁过期处理，关闭从门户进入第三方的所有入口。

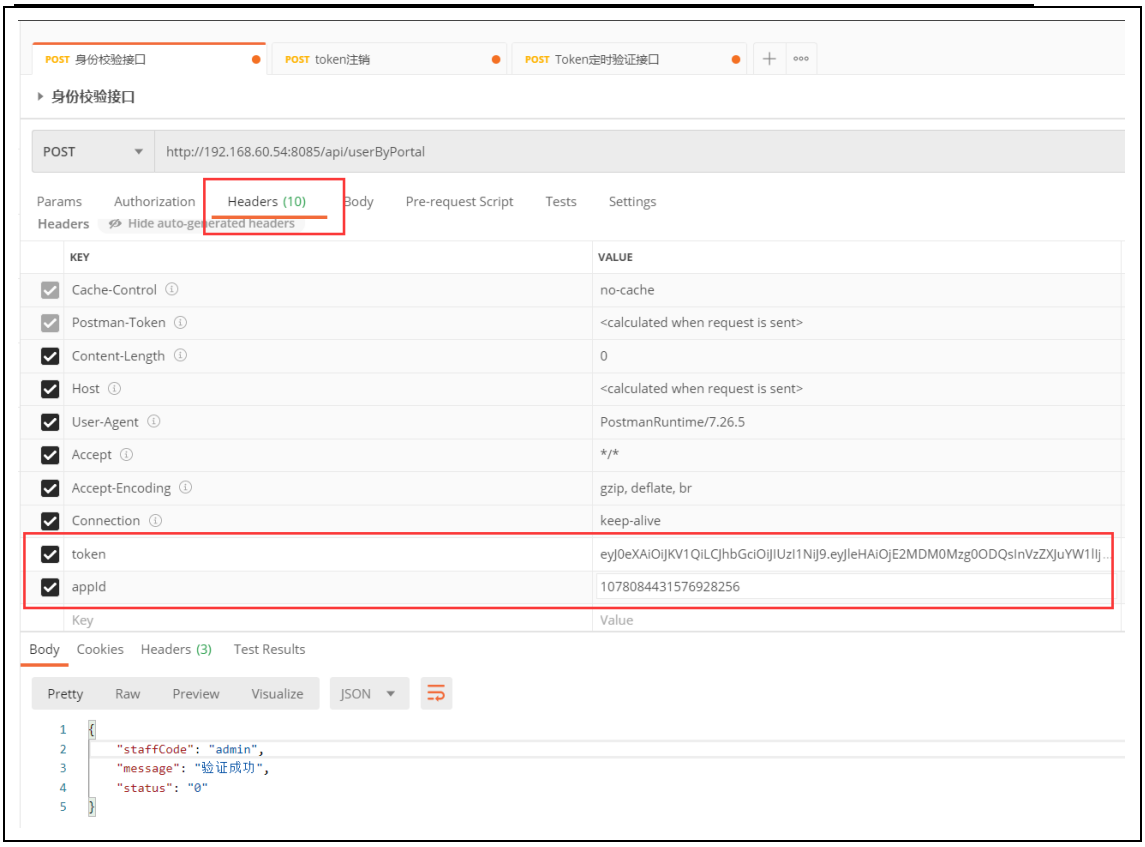
## 门户接口

以下是各个接口的说明

# 身份校验接口

| | | |
|---|---|---|
| 服务名称 | http://localhost:8085/api/userByPortal | |
| 服务说明 | 授权回调 | |
| 请求方式 | POST | |
| 请求头 | 参数名称 | 参数说明 |
| | appId | 接入门户的系统标识 |
| | token | 登录时获取到的用户标识 |
| 参数列表 | 参数名称 | 参数说明 |
| | | |
| | | |
| | | |
| 返回值 | {<br>"status": “0”, //0：成功 1：失败<br>“staffCode”:”xxx”, //门户工号<br>“参数 A”:”xxxx”, //配置在门户的第三方参数<br>“参数 B”:”xxxx” , //配置在门户的第三方参数<br>……<br>"message": "返回描述"<br>} | |

示例:




代码示例：

BS:

```
@RequestMapping(value = "/loginPortal", method = RequestMethod.POST)
@ResponseBody
public void loginPortal(HttpServletResponse response, @RequestParam("appId") String appId, @RequestParam("token") String token)
{
    String loginType = "front";
    ResultInfo resultInfo = new ResultInfo();
    Map map = new HashMap();
    map.put("appId",appId);
    map.put("token",token);
    String resMsg = "";
    try {
        resMsg = HttpClientUtil.httpPost( url: "http://192.168.10.75:8085/api/userByPortal",map);
        System.out.println("httpResponse success==="+resMsg);
    } catch (IOException e) {
        e.printStackTrace();
    } catch (CaHelperException e) {
        e.printStackTrace();
    }
    JSONObject jsonObject = JSONObject.fromObject(resMsg);
    String status = jsonObject.get("status").toString();
    String staffCode = jsonObject.get("staffCode").toString();
    String username = "";
    String pwd = "";
    System.out.println("status==="+status);
```

门户提供的http接口

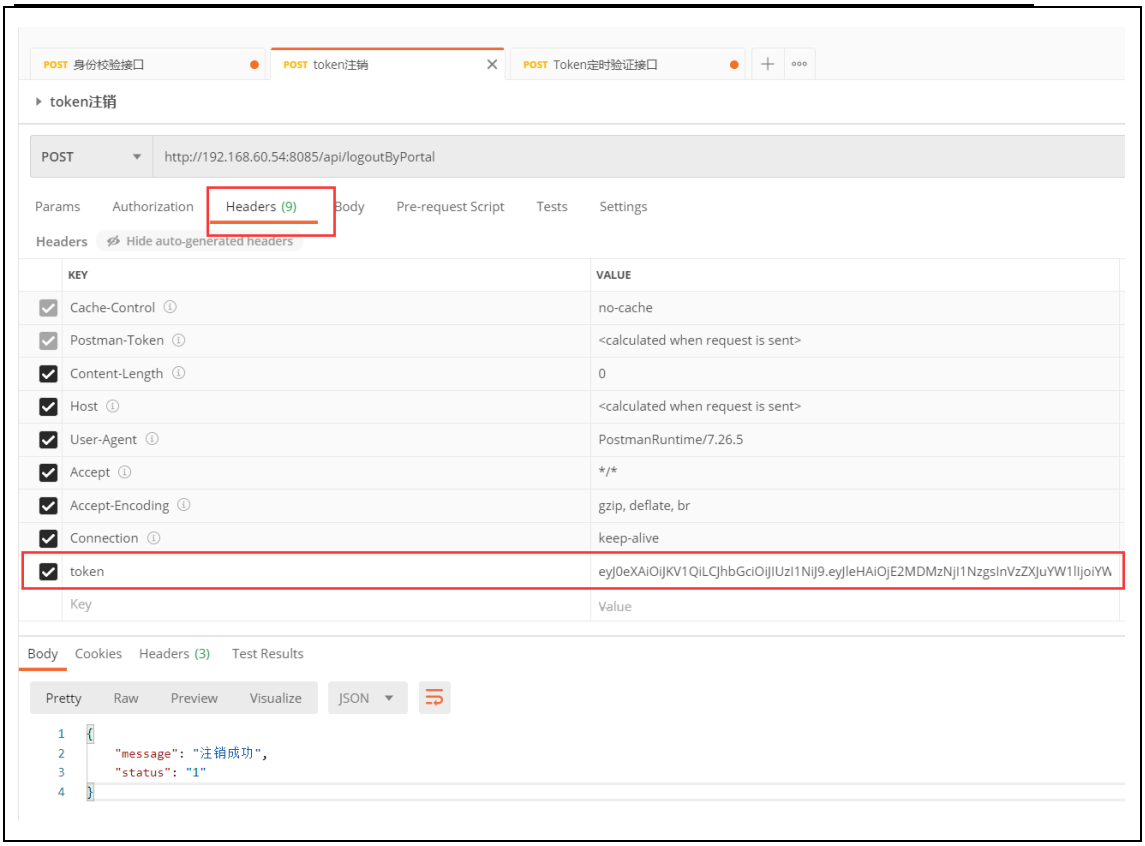
CS:

```
internal protected virtual Process StartProcess(LoginArguments args)
{
    var startInfo = new ProcessStartInfo();
    startInfo.FileName = args.Config.AppPath;
    startInfo.UseShellExecute = false;
    startInfo.WorkingDirectory = Path.GetDirectoryName(args.Config.AppPath);
    if (!string.IsNullOrEmpty(args.Config.AppArgs))
    {
        startInfo.Arguments = args.Config.AppArgs
            .Replace("@appId", args.AppId)
            .Replace("@token", args.Toekn);
    }
    try
    {
        return Process.Start(startInfo);
    }
    catch (Exception ex)
    {
        // 打开失败,尝试用管理员权限打开
        bool isAdmin = new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator);
        if (!isAdmin) startInfo.Verb = "runas";
        return Process.Start(startInfo);
    }
}
```

## 注销接口

| | | |
|---|---|---|
| 服务名称 | http://localhost:8085/api/logoutByPortal | |
| 服务说明 | 注销 | |
| 请求方式 | POST | |
| 请求头 | 参数名称 | 参数说明 |
| | appId | 接入门户的系统标识 |
| | token | 登录时获取到的用户标识 |
| 参数列表 | 参数名称 | 参数说明 |
| | | |
| | | |
| | | |
| 返回值 | {<br>"status": “0”, //0：成功 1：失败<br>"message": "返回描述"<br>} | |

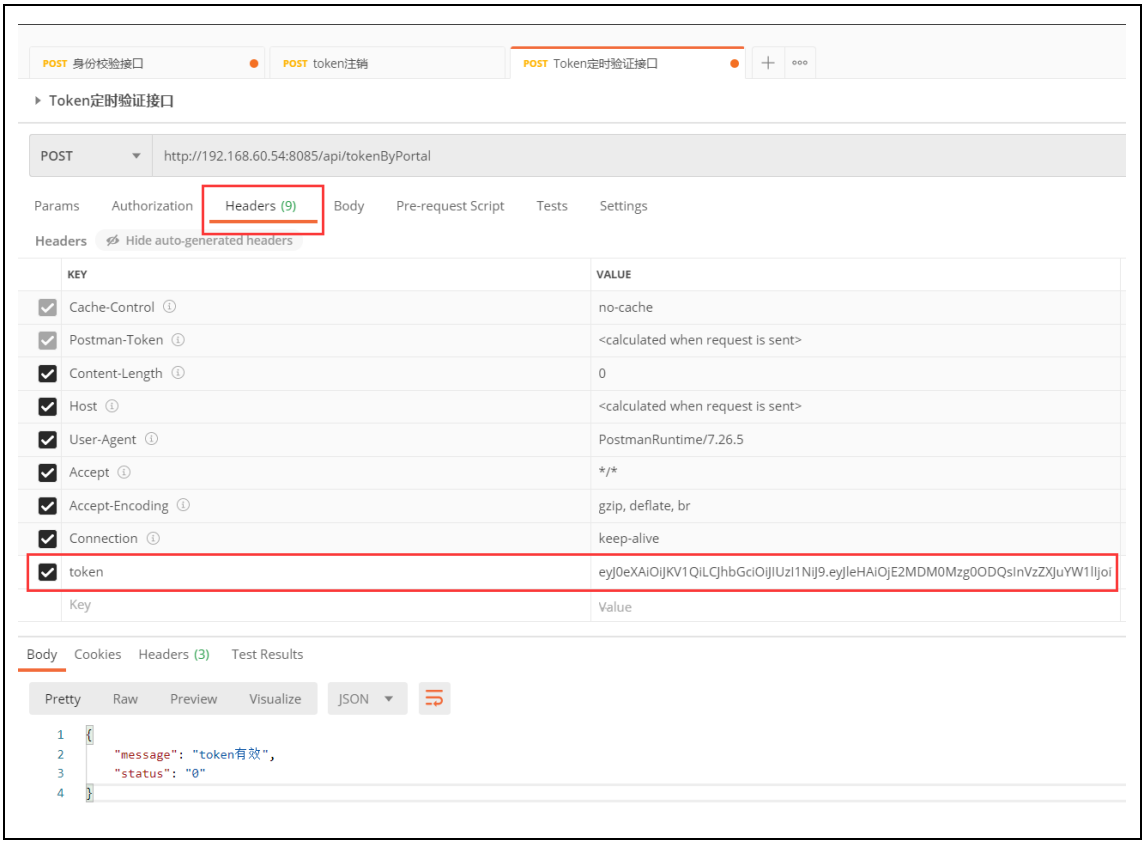
示例:




## Token 验证接口

| | | |
|---|---|---|
| 服务名称 | http://localhost:8085/api/tokenByPortal | |
| 服务说明 | Token 时效验证 | |
| 请求方式 | POST | |
| 请求头 | 参数名称 | 参数说明 |
| | appId | 接入门户的系统标识 |
| | token | 登录时获取到的用户标识 |
| 参数列表 | 参数名称 | 参数说明 |
| | | |
| | | |
| | | |
| 返回值 | {<br>"status": “0”, //0：成功 1：失败<br>"message": "返回描述"<br>} | |

示例:


POST 身份校验接口
POST token注销
POST Token定时验证接口
Token定时验证接口
POST
http://192.168.60.54:8085/api/tokenByPortal
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Headers
Hide auto-generated headers
KEY
VALUE
Cache-Control
no-cache
Postman-Token
<calculated when request is sent>
Content-Length
0
Host
<calculated when request is sent>
User-Agent
PostmanRuntime/7.26.5
Accept
*/*
Accept-Encoding
gzip, deflate, br
Connection
keep-alive
token
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE2MDM0Mzg0ODQsInVzZXJuYW1lIjoi
Key
Value
Body
Cookies
Headers (3)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"message": "token有效",
"status": "0"
}