Q/SH

国家能源集团有限责任公司企业信息化标准

Q/SH XA888－V8

国家能源身份管控平台技术标准

应用系统接入规范

报批稿

2017-07-10

XXXX- XX -XX 发布　　XXXX-XX -XX 实施

国家能源集团信息管理部　　发布

目　　录

前　言

国家能源身份管控平台(以下简称管理平台或平台)是集团信息化一体化纵向管理平台的重要组成部分。一体化纵向管理平台是依托数据仓库、集成总线、门户、业务流程管理、内容管理、统一身份管理等支持功能实现纵向业务信息的穿透和决策支持系统的应用支撑平台。国家能源身份管控平台对集团信息系统进行用户身份信息、组织结构信息的统一集中管理，并为应用系统提供访问控制功能。

本规范规定了国家能源身份管控平台与第三方应用系统的身份供应接口、访问控制接口的相关规范。

本部分由集团信息管理部提出。

本部分由集团信息管理部归口。

国家能源身份管控平台第三方应用系统的集成和开发规范

1 范围

本规范适用于集团需要和国家能源身份管控平台（以下简称“平台”）进行集成的第三方应用系统。其中，身份供应相关内容适用于 B/S、C/S 系统，访问控制适用于 B/S 系统。

本规范规定了第三方应用系统在建设时需要遵守的技术规范，包括身份供应接口、访问控制接口。

本规范对和平台进行集成的第三方应用系统提供指导和技术限定。

本规范针对的人员为第三方应用系统的技术开发人员、国家能源身份管控平台技术人员。

2 目标

通过遵循和适用本标准，达到以下目的：

a)指导业务系统软件前期的身份和权限管理的设计。

b)指导业务系统软件开发阶段中的身份和权限管理的实现。

c)指导业务系统软件实施阶段中的统一权限平台的接入实施工作。

3 规范性引用文件

《身份管理平台技术标准应用系统接入规范——v2.0》
《身份管理平台技术标准应用系统接入规范——v3.0》

4 术语和定义

下列术语和定义适用于本文件。

4.1 单点登录 （SSO） single sign on

在多个应用系统中，用户只需要登录一次（即提供一次身份凭证）就可以访问所有相互信任的应用系统,包括可以将这次主要的已认证的会话映射到其它应用中用于同一个用户的会话的机制。

4.2 身份供应

指平台将用户账号信息供应到应用系统,根据集团现有信息架构,身份管理平台仅对外供应账号ID与账号状态。详细人员信息(例如人员所属组织关系、业务手机号码、身份证等)与组织信息需要与MDM主数据系统对接提供。

4.3 第三方应用系统

第三方应用系统是需要接收国家能源身份管控平台用户身份信息的业务应用系统，对于B/S系统还可实现统一的访问控制。移动端、C/S端暂不支持。

4.4 业务应用客户端

业务应用客户端是业务系统的用户通过它请求业务应用服务端，展现业务应用服务端提供的服务，实现业务系统业务功能的介质。

4.5 业务应用服务端

业务应用服务端是业务系统为用户提供各种业务功能的服务，部署在服务器上的应用服务。

4.6 国家能源统一身份管理平台

又称国家能源身份管控平台是集团信息化一体化纵向管理平台的重要组成部分，对集团信息系统进行用户账号信息、应用授权信息的统一集中管理，为应用系统提供访问控制功能。

4.7 身份供应接口

身份供应接口是指第三方应用系统，按照平台技术标准开发实现的服务接口。

4.8 访问控制接口

访问控制接口是指由统一身份管理平台提供的统一认证服务接口，基于OAUTH2.0技术标准实现。第三方应用可以通过HTTP方式依次调用平台接口,实现登录人的身份识别。目前平台仅提供全局认证功能，第三方应用授权由应用自行控制。

5 平台架构及集成技术流程

5.1 平台架构

国家能源身份管控平台由身份认证、身份推送和管理控制台三个组件构成:

- 身份推送组件：为第三方应用系统提供身份信息供应，接口调用等接口服务。
- 身份认证组件：为第三方应用系统提供访问时的身份验证和访问控制功能。
- 管理控制台：为管理员用户提供用户管理界面。

5.2 集成方案简述

第三方业务系统接入平台一般需要进行访问控制接口改造和身份供应接口开发。

身份供应接口是为了实现第三方系统的账号的创建、禁用、启用、信息更新、查询等操作，权限管控平台负责调用业务系统的接口，实现账号信息的准实时推送。技术上统一约定基于Webservice

SOAP1.2规范,且通过HTTP方式调用（暂不支持TCP/RFC方式调用）并按照本规范提供的接口文档开发实现。

访问控制接口是为了实现第三方系统的登录认证功能，国家能源身份管控平台负责集中认证鉴权，并将认证结果反馈给业务系统。业务系统通过改造登录模块，信任平台提供的登录凭据，在本系统对登录账号授权后，提供业务服务。

5.3 平台与应用系统集成的技术流程

平台与应用系统集成的技术流程用以帮助应用系统的开发人员理解应用系统和平台集成的步骤，平台和第三方应用系统的集成分为身份供应和访问控制两部分。

身份供应是指将平台的用户身份信息通过第三方应用系统的身份供应接口供应到第三方应用系统。身份供应的技术流程如下图所示：

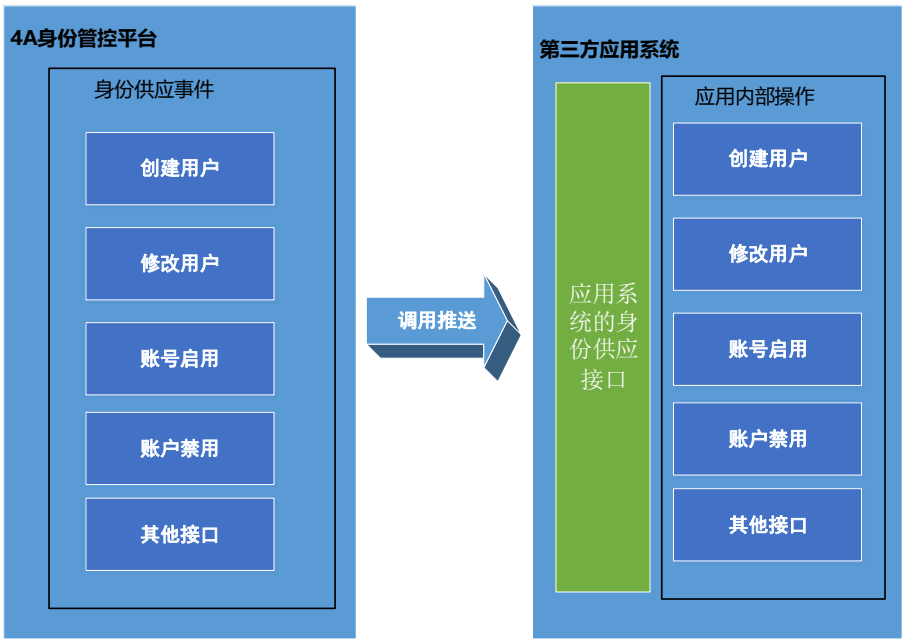


图2：身份供应的技术流程图

访问控制是指完成用户认证功能，确定用户能否访问应用系统，技术流程如下图所示：

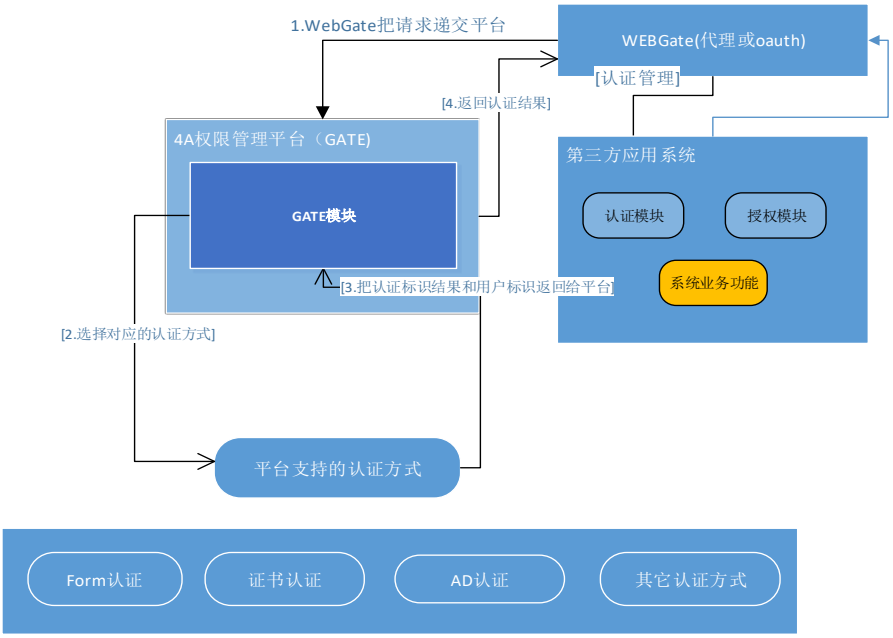


图3：访问控制流程图

6 集成流程

应用系统接入平台时应按照如下流程来进行集成：

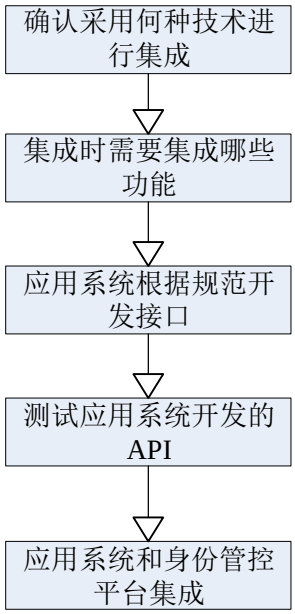


图4：集成流程图

（1）确认采用何种技术进行集成：原则上，应采用本规范实现第三方应用系统与国家能源身份管控平台的集成。具体技术选型请参照章节7中对不同应用分类推荐的集成方案。对于不能采用本规范进

行集成的情况，在选择接入方式时必须提供应用系统信息和实现方案，并且把选型过程在写入模板提交信息管理部。

（2）集成时需要集成哪些功能：该步骤由应用系统的特性决定，可以选择的接口参见“8 Web Service 身份供应接口”。

（3）应用系统根据规范开发接口：应用系统根据本规范进行接口开发，包括身份供应接口和访问控制接口的开发。

（4）测试应用系统开发的API：平台的接口测试程序将测试应用系统开发的接口。

（5）应用系统和国家能源身份管控平台集成：部署应用系统的API服务，和平台进行集成。

7 应用系统技术要求

7.1 通用技术要求

与国家能源身份管控平台进行集成，对第三方应用系统存在以下技术要求：

(1) 密码维护功能重定向

应用系统与国家能源身份管控平台集成后，应禁用系统自身的账号密码维护功能（或者不对普通用户展示），如密码修改，密码重置等，密码修改和重置功能由国家能源身份管控平台的个人信息自助管理插件提供。

(2) 用户禁用要求

应用系统与国家能源身份管控平台集成后，应禁用本系统账号的维护功能（如账号创建、删除等）。对于用户增删之外的管理功能(如修改)可根据具体情况进行保留。

(3) 系统登出

单系统登出：国家能源身份管控平台实现单点登出功能，应用系统中“注销”按钮更改为“关闭”，点击“关闭”时，业务系统关闭自身浏览器窗口，实现本系统的退出功能。普通业务系统的登出不能影响全局 SSO 会话有效性。

全局注销：该功能仅适用于 SD 服务平台与系统门户。应用系统需要将“注销”按钮的 url 变更为 4A 平台提供的全局注销地址（参照 9.2.3 注销会话），实现所有系统的统一注销功能。统一身份认证不提供单一会话机制，业务系统有需要时，需要自行管理其 session 有效性。

(4) 接口实现

第三方应用系统应遵循本规范实现身份供应接口、访问控制接口（C/S 应用暂无需实现访问控

制接口）。具体实现哪几个身份供应接口可根据实际情况进行选择确定，其余未真正实现的接口需返回约定的错误代码。对于通过其它的方式来和国家能源身份管控平台做集成的方案，则须经集团信息管理部审核批准。

(5) 身份供应请求重发判断

对于是否重发的身份供应的请求需依据请求中的参数 requestID 进行判断，建议应用系统记录 requestID, 以进行是否重发判断，方便问题诊断定位以及重值处理。

(6) 第三方应用系统域名

第三方应用系统的域名应为集团主域名下的二级或三级域名。(不允许使用 ip)

(7) 对于 B/S 方式的应用系统，浏览器客户端须支持 Cookie。

(8) 分配的应用 ID 和密钥，必须和真实业务一一对应。不得一对多、混用、冒用，集成信息变更需要及时通知统一身份管理平台进行信息变更登记。

(9) 应用系统应记录来自国家能源身份管控平台的身份供应日志，保留时间不短于一个月；

(10)应用系统供应记录来自用户访问的日志，保留时间不短于一个月。

7.2 方案适用场景简述

第三方系统厂商需根据下列方案描述，识别自己产品的所在分组，选择对应的规范进行实现。如果不能在表中找到对应的解决方案,需和信息管理部达成一致意见后,进行方案定制,并更新本规范文档。

- 身份供应方案：

  身份供应方案，为了保持国家能源信息化建设的一致性。延用规范 2.0 的约定，接口方法与字段原则上维持不变。

  适用于所有 C/S、B/S 套装软件或自研系统。

- 访问控制方案 A（强制）：

  该方案采用通用的 OAuth2.0 协议，具备通用、轻量、易接入的特点。同时能够方便的管理第三方的授权状态。

  适用于晚于平台上线的新接入的自研产品或系统、中小型产品套件。部署中间件一般为 Tomcat 或 Weblogic 或 WAS。

- 访问控制方案 B（废弃）:

  该方案延用 1003 的访问控制方式，最大限度地保证了与 1003 系统的兼容性，适合 0 改造的迁移现有业务系统。

适用于早于平台上线的已经接入过 1003 系统的自研产品或系统、中小型产品套件、大型产品套件等。

- 访问控制方案 C:

  该方案属于定制方案。由于技术发展或者业务系统限制，不能采用方案 A、方案 B 时，由平台运维方与业务系统方进行技术会商，制定符合实际需求的接入方案。并更新本文档。

8 身份供应接口

国家能源身份管控平台提供多种身份供应的连接器，如JDBC连接、Restful连接器等，为方便大量应用系统接入国家能源身份管控平台的管理和维护，规定以 Web Service方式的接口作为首选接口规范。第三方应用系统遵循该规范实现Web Service接口，技术上统一约定基于Webservice SOAP1.2规范，且通过HTTP方式调用（暂不支持TCP/RFC方式调用）并按照本规范提供的接口文档开发实现。 由国家能源身份管控平台进行调用，实现账号信息同步。详细人员信息(例如人员所属组织关系、业务手机号码、身份证等)与组织信息需要与MDM主数据系统对接提供，本接口不提供相关数据。

8.1 验证密钥

为保证Web服务调用的安全性，国家能源身份管控平台采用非对称密钥，遵循WS-Security标准，在身份供应时，使用国家能源身份管控平台私钥进行 Web Service 数字签名或加密，第三方应用系统使用国家能源身份管控平台的公钥进行数字签名验证或解密。目前该项目未作强制要求，安全性保障建议采用网络或者系统防火墙策略，限制调用方IP。

8.2 核心参数及返回结果

在Web Service接口中，核心的参数为用户身份信息和组织结构信息。

8.2.1 核心参数

8.2.1.1 requestID

requestID 为32位UUID，由身份管控平台调用身份供应接口时生成，用于唯一标识该请求。第三方应用系统（Web Service服务端）接收请求时，若判断该 requestID已处理，则返回上次已处理消息的结果。

8.2.1.2 用户信息

UserEntity

字段名	描述	类型	必须	长度	备注
userId	**账号标识**	**String**	**是**	**256**	**登录帐号标识**
Password	*密码*	*String*	*否*	*256*	*密码（预留，4A 平台不提供）*
orgCode	*直属单位编号*	*String*	*否*	*8*	*对应 MDM 组织机构-单位的编号（预留，4A 平台不提供）*
orgName	*直属单位名称*	*String*	*否*	*40*	*对应员工直接所属的 MDM 组织机构-单位名称（预留，4A 平台不提供）*
departmentNum	*直属部门编号*	*String*	*否*	*8*	*对应 MDM 组织机构-部门的编号（预留，4A 平台不提供）*
departmentName	*直属部门名称*	*String*	*否*	*40*	*对应 MDM 组织机构-部门的名称（预留，4A 平台不提供）*
firstName	**名**	**String**	**是**	**150**	**身份证姓名的名**
lastName	**姓**	**String**	**是**	**150**	**身份证姓名的姓**
Sex	*性别*	*String*	*否*	*10*	*员工的性别（预留，4A 平台不提供）*
userType	*用户类型*	*String*	*否*	*256*	*用户类型，为枚举值（预留，4A 平台不提供）*
Mail	*Mail 地址*	*String*	*否*	*256*	*员工的邮箱（预留，4A 平台不提供）*
displayIndex	*用户排序索引*	*integer*	*否*	*256*	*排序值（预留，4A 平台不提供）*
Mobile	*手机号码*	*String*	*否*	*50*	*员工的工作手机号,可能存在多值（预留，4A 平台不提供）*
Telephone	*公司电话*	*String*	*否*	*50*	*员工的办公室电话,可能存在多值（预留，4A 平台不提供）*
PID	*身份证号*	*String*	*否*	*50*	*身份证号(预留，4A 平台不提供)*
passportID	*护照号*	*String*	*否*	*50*	*护照号(预留，4A 平台不提供)*
userStatus	**账号状态**	**String**	**是**	**256**	**为枚举值：** **Active — 激活状态** **Disabled — 已禁止**
empNumber	*员工编号*	*String*	*否*	*256*	*员工在 ERP 系统中的唯一编号，默认与账号标识一致（预留，4A 平台不提供）*

Title	*员工主职位编号*	*String*	*否*	*40*	*(预留，4A 平台暂不提供)*
titleName	*主职位名称*	*String*	*否*	*256*	*职位的名称（预留，4A 平台不提供）*
isBywork	*是否兼职*	*String*	*否*	*10*	*为枚举值：（预留，4A 平台暂不提供）* ***全职— MDM 推送的全职人员信息*** ***兼职 —MDM 推送的兼职人员信息***
Extensions	*扩展信息*	*KeyValues[]*	*否*	*--*	*以 key-value 方式定义，当前面约定的字段不满足业务场景时，在此进行额外的字段约定*

8.2.1.3 时间戳

用于标识身份供应请求发出的日期及时间。

8.2.2 返回结果

所有接口调用都以OperationResult作为返回值类型，OperationResult中的操作成功是指业务方法的操作成功，比如，执行更新用户操作时，如果发现用户不存在，则应用系统应认为操作失败。操作失败时，returnCode和returnMsg将在OIM控制台上显示。当操作失败时， returnCode返回错误代码，returnMsg返回错误原因

OperationResult					
字段名	描述	类型	长度	备注	必须
requestID	**请求唯一标识**	**String**	**32**	**32 位编码**	**是**
returnFlag	**处理结果标识**	**boolean**		**true — 处理成功** **false — 处理失败**	**是**
returnCode	**返回结果编号**	**String**	**256**	**由应用方参照附录三返回编码值；** **当 returnFlag 为 true 时，returnCode 为 0；** **当 returnFlag 为 false 时，returnCode 为错误编码**	**是**
returnMsg	**返回结果信**	**String**	**256**	**由应用参照附录三返回，可自定义**	**否**

	息			**较为详细的简短描述，returnFlag 是 false 时，必须填写**	

注：具体错误代码约定，请参照附录3实现。服务接口应捕获处理所有异常。对调用方只返回约定的错误代码。

8.3 Web Service 身份供应接口

所有接入系统请严格按照提供的wsdl文件进行接口开发。第三方系统请依照附录提供的wsdl文件生成服务端(确保命名空间、方法名称、服务名、端口名一致），并进行内部逻辑开发。确实无需要的接口方法可以适当进行剪裁，但务必提供接口方法名和约定的异常标识。

8.3.1 创建新账号

接口名称	addAccount			
功能描述	在应用系统创建新账号			
参数	**字段名**	**类型**	**备注**	**必须**
	requestID	String	请求唯一标识。	是
	userEntity	UserEntity	账号信息。	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台创建的用户如果已经在第三方应用系统中存在，第三方应用系统需要更新用户信息并返回创建成功标志，不得直接报错并返回用户已存在。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	新建用户时需要查看是否已经存在用户数据，如果不存在，则创建成功后返回结果。 如果用户已存在，需更新用户信息，并根据 userStatus 状态禁用或启用当前用户，逻辑处理完毕后返回创建成功的结果。			

8.3.2 修改账号信息

接口名称	modifyAccount			
功能描述	修改账号可修改属性的服务			
参数	**字段名**	**类型**	**备注**	**必须**

	requestID	String	请求唯一标识。	是
	userEntity	UserEntity	账号信息	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台需要修改的用户如果在第三方应用系统中不存在，第三方应用系统需要返回用户不存在的消息。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	修改帐号时如果 UserEntity 中为增量传参。未做变化的字段为空，发生变化的字段会传空串或字符串。参数为空串时，业务应将当前字段修改为空串，参数为空时，业务应不对当前字段做任何处理。			

8.3.3 删除帐号

接口名称	deleteAccount			
功能描述	提供删除账号的服务，表示该用户已不具备权限再使用本系统。			
参数	**字段名**	**类型**	**备注**	**必须**
	requestID	String	请求唯一标识。	是
	userId	String	账号信息对象 Id	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台需要删除的用户如果在第三方应用系统中不存在，第三方应用系统需要返回删除成功。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	删除的服务需要应用系统自主决定对用户数据进行的删除方式（逻辑删除、物理删除），调用该接口后，查询接口应不能查询该账号状态。若账号删除成功，或者无法通过查询接口查询到，则认为删除成功。			

8.3.4 禁用账号

接口名称	suspendAccount

功能描述	提供暂挂一个有效账号的服务，挂起帐户是对用户状态进行修改，表示用户目前处于禁用或冻结状态，后期可以进行用户的启用。			
参数	字段名	类型	备注	必须
	requestID	String	请求唯一标识。	是
	userId	String	账号信息对象 Id	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台需要修改的用户如果在第三方应用系统中不存在，第三方应用系统需要返回用户不存在的消息。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	若账号有效，则暂时禁用账号并返回处理结果；如果该状态在挂起状态，则返回成功。			

8.3.5 恢复账号

接口名称	restoreAccount			
功能描述	提供恢复一个有效账号的服务，恢复帐户是对用户状态进行修改，表示用户目前处于可用状态，对应的操作为恢复或启用帐户			
参数	字段名	类型	备注	必须
	requestID	String	请求唯一标识。	是
	userId	String	账号信息对象 Id	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台需要修改的用户如果在第三方应用系统中不存在，第三方应用系统需要返回用户不存在的消息。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	若账号有效，则恢复账号并返回处理结果；如果该帐号已经在启用状态，则返回成功。			

8.3.6 查询帐号

接口名称	searchAccount			
功能描述	提供查询并返回全部或符合查询条件的账号集合的服务。根据查询条件查询并返回账号集合 查询条件的组合规则的说明如下： 若某个字段的值为 null 或空，则该条件字段可忽略； 对于 displayIndex 字段，如果值为 0 则认为此字段条件可忽略。 条件字段之间为“并（and）”的关系； 将所有不为 null 或空的条件字段值以“并(and)”的方式进行组合，并进行账号查询； 若某个条件字段值出现多字符通配符（*）、单字符通配符（？），则说明以模糊方式查询，否则为精确查询，以 displayName（显示名）字段举例： 张*：查询所有姓名以“张”开头的账号； *张：查询所有姓名以“张”结尾的账号； 张*三：查询所有姓名以“张”开头，以“三”结尾的账号； *张*：查询所有姓名当中包含“张”的账号； *：查询并返回所有账号；			
参数	字段名	类型	备注	必须
	requestID	String	请求唯一标识。	是
	userEntity	UserEntity	账号信息	是
	timeStamp	DateTime	时间戳	是
异常处理	当查询无此用户对象时，返回一个 userEntities 的空列表			
返回结果描述	参照下面“**SearchUserResponse**”定义。			

SearchUserResponse				
字段名	描述	类型	备注	必须
userEntities	账号信息对象	List<UserEntity>	如果没有搜索到信息，则返回空。返回的 UserEntity 应当包含账号的统一数据，例如：addAccount 已经从身份供应平台接收了 Mail，UserEntity 就应该返回 Mail 数据。	是

8.3.7 修改密码

接口名称	ChangePassword

功能描述	修改已有账号的密码的服务。			
参数	字段名	类型	备注	必须
	requestID	String	请求唯一标识。	是
	userId	String	账号 ID	是
	newPassword	String	修改后的密码	是
	timeStamp	DateTime	时间戳	是
异常处理	国家能源身份管控平台需要修改的用户如果在第三方应用系统中不存在，第三方应用系统需要返回用户不存在的消息。			
返回结果描述	参见“8.2.2 返回结果”。			
其它	兼容性接口，新接入系统不要求实现。			

9 身份认证接口

9.1 认证机制

国家能源身份管控平台的认证协议包括两大类：一是基于密码的直接认证；二是基于标准协议的单点登录。

密码认证由平台来统一管理对端系统的密码，即由平台来主动推送密码到不同系统所依赖的凭证库，密码采用固定的加密方式，由对端系统进行自行验证。

单点登录协议认证包括基于SAML2.0和Oauth2.0，对端系统通过选择一种协议来接入国家能源认证中心，由认证中心统一确认用户的凭证，然后再通过接口转换为本地系统的认证状态。

9.2 Oauth2.0 认证协议

9.2.1 授权类型

国家能源认证中心支持 Oauth2.0 协议中的所有授权类型：

- ✓ Authorization Code Grant
- ✓ Implicit Grant
- ✓ Resource Owner Password Credentials Grant
- ✓ Client Credentials Grant

在做集成设计时，根据不同的场景选择合适授权类型，出于集团安全性和统管性需求，强制第三方应用选择 Authorization Code Grant 方式来实现单点登录的集成。

9.2.2 集成流程

对于 B/S 模式的应用系统，原则上需要实现统一的访问控制及与国家能源集团应用中心的单点登录集成。统一认证是指认证请求由国家能源身份管控平台统一处理，应用系统需要实现访问控制模块。

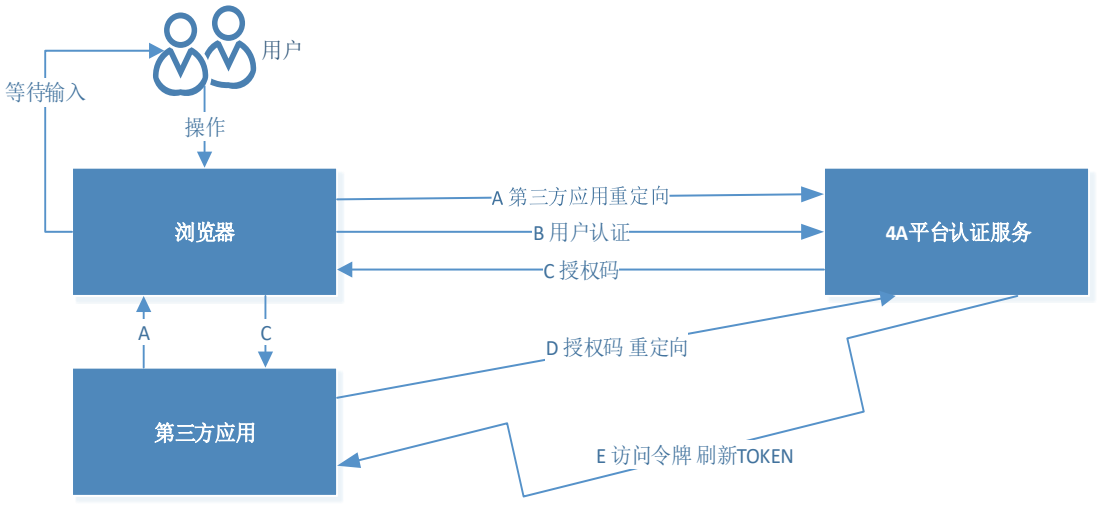
国家能源身份管控平台的统一认证组件集成首先推荐采用OAuth2.0协议，它是目前国际通用的授权方式，流程更简单。在应用系统的使用者登录应用系统时，应用系统向统一认证组件确认用户是否合法。

- 应用厂商向客户申请统一授权OAuth客户端ID和OAuth客户端秘钥
  - OAuth客户端ID：应用的唯一标识。

    在OAuth2.0认证过程中，OAuth客户端ID的值即为client_id的值。
  - OAuth客户端秘钥：对应的密钥

    访问用户资源时用来验证应用的合法性，在OAuth2.0认证过程中，客户端密钥的值即为client_secret的值。

- OAuth2.0 授权码认证过程如下:



（A）用户通过浏览器打开第三方应用以后，第三方应用要求用户给予授权。
（B）用户同意给予授权并在浏览器输入账户口令，提交到平台认证。
（C）第三方应用使用浏览器获得的授权码，向认证服务器申请令牌。
（D）认证服务器对第三方应用进行认证以后，确认无误，同意发放令牌。
（E）第三方应用使用令牌，用户可以访问所有资源。

9.2.3 技术规范

1. 基础信息

认证平台提供：client_id：例如 test_client

client_secret：例如 1e2e396b-2682-4080-9167-45c6901206bd

访问端点及数据结构：https://devid.shenhua.cc/auth/realms/sh4a/.well-known/openid-configuration

从该链接获取以下信息：

registration_endpoint：基础服务地址

authorization_endpoint：获取 code 的服务地址

token_endpoint：获取 access_token 的服务地址

userinfo_endpoint：获取 user_info 的服务地址

end_session_endpoint：登录状态注销服务

特别声明：以上所有地址及秘钥，均要写入配置文件，方便后期修改与环境迁移。并自行定义本登录模块保护的资源范围。一套 id 与 secret 唯一对应一套业务应用，不得共用、混用。

2. 业务应用获取 code

业务系统想要获取登录用户信息，需要先获取 code，需构造如下的链接：

authorization_endpoint?client_id=<LIENT_ID >&redirect_uri=<REDIRECT_URI>&response_type=code&scope=openid&state=<STATE>

本链接 支持 GET 和 POST 方法

参数	必须	说明
client_id	是	认证中心分配的 client_id，认证系统管理员确认
redirect_uri	是	授权后重定向的回调链接地址，请使用 urlencode 对链接进行处理，该地址需提前告知认证平台进行注册
response_type	是	返回类型，此时固定为：code
scope	是	应用授权作用域。 此时固定为：openid。
state	否	重定向后会带上 state 参数，业务系统可以填写 a-zA-Z0-9 的参数值，长度不可超过 128 个字节，业务系统自行用来处理重复提交等。

返回值：

无返回值。

3. 根据 code 获取访问令牌（access_token）

获取 code 后，获取访问令牌，需构造如下的链接：
token_endpoint?
client_id=<CLIENT_ID>&client_secret=<CLIENT_SECRET>
&redirect_uri=<REDIRECT_URI>&code=<CODE>&grant_type= authorization_code

本链接 支持 POST 方法、basic 方法

参数	必须	说明
code	是	上一步骤中获取的 code
client_id	是	认证中心分配的 client_id，认证系统管理员确认
client_secret	是	认证中心分配的客户端秘钥，认证系统管理员确认
redirect_uri	是	授权后重定向的回调链接地址，请使用 urlencode 对链接进行处理，该地址需提前告知认证平台进行注册
grant_type	是	接口授权类型 此时固定为：authorization_code

返回值：JSON 对象


```
{

"access_token":"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJTeUhUemJ… …
… …
V9S0o5zvF16MYb1q4PAbMU3z5gZKdFlg3j_UQ9w6XY4o3kEnyTDVRY_6Q",
"expires_in":240,
"refresh_expires_in":14400,
"refresh_token":"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJTeUhUem
… …
… …
gFoYQ0OjuptVl8Hd3dXVIlJLXHFZl6EoqdHxw",
"token_type":"bearer",
"not-before-policy":1559227768,
"session_state":"d3ced2a7-bb7b-466e-946b-9c2e32842858"
}
```

4. 根据 access_token 获取账号详细信息（用户信息）

获取 access_token 后，获取用户信息，需构造如下的链接：
userinfo_endpoint?
code=<CODE>& access_token =<ACCESS_TOKEN>

本链接 支持 POST 方法,返回有权限查阅的用户信息

参数	必须	说明
code	否	response 中获取的 code(code 每次交互均会发生变化)
access_token	是	上一步骤中获取的 access_token

返回值：JSON 对象（认证平台，为了符合中国人阅读习惯，暂时将名和姓进行了倒置）

```
{
"sub":"8da87599-ef8a-46ff-8b92-1c9daf4e59c8",
"name":"姓 名字",

"preferred_username":"90029999",
"given_name":"姓",
"family_name":"名字"

}
```

5. 根据应用地址，注销当前登录用户的 session
end_session_endpoint?
&redirect_uri=<REDIRECT_URI>

本链接 支持 GET 方法

参数	必须	说明
redirect_uri	是	授权后重定向的回调链接地址，请使用 urlencode 对链接进行处理，该地址需提前告知认证平台进行注册

9.2.4 出错处理

认证过程前后，包括在业务系统鉴权时，可能会出现各个环节的信息处理错误，为了方便审计，要求将全部出错信息返回至信息认证平台错误处理页面。

页面地址要实现配置化处理，生产环境的统一错误地址为：

https://id.shenhua.cc/error/index.html

页面地址要实现配置化处理，开发环境的统一错误地址为：

https://devid.shenhua.cc/error/index.html

错误代码、错误信息两个参数请参照附录4，以GET方式返回

附录 1 基于 Oauth2.0 的应用系统的认证模块实现逻辑示例

登录逻辑示例：

```
package cn.xpie.idm.sh4a.sso;
import java.io.IOException;
import java.net.URLEncoder;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.http.Header;
import org.apache.http.client.fluent.Request;
import org.apache.http.client.fluent.Response;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.json.JSONObject;

public class Oauth2SampleServlet extends HttpServlet {

    private static final long serialVersionUID = 7687835458220948236L;
    private static final String KEY_SSO_USERNAME = "ssousername";

    private String ssoUrl;
    private String clientId;
    private String clientSecret;

    private String authUrl;
    private String tokenUrl;
    private String userinfoUrl;
    private String ssoCallbackUrl;
    @Override
    public void init(ServletConfig config) throws ServletException {
        // 认证服务地址,平台统一提供
        ssoUrl = config.getInitParameter("ssoUrl");
        // 认证服务为每一个集成客户端分配的一个clientId,需向平台申请
        clientId = config.getInitParameter("clientId");
        // 认证服务为每一个集成客户端分配的一个clientSecret,需向平台申请
        clientSecret = config.getInitParameter("clientSecret");
```

```
        // 构造认证服务API的最终地址
        authUrl = ssoUrl + "/auth";
        tokenUrl = ssoUrl + "/token";
        userinfoUrl = ssoUrl + "/userinfo";
        // 回调地址,由应用自行定义后向平台备案.支持*通配符,兼容动态地址.
        ssoCallbackUrl = "http://localhost:8080/oauth-api-sample/ssologin";
        // 也可以考虑认证后,再自行调转到指定业务页面
        // ssoCallbackUrl = "http://localhost:8080/oauth-api-
sample/ssologin?path=/workerlist/EA3223FBA";
        super.init(config);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        this.doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        // 获取code,固定参数
        String code = request.getParameter("code");
        // 以下loginName传参为示例逻辑,实际业务中请勿参考.
        String loginName = request.getParameter("loginName");
        // 构建http客户端
        CloseableHttpClient httpclient = HttpClients.createDefault();
        // 检查本应用是否已经登录过,从服务器取session判断,示例代码,请根据业务实际调整.
        HttpSession session = request.getSession();
        String ssousername = (String) session.getAttribute(KEY_SSO_USERNAME);
        // 如果本地无用户会话,则进入SSO认证逻辑
        if (ssousername == null || ssousername.equals("")) {
            try {
                // 尝试读取code,如无code,跳转到认证平台进行用户登录验证
                if (code != null) {
                    //code不为空,代表浏览器端用户已经登录,第三方应用服务端开始认证
                    //根据sso分配的clientId、clientSecret以及获取到的临时票据，发送获取
accessToken的请求
                    System.out.println(code);
                    String url = tokenUrl;
                    String param = "grant_type=authorization_code" + "&client_id="
```

```
+ clientId + "&client_secret="
                            + clientSecret + "&code=" + code + "&redirect_uri="
                            + URLEncoder.encode(ssoCallbackUrl, "UTF-8");
                    // 发送token获取请求,返回值为json对象,值长度不固定.
                    JSONObject result = new JSONObject(HttpRequest.sendPost(url,
param));
                    String accessToken = result.getString("access_token");
                    System.out.println("accessToken: " + accessToken);

                    // 发送userinfo获取请求，获取登录用户属性,返回值为json对象.
                    Response userinfoResponse = Request.Post(userinfoUrl)
                            .addHeader("Authorization", "bearer " +
accessToken).execute();

                    String userinfoJson =
userinfoResponse.returnContent().asString();
                    System.out.println(userinfoJson);
                    JSONObject userinfo = new JSONObject(userinfoJson);
                    // 取当前登录用户ID(8位工号,非拼音账号)
                    request.getSession().setAttribute(KEY_SSO_USERNAME,
userinfo.getString("preferred_username"));
                    // 重定向到业务主页面
                    response.sendRedirect("index.jsp");

                } else {
                    //未认证的状态，重定向到认证中心进行集中认证
                    String loginurl = authUrl;

                    String para = "response_type=code&client_id=" + clientId +
"&redirect_uri="
                            + URLEncoder.encode(ssoCallbackUrl, "UTF-8");
                    response.sendRedirect(loginurl + "?" + para);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

注：所有代码均为示例，用以说明接口的调用过程，强烈不建议在生产环境直接套用，避免未知风险。非JAVA系统，请使用http协议自行调用对应访问控制接口

Web.xml样例：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    version="3.0">

    <module-name>oauth-api-sample</module-name>

    <servlet>
        <description></description>
        <display-name>ssologin</display-name>
        <servlet-name>ssologin</servlet-name>
        <servlet-class>cn.xpie.idm.sh4a.sso.Oauth2SampleServlet</servlet-class>
        <init-param>
            <param-name>ssoUrl</param-name>
            <param-
value>http://devid.shenhua.cc/auth/realms/sh4a/protocol/openid-connect</param-
value>
        </init-param>
        <init-param>
            <param-name>clientId</param-name>
            <param-value>JTtest</param-value>
        </init-param>
        <init-param>
            <param-name>clientSecret</param-name>
            <param-value>68f9f290-bf94-efg7-abc4-cec278b6b50a</param-value>
        </init-param>

    </servlet>
    <servlet-mapping>
        <servlet-name>ssologin</servlet-name>
        <url-pattern>/ssologin</url-pattern>
    </servlet-mapping>

</web-app>
```

附录 2　身份供应接口标准文件与返回标识对照

接口标准开发wsdl文件：

ProvisioningWSService.wsdl

为了减少错误，强烈建议使用本文件直接反向生成服务端。

返回标识列表：

方法名称	错误标识（errcode）	错误信息(msg)
addAccount	0	保存用户成功
addAccount	userexception-001	用户已经存在
addAccount	userexception-002	保存用户出现错误
modifyAccount	userexception-010	用户不存在
modifyAccount	0	修改用户成功
modifyAccount	userexception-003	修改用户信息出现错误
deleteAccount	0	删除用户成功
deleteAccount	userexception-004	删除用户出错
suspendAccount	0	暂停帐户成功
suspendAccount	userexception-005	暂停帐户出现错误
restoreAccount	0	恢复帐户成功
restoreAccount	userexception-006	恢复帐户出现错误
searchAccount	0	查询账户成功
searchAccount	userexception-007	查询账户出现错误
不做具体实现的接口	userexception-008	业务系统未实现该接口

附录 3　访问控制接口统一报错信息对照

错误序号	错误标识（errcode）	错误信息(msg)
1	LoginErr-001	您没有权限登录该系统
2	LoginErr-002	系统内部无此用户
3	LoginErr-003	用户已被禁用
4	LoginErr-004	无法获取登录用户
5	LoginErr-005	用户来源非法
6	LoginErr-006	提交数据异常
7	LoginErr-007	认证服务无法访问
8	LoginErr-008	自定义错误文本

注:参数以get方式传参到出错界面