ADVANTECH

Enabling an Intelligent Planet

WISE-PaaS 用户权限整合说明

tingting.ning

2020/05/09

WISE-PaaS 平台角色权限介绍

角色维度	角色	订阅号创建	添加订阅号一般用户	集群订阅管理（MP Buy）	集群资源访问（Kubectl）	工作空间管理（MP）	工作空间访问（Kubectl）	命名空间管理（MP）	命名空间访问（Kubectl）	应用订阅（APP Buy）	专属应用访问（APP Login）	托管服务订阅（Service Buy）	托管服务访问（Service Login）
	全球数据中心管理员 Gobal Admin	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×
平台管理 Platform Management	数据中心管理员 Datacenter Admin	×	×	✓	✓	✓	✓	✓	✓	✓	×	✓	×
	集群管理员 Cluster Admin	×	×	✓	✓	✓	✓	✓	✓	×	×	×	×
订阅号管理 Subscription Management	Subscription Admin 订阅号管理员	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Subscription User 订阅号一般用户	×	×	×	✓	✓	✓	✓	✓	×	✓	×	✓
租户空间管理 Tenant Space Management	集群所有者 Cluster Owner	×	×	×	✓	✓	✓	✓	✓	×	✓	×	×
	工作空间所有者 Workspace Owner	×	×	×	×	×	✓	✓	✓	×	✓	×	×
	命名空间开发者 Namespace Developer	×	×	×	×	×	×	×	✓	×	✓	×	×
应用管理 APP Management	应用访问者 SRPUser	×	×	×	×	×	×	×	×	×	✓	×	×

ADVANTECH

整合注意事项

- 订阅号admin 和订阅号user 对订阅号订阅的app 默认有最高的访问权限，调用SSO接口，整合参考第8, 9页
- Cluster Owner 默认有空间下面部署的应用的最高访问权限。
- Workspace Owner 默认有空间下面部署的应用的最高访问权限。
- Namespace Developer依据SRP自己的情况给权限，一般给viewer的比较多。
- 以上空间资源权限的整合调用MP的接口获取用户在当前空间下的最高资源权限即能判断给出权限。整合参考第10页
- 如果有服务间整合的需求可整合服务间调用的客户端token，使用方法参考ppt 第13页
- SSO 和MP 整合建议大家用内网地址，速度会比较快。这个地址可从环境变量中取得，要读取的变量是global.ensaasApps，部部署时会将这些链接注入到环境变量中

```
global:
  database:
    secretName: datahub-all-in-one-secret
  ensaasApps:
    apiDccs:
      externalUrl: "https://api-dccs-ensaas.sa.wise-paas.com/v1"
      internalUrl: "http://api.dccs.ensaas.en.internal/v1"
    apiLicense:
      externalUrl: "https://api-license-ensaas.sa.wise-paas.com/v1"
      internalUrl: "http://api.license.ensaas.en.internal/v1"
    apiListingsystem:
      externalUrl: https://api-listingsystem-ensaas.sa.wise-paas.com/v1
      internalUrl: "http://api-listingsystem.ensaas.en.internal"
    apiMg:
      externalUrl: "https://api-mg-ensaas.sa.wise-paas.com/v2"
      internalUrl: "http://api.mg.ensaas.en.internal/v2"
    apiService:
      externalUrl: "https://api-service-ensaas.sa.wise-paas.com/v2"
      internalUrl: "http://api.service.ensaas.en.internal/v2"
    apiSso:
      externalUrl: "https://api-sso-ensaas.sa.wise-paas.com/v4.0"
      internalUrl: "http://api.sso.ensaas.en.internal/v4.0"
    apiMp:
      externalUrl: "https://api-mp-ensaas.sa.wise-paas.com/v1"
      internalUrl: "http://api.mp.ensaas.en.internal/v1"
    ensaas:
      datacenterCode: "sa"
      externalUrl: "sa.wise-paas.com"
      internalUrl: "en.internal"
```

MP和SSO权限拆分之后还兼容的接口

SA, JE, HZ 5月9号已上线拆分后的版本，用户的资源权限管理会移入MP进行管理， SSO只负责订阅号权限管理

MP和SSO进行权限拆分，主要为了较少耦合，提高效率，目前还兼容已提供给服务整合的接口，可获取订阅号和资源权限，由SSO的接口去调用MP的接口获取，来保证兼容性。 **已整合这两个接口的服务不受影响。**

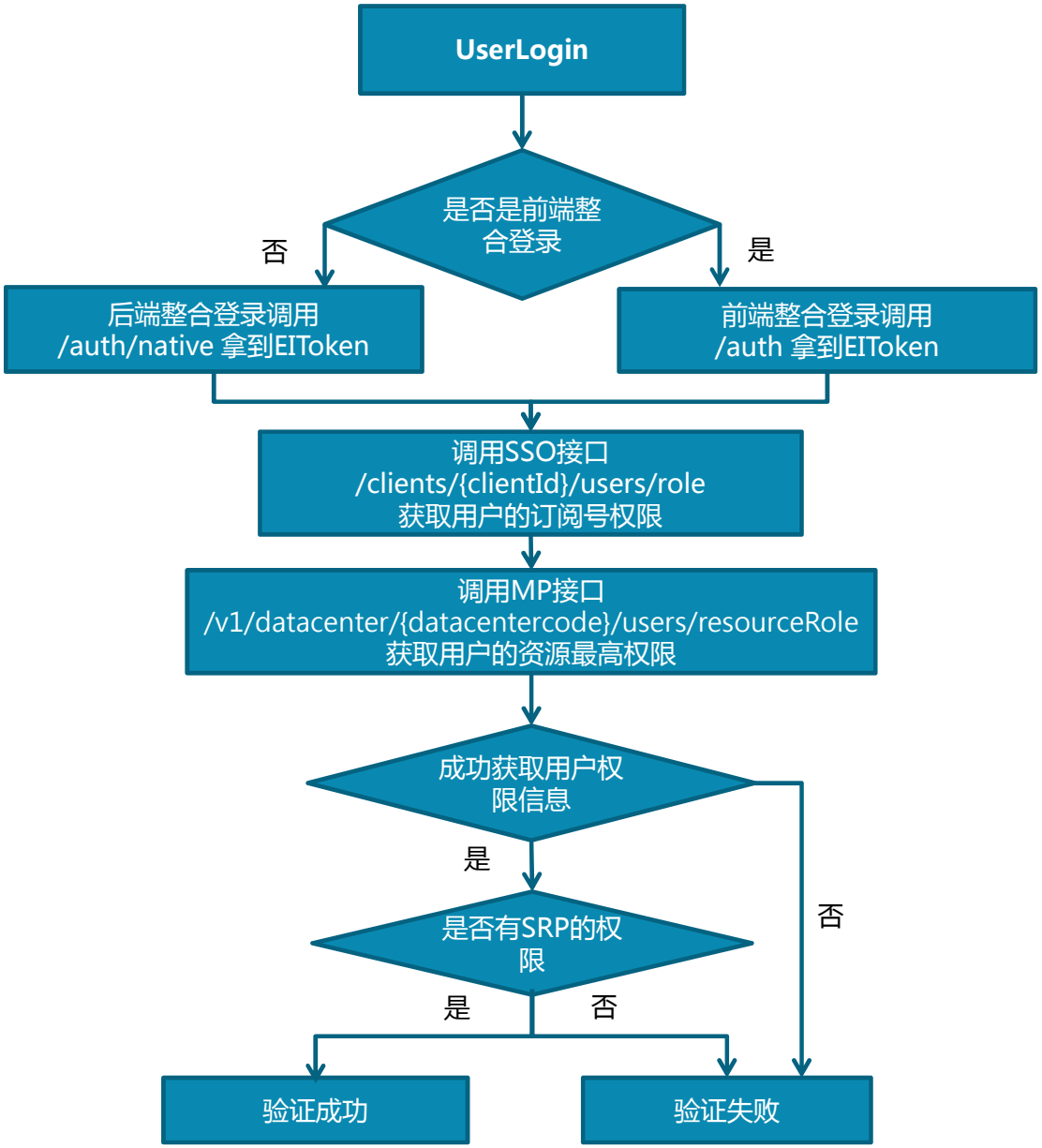
GET /users/srpRole?clientId=xxx&subscriptionId=xxx

GET /users/me

ADVANTECH

整合流程

用户验证流程：


UserLogin
是否是前端整合登录
否
是
后端整合登录调用 /auth/native 拿到EIToken
前端整合登录调用 /auth 拿到EIToken
调用SSO接口 /clients/{clientId}/users/role 获取用户的订阅号权限
调用MP接口 /v1/datacenter/{datacentercode}/users/resourceRole 获取用户的资源最高权限
成功获取用户权限信息
是
否
是否有SRP的权限
是
否
验证成功
验证失败

整合流程差异 – Token获取

API:

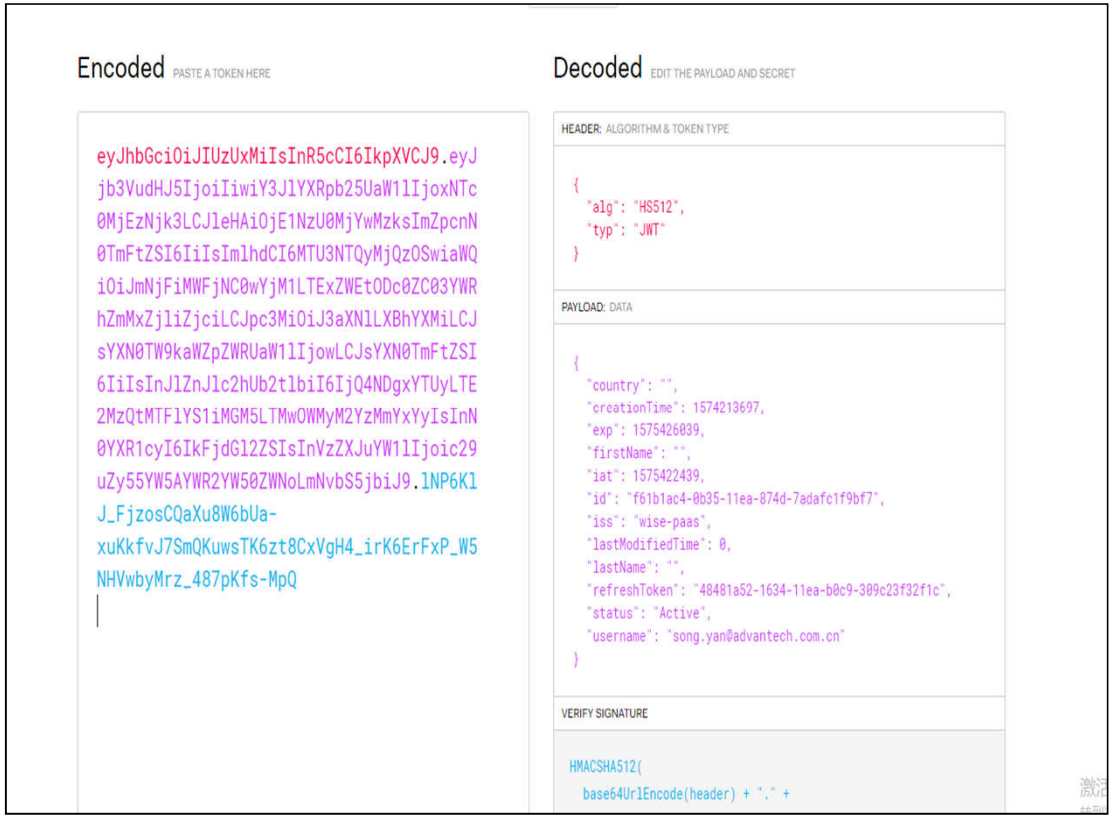
Post　/auth　(前端使用)

Post　/auth/native　（后端使用）

Token 变短解决的问题：

v4版本的token大大缩减了长度，只包含了用户的基本信息，从而解决了cookie超过长度限制的问题（chrome cookie长度不能超过4kb，v2.0版本cookie长度超过限制导致token信息不完全）

Token内容



前端整合SSO的Token刷新建议方式

EIToken :

时效一小时

属性 (attributes): HttpOnly (提供给后端 backend), secure (https)

内容：加密的账户信息，包括状态、过期时间、RefreshToken等等

获取方法：帐号密码获取、RefreshToken获取

EIName :

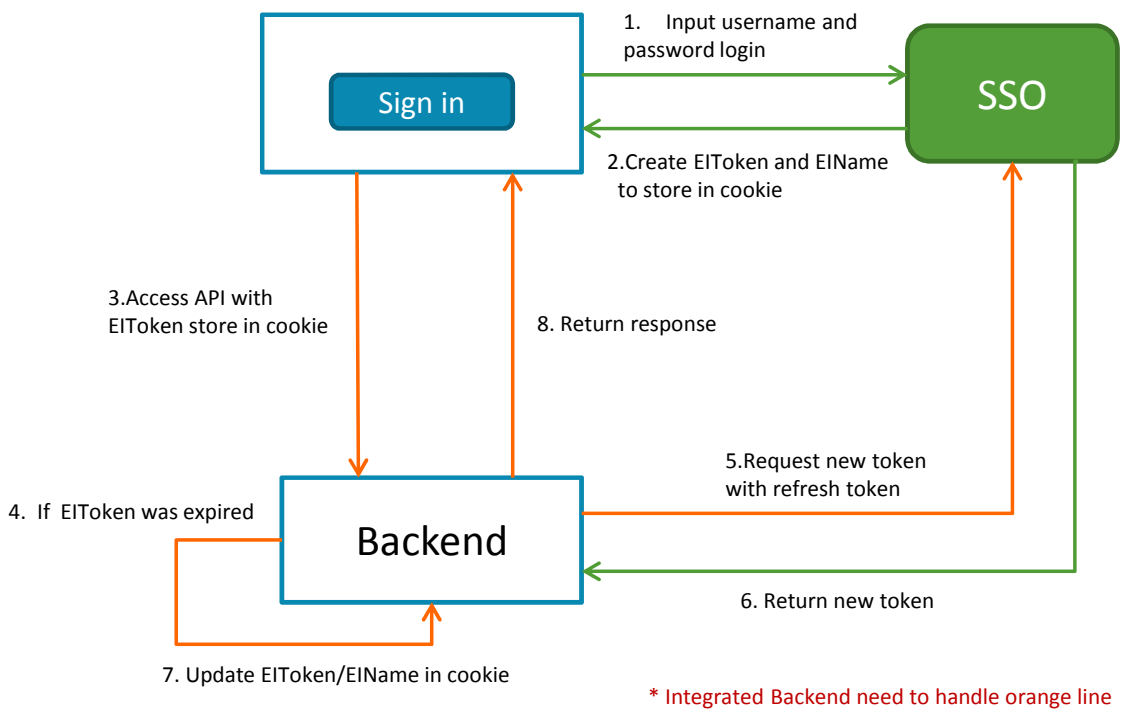
时效一小时

属性 (attributes): secure、前端程序如 java script 可以取得，并显示在页面上，并作为登入成功的识别

Token刷新的接口:

POST /token

POST /tokenvalidation



整合流程 – 订阅号权限获取

整合SSO接口获取，订阅号权限由SSO进行管理

V4版本新增一支API，用户只需要传入clientId ，并使用user的EIToken （短的）进行调用，SSO 即可返回如下信息，免去Client端整合时自己解析Token的负担，并在此接口中会进行Token有效性的验证，不用整合方再调用token验证的接口

1. User对Client的ClientRole （scope中的权限）
2. User在传入的订阅号下的权限

GET /clients/{clientId}/users/role ? subscriptionId=xxx

Response：

```
{
    "clientRole"        :  "admin",
    "subscriptionRole"  :  "admin"
}
```

如果上面那个权限没有就返回是空字符串，根据以上的权限进行判断可知道要给当前的用户什么权限

Note：

1. 如果是不部署在用户空间的服务，要整合订阅号用户放行登录就可以，而不需要整合MP接口获取资源权限。
2. 如果不在平台上架，可不整合 subscription用户相关的登录权限，调用这个接口也可不传入subscriptionId的参数

ADVANTECH

如何获取订阅号ID

此部分为平台的上架可订阅服务需要整合的部分

订阅号admin 和订阅号user 对订阅号订阅的app 默认有最高的访问权限。

APP调用License server的API获得订阅此APP的订阅号的id， license server 整合可联系wenjing.wang

/v1/api/partNum/licenseQty?pn=partNumber&id=xxx

partNumber 是产品自己的料号，id是clustername+workspaceId+namespaceName 直接拼接组成，返回值格式如下：


```
{
    "id": "slave0420a957f4-0bf9-4faf-90cd-694919cd4b68dashboard",
    "subscriptionId": "22ec5794-dcdc-43a1-ab88-xxxxxx",
    "isValidTransaction": true,
    "number": 1,
    "authcode": "5785-xxxx-xxx",
     "activeInfo": "“
}
```


判断是否给订阅号用户权限的方式：

如果是调用/users/srpRole?clientId=xxx&subscriptionId=xxx， 传入app注册的ClientId 和从License server获取的subscriptionId，则SSO会帮助用户判断用户所有的订阅号是否包含app所属订阅号，并返回角色，否则为空

ADVANTECH

整合流程 – 资源权限获取

整合MP接口获取, 资源权限有MP进行管理

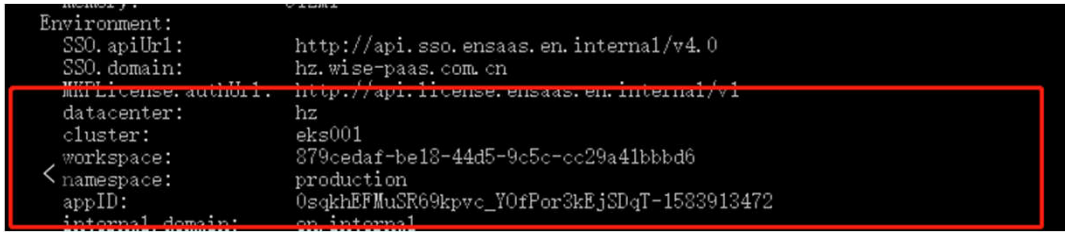
MP提供一支接口用来获取用户在给定资源条件下的最高资源权限，整合服务按照最高权限给用户分配权限即可。使用user的EIToken （短的）进行调用，MP 即可返回如下信息，免去服务整合自己解析资源权限的负担，并在此接口会进行Token有效性的验证。

API: GET /v1/datacenter/{datacentercode}/users/resourceRole

Params：

name	type	value	Is required
Authorization	header	SSO Token	true
cluster	query	Cluster name	true
workspace	query	workspaceId	true
namespace	query	namespaceName	true

Note：调用接口需要的datacentercode， clustername， workspaceId，namespaceName都可以通过服务自己的环境变量获取，部署的时候由平台注入



Response：

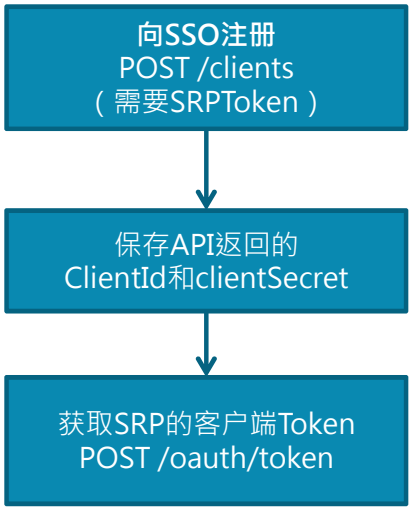
```
{
    HighestRole:namespacedeveloper
}
```

Example： GET/v1/datacenter/{datacentercode}/users/resourceRole?cluster=ews001&workspace=bc7b7397-9e73-47c2-8349-09def78ed868&namespace=production

Response：200 OK

```
{
  "HighestRole":"workspaceOwner"
}
```

整合流程差异 – Client 注册



V2.0
post /srps

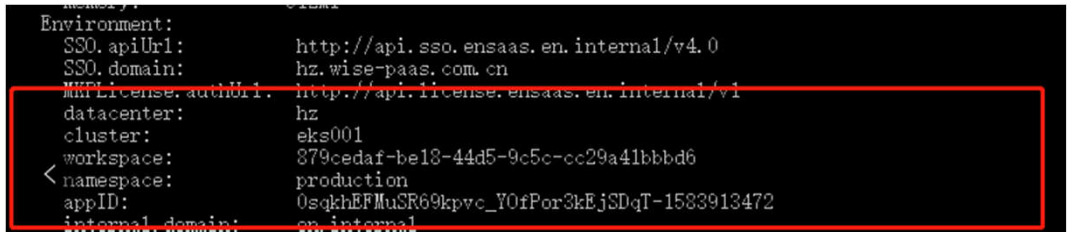
```
{
 "name": "string",    *required
 "appId": "string",    *required
 "orgId": "string",    *required
 "spaceId": "string", *required
 "scopes": [          *required
  "string"
 ]
}
```

V4.0
post /clients

```
{
 "appName": "string",  *required
 "appId": "string",       *required
 "serviceName": "string", *required
 "cluster": "string",      *required
 "workspace": "string",  *required
 "namespace": "string", *required
 "datacenter": "string",   *required
 "redirectUrl": "string",
 "scopes": [
  "string"
 ]
}
```

Notes：

1. Client注册成功后应该保留**clientId**以及**clientSecret**，方便以后添加srpUser或者获取clientToken
2. appName 等同于2.0的SRPName， serviceName是上架的服务名
3. 注册过的client id 和client Secret可以使用这种接口查询 **GET /clients/{clientIdOrName}**
4. 如果不是用标准OAuth整合的方式，redirectUrl的参数不用给，如果是OAuth整合会检查注册时传入redirectUrl 和/oauth/authorize接口调用时传入的 redirect_uri 参数做对比，要求一致
5. cluster, workspace, namespace,datacenter, appId 都是从环境变量中读取，部署的时候由平台注入。



整合流程差异 – 灵活多层的用户权限

SRP Patch User Scopes

patch /users/{username}/scopes

1. Scope的选择支持json格式，可以表达复杂权限
2. SRP注册时如果没有填写scopes字段，在创建user时，user的scope 可以为任意的字符串。
3. V4.0 patch /users/{username}/scopes支持clientToken，同时兼容旧版v2.0的方式（srpToken）

```
{
  "orgRoles": [{
    "orgId": "1",
    "orgRole": "admin"
  }, {
    "orgId": "2",
    "orgRole": "admin"
  }]
}
```

V2.0: dashboard-123456789.admin

V4.0: dashboard-123456789.{"orgRoles":[{"orgId":"1","orgRole":"admin"},{"orgId":"2","orgRole":"admin"}]}

ADVANTECH

服务间调用鉴权

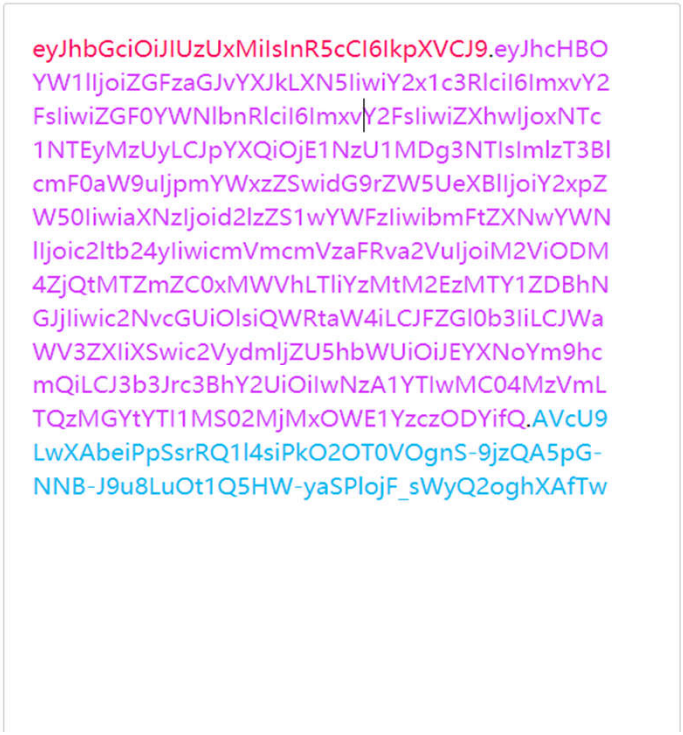
API: /oauth/token　获取服务间调用的ClientToken

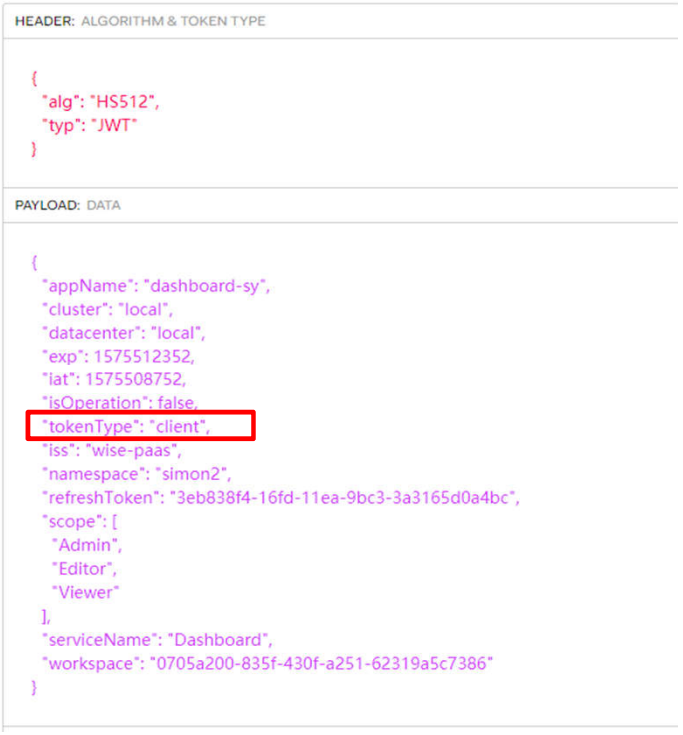
curl -X POST "http://172.21.92.144/v4.0/oauth/token?grant_type=client_credentials&**client_id**=dashboard-sy-1575450866&**client_secret**=NzdkNWM1YWMtMTY3Ni0xMWVhLTliYzMtM2EzMTY1ZDBhNGJj" -H "accept: application/json"

1. 默认是颁发1个小时过期的token
2. 如需使用不过期的clientToken, 在请求时加上参数 duration=eternal
3. client_id 和client_secret 是client注册的时候返回的

服务间调用的流程

ClientToken解析结果





ADVANTECH

SSO 4.0 on k8s优势

优势	4.0	3.0	备注
性能提升	平均API提升约50倍	性能慢	
新增订阅号管理	新增订阅号管理，可以管理资源和订阅号之间的关系，方便进一步运营管理	无此功能	
解决token过长set 到cookie无法获得完整信息的问题	/auth 获得的Token只提供基本用户信息， 提供/users/me API获取用户详细信息	v2.0版本token太长导致set到浏览器cookie中的信息不全	chrome cookie长度不能超过4kb
SSO管理的权限层次丰富	对于scope中的权限开放可以管理多层次的权限， 比如可以注册dashboard的用户 org是什么，这个org下的权限是什么	只有单一的一层，比如admin，editor，viewer等	
快速获取用户访问client端的权限，免去自己解析Token	新增接口免去Client端解析Token复杂格式获取权限。 新增/users/srpRole 接口只需传入clientId，即可返回和此client相关的角色，权限是什么	各SRP整合需要解析Token，Token格式有变动需要更新	降低整合门槛
提供服务间鉴权机制	新增OAuth2.0 标准客户端Token	无此功能	

API 文档地址

SSO API: https://api-sso-ensaas.hz.wise-paas.com.cn/public/apidoc/index.html

Co-Creating the Future of the IoT World