单点登录和JWT校验网关接入说明书

V2.0（2020/07/27）

# 概述

单点登录系统提供两种登录（也就是获取JWT）方式：一种是为通过浏览器访问的web页面提供的、一种是为移动app或后台系统提供的，下面进行分别介绍。

登录地址（测试环境）：

登录页面：<https://cars.crbim.win:9346/sso/login>

API接口地址：<https://cars.crbim.win:9346/sso/api/v1.0/login>

登录地址（生产环境）：

登录页面：[https://sso.r93535.com/login](https://cars.crbim.win:9346/sso/login)

API接口地址：[https://sso.r93535.com/api/v1.0/login](https://cars.crbim.win:9346/sso/api/v1.0/login)

# web页面的登录和注销

## 2.1、页面登录

需要接入单点系统的web应用，只需采用**JWT校验网关**进行访问路由转发即可。

单点登录系统提供统一的登录页面，当web应用采用**JWT校验网关**进行路由转发后，所有未登录的请求都会转发到该登录页面，因此各应用不需要再提供自己的登录页面。

登录后的所有请求，都会经JWT校验网关的校验，网关会将JWT中的负载信息解密，从中得到本次请求的用户ID，然后将用户ID添加到本次请求的Header中的**CRBIMUID**变量中，这样对各应用系统的请求都会携带这个**CRBIMUID**，应用系统后台从Header中获取这个UID值即可。

## 2.2、注销

系统提供统一的注销功能，各应用的注销功能只需调用单点登录系统的注销功能即可，调用方式如下：



调用URL：http://ip:port/logout

注意：

1. 调用该功能后，该用户登录的所有应用都会注销。
2. 对于只想注销本应用的，只需清除本次调用的cookie就可以了。

# 3、移动端应用通过API接入

单点登录系统为移动端app提供API进行登录、注销和刷新功能，登录后app端可以携带JWT通过JWT校验网关来访问自己的应用接口。

**公钥：**

|  |
| --- |
| MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCEJiVVdmEzgRk/Lqg+I8JyavmI6eRZCv9d3rJ/yDGUS0BbxrrUrhdK4t/hLZUjkjtCsX80eLxecn6HkMBHaEM0tQYmZyxpB5NBPHioggQHDrJVskzDdqliHobem+X54INPznlXXwLtRl4vh1/ducI7laD59bVg/I+h/SrepgOvqwIDAQAB |

## 3.1、登录接口

|  |  |  |
| --- | --- | --- |
| **接口编号** | 3.1 | |
| **接口名称** | 登录接口 | |
| **接口说明** | 通过账号、密码进行登录 | |
| **接口地址** | http://ip:port/api/v1.0/login | |
| **调用协议** | POST | |
| **请求参数** | | |
| appname | 应用名称 |  |
| account | 用户账户 |  |
| password | 密文密码 | 使用基础数据平台的AES加密算法得出的密文 |
| timestamp | 请求时间戳 | 调用本api时的时间戳 |
| token | 本次请求的令牌 | 采用RSA加密算法，技术原理如下  token=encrypt(publicKey, data)  data=  {  “timestamp”:"",  “appname”:"",  “password”:"",  “account”:""  } |
| 失败返回值  {  "code": "1",  " msg": "错误信息提示"  } | | |
| 成功返回值  {  "code": "0",  "msg": "success",  "jwt":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmb28iOiJiYXIifQ.VAoRL1IU0nOguxURF2ZcKR0SGKE1gCbqwyh8u2MLAyY",  "expiredtime": "yyyy-MM-dd hh:mm:ss"  " token ": " token "（请求时的token）  } | | |

Token生成规则

1. Token应使用发布的公钥进行加密(所给公钥为base64编码)
2. 被加密的数据data为json格式
3. 为了防止重放攻击，时间戳有效期为当前时间上下浮动5分钟
4. 所以参数均为非空！
5. 时间戳格式如："timestamp":"1595817807552"
6. RSA加密算法使用RSA/ECB/PKCS1Padding，Token使用算法生成密文后应进行十六进制编码(为适应各系统 windows/linux/mac等)
7. 相关算法示例见最后部分附件

## 3.2、注销接口

|  |  |  |
| --- | --- | --- |
| **接口编号** | 3.2 | |
| **接口名称** | 注销接口 | |
| **接口说明** | 注销当前登录 | |
| **接口地址** | http://ip:port/api/v1.0/logout | |
| **调用协议** | POST | |
| **请求参数** | | |
| appname | 应用名称 |  |
| jwt | 当前JWT |  |
| timestamp | 请求时间戳 | 调用本api时的时间戳 |
| token | 本次请求的令牌 | 采用RSA加密算法  token=encrypt(publicKey, data)  data=  {  “timestamp”:"",  “appname”:"",  “jwt”:""  } |
| 失败返回值  {  "code": "1",  " msg ": "错误信息提示"  } | | |
| 成功返回值  {  "code": "0",  " msg ": "success",  " token ": " token "（请求时的token）  } | | |

## 3.3、刷新JWT接口

|  |  |  |
| --- | --- | --- |
| **接口编号** | 3.3 | |
| **接口名称** | 刷新JWT接口 | |
| **接口说明** | 用老的未过期的JWT换取新的JWT，以延长JWT的过期时间 | |
| **接口地址** | http://ip:port/api/v1.0/reload | |
| **调用协议** | POST | |
| **请求参数** | | |
| appname | 应用名称 |  |
| jwt | 用户账户 |  |
| timestamp | 请求时间戳 | 调用本api时的时间戳 |
| token | 本次请求的令牌 | 采用RSA加密算法  token=encrypt(publicKey, data)  data=  {  “timestamp”:"",  “appname”:"",  “jwt”:""  } |
| 失败返回值  {  "code": "1",  " msg ": "错误信息提示"  } | | |
| 成功返回值  {  "code": "0",  " msg ": "success",  "jwt":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmb28iOiJiYXIifQ.VAoRL1IU0nOguxURF2ZcKR0SGKE1gCbqwyh8u2MLAyY",  "expiredtime": "yyyy-MM-dd hh:mm:ss"  " token ": " token "（请求时的token）  } | | |

# 4、获取JWT后，接口调用

铁路工程管理平台的所有应用都部署在单点网关后面，由单点网关进行负载，获取JWT后就相当于获取到了通过单点网关访问网关后所有系统的钥匙。

所有的Http请求都要求在Cookie中携带这个JWT，Cookie是一个Key-Value格式的键值对，其中Key是CRBIMSSOJWT，Value就是你获得到的JWT。

# 5、相关算法

基础数据平台的密码采用AES加密算法，java示例代码如下：



Token采用RSA加密算法，java示例代码如下：



*注：其中Java示例中的加解密、转码等操作均使用了hutool 类库的相关方法。*