JavaChain平台CA节点子系统

服务器\_用户操作手册

2018-07-14

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V3.0.0.0 | 张翀 |  | 2018.5.28 | 初始版本 |
| V3.0.0.1 | 张翀 |  | 2018.6.21 | 修改初始化操作说明 |
| V3.0.0.2 | 张翀 |  | 2018.7.14 | 新增数据库说明 |
| V3.0.0.3 | 张翀 |  | 2018.7.30 | 新增更换CA说明  新增注册用户数据库说明  删除错误代码 |
| V3.0.0.5 | 张翀 |  | 2018.9.10 | 将路径名等修改为符合linux的描述 |
|  |  |  |  |  |

目 录

[**1** **项目描述** 5](#_Toc524447485)

[1.1 文档目标 5](#_Toc524447486)

[1.2 读者对象 5](#_Toc524447487)

[**2** **环境需求** 5](#_Toc524447488)

[2.1 部署需求 5](#_Toc524447489)

[2.2 证书要求 5](#_Toc524447490)

[2.2.1 CA服务器签名证书 5](#_Toc524447491)

[2.2.2 CA签证的身份证书 6](#_Toc524447492)

[2.2.3 RA通信证书配置 6](#_Toc524447493)

[**3** **使用说明** 6](#_Toc524447494)

[3.1 Getcainfo服务使用 6](#_Toc524447495)

[3.2 Enroll服务使用 7](#_Toc524447496)

[3.2.1 Enrollment身份证书管理 7](#_Toc524447497)

[3.3 Reenroll服务使用 7](#_Toc524447498)

[3.3.1 Reenrollment身份证书管理 7](#_Toc524447499)

[3.4 Register服务使用 8](#_Toc524447500)

[3.4.1 Register用户信息管理 9](#_Toc524447501)

[3.5 Revoke服务使用 10](#_Toc524447502)

[3.6 更换CA 10](#_Toc524447503)

[**4** **HTTP接口定义** 10](#_Toc524447504)

[4.1 接口概要 10](#_Toc524447505)

[4.1.1 接口协议版本 11](#_Toc524447506)

[4.2 获取CA信息接口（cainfo） 11](#_Toc524447507)

[4.2.1 接口地址 11](#_Toc524447508)

[4.2.2 HTTP请求方式 11](#_Toc524447509)

[4.2.3 同步异步 11](#_Toc524447510)

[4.2.4 请求参数 11](#_Toc524447511)

[4.2.5 返回参数 11](#_Toc524447512)

[4.3 签发证书接口（enroll） 12](#_Toc524447513)

[4.3.1 接口地址 12](#_Toc524447514)

[4.3.2 HTTP请求方式 12](#_Toc524447515)

[4.3.3 同步异步 12](#_Toc524447516)

[4.3.4 请求参数 12](#_Toc524447517)

[4.3.5 返回参数 13](#_Toc524447518)

[4.4 重新签发证书接口（reenroll） 13](#_Toc524447519)

[4.4.1 接口地址 13](#_Toc524447520)

[4.4.2 HTTP请求方式 13](#_Toc524447521)

[4.4.3 同步异步 13](#_Toc524447522)

[4.4.4 请求参数 13](#_Toc524447523)

[4.4.5 返回参数 14](#_Toc524447524)

[4.5 注册用户实体身份接口（register） 14](#_Toc524447525)

[4.5.1 接口地址 14](#_Toc524447526)

[4.5.2 HTTP请求方式 14](#_Toc524447527)

[4.5.3 同步异步 14](#_Toc524447528)

[4.5.4 请求参数 15](#_Toc524447529)

[4.5.5 返回参数 15](#_Toc524447530)

[4.6 吊销证书接口（revoke） 16](#_Toc524447531)

[4.6.1 接口地址 16](#_Toc524447532)

[4.6.2 HTTP请求方式 16](#_Toc524447533)

[4.6.3 同步异步 16](#_Toc524447534)

[4.6.4 请求参数 16](#_Toc524447535)

[4.6.5 返回参数 17](#_Toc524447536)

[4.7 批量查询证书（tcert） 17](#_Toc524447537)

[**5** **HTTP请求示例** 17](#_Toc524447538)

[**6** **数据库使用说明** 18](#_Toc524447539)

# **项目描述**

该项目的主要内容是为JavaChain平台提供CA服务器，主要功能包括：申请下载，重新申请并下载以及吊销SM2 国密算法签名的证书，管理用户信息和权限，管理CA信息。

## 文档目标

CA子系统服务器系统使用说明书。

## 读者对象

供BCIA javachain的平台开发者、应用开发者及应用最终用户参考使用。

# **环境需求**

## 部署需求

JAVA环境需要Oracle JDK 1.8以上，具体部署步骤参阅《JavaChain平台CA节点子系统服务器\_部署手册》。

## 证书要求

### CA服务器签名证书

必须要求是使用 SM2 国密算法签名的证书，来自于自建RA所连接的CA的签名证书或者证书链文件。必须事先将该证书放置于指定CA名称的目录下（/工作目录/CA\_Name/），便于getcainfo命令下载以及enroll命令需要。

### CA签证的身份证书

必须要求是使用 SM2 国密算法签名的证书。

### RA通信证书配置

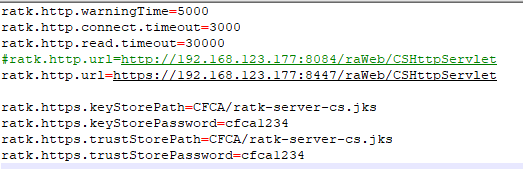
在config目录下配置ratk.properties文件。其中指定连接RA的双向SSL通信证书：

ratk.https.keyStorePath=CFCA/ratk-server-cs.jks

ratk.https.keyStorePassword=cfca1234

ratk.https.trustStorePath=CFCA/ratk-server-cs.jks

ratk.https.trustStorePassword=cfca1234



# **使用说明**

## Getcainfo服务使用

用户通过命令行客户端或者Restful接口进行访问，返回给用户指定CA的p7b证书链文件(是二进制编码,没有经过Base64编码的)以及CA的相关信息，比如名称，版本等。

注意，使用时首先必须将ca的P7B证书链文件(是二进制编码,没有经过Base64编码的)放入指定CA名称目录下的，要保证该目录下有证书文件。

## Enroll服务使用

用户通过命令行客户端或者Restful接口进行访问，返回给用户身份证书(ECert)，之后用户除了GetCaInfo 操作的所有操作，都会使用身份证书(ECert)进行签名。同时，服务器会绑定用户与身份证书，用于后续操作签名的验签。

注意，使用时首先必须将ca的证书和证书链文件放入指定CA名称目录下的，要保证该目录下有证书文件和证书链文件。

### Enrollment身份证书管理

在用户调用enroll 命令对应的Restful接口后，服务器会将生成的身份证书（ECert）与调用者的身份进行绑定，绑定的方式是：身份证书会以证书的 SubjectName作为文件名进行存储，存储的目录在/工作目录/ <对应CA 名称>/目录下，之后用户的其他操作都会采用身份证书签名来生成 token，token会放入报文的头部，token 的结构是B64SubjectNameAndBody.B64Signature，其中就有证书的 SubjectName以告知服务器用哪一个公钥证书来验签。

## Reenroll服务使用

用户通过命令行客户端或者Restful接口进行访问，假如你的证书期满终止了，可以重新enroll一个。成功则返回给用户新的身份证书(ECert)，之后用户除了GetCaInfo 操作的所有操作，都会使用新的身份证书(ECert)进行签名。同时，服务器会绑定用户与身份证书，用于后续操作签名的验签。

注意，使用时首先必须将ca的证书和证书链文件放入指定CA名称目录下的，要保证该目录下有证书文件和证书链文件。

### Reenrollment身份证书管理

在用户调用reenroll 命令或者reenroll对应的Restful接口后，服务器会将生成的身份证书（ECert）与调用者的身份进行绑定，绑定的方式是：身份证书会以证书的 SubjectName作为文件名进行存储，存储的目录在/工作目录/ <对应CA 名称>/目录下，之后用户的其他操作都会采用身份证书签名来生成 token，token会放入报文的头部，token 的结构是B64SubjectNameAndBody.B64Signature，其中就有证书的 SubjectName以告知服务器用哪一个公钥证书来验签。

## Register服务使用

该服务提供用户的注册功能，成功注册则返回该用户的服务器加密密码。

执行注册请求的身份必须是已经执行过enroll过的，并且还必须具有正在注册的身份类型的适当权限(该版本权限管理暂不支持)。

权限管理主要分三方面：

1. 需要支持注册服务商（即调用者）必须具有用逗号分隔值列表表示的“hf.Registrar.Roles”属性，其中一个值等于正在注册的标识类型; 例如，如果注册服务商的“hf.Registrar.Roles”属性的值为“peer，app，user”，则注册服务商可以注册peer，app和user类型的身份，但不能注册 order 身份。
2. 注册的发起者所具备的分组隶属关系必须是将注册的用户的分组隶属关系的前缀或者直接就是这个隶属关系，也就是说分组隶属关系必须具有继承性。例如，一个调用者的从属关系是“a.b”，那么他可以register一个拥有“a.b.c”的用户实例，但不能是“a.c”。
3. 如果满足以下所有条件，才可以注册用户属性：
   1. 只有当注册的发起者拥有该属性并且它是hf.Registrar.Attributes属性值的其中之一时，注册服务商才能注册具有前缀“hf.”的 CA保留属性。 此外，如果属性是类型列表，则被注册的属性的值必须等于或者是注册发起者具有的值的子集。 如果该属性的类型为布尔型，则注册发起者只有在自己这个属性的值为‘true’的情况下才能注册该属性。
   2. 注册自定义属性（即名称不以'hf.'开头的任何属性）要求注册d发起者已经注册了“hf.Registar.Attributes”属性。 唯一支持的模式是末尾带有“\*”的字符串。 例如，“a.b. \*”是匹配以“a.b.”开头的所有属性名称的模式。 例如，如果注册发起者的hf.Registrar.Attributes = orgAdmin，则注册服务商可以添加或删除身份的唯一属性是'orgAdmin'属性。

### Register用户信息管理

会将注册的用户信息保留在数据库admin/t\_registryuser/表中，对应的存储的数据示例如下图3-5所示:

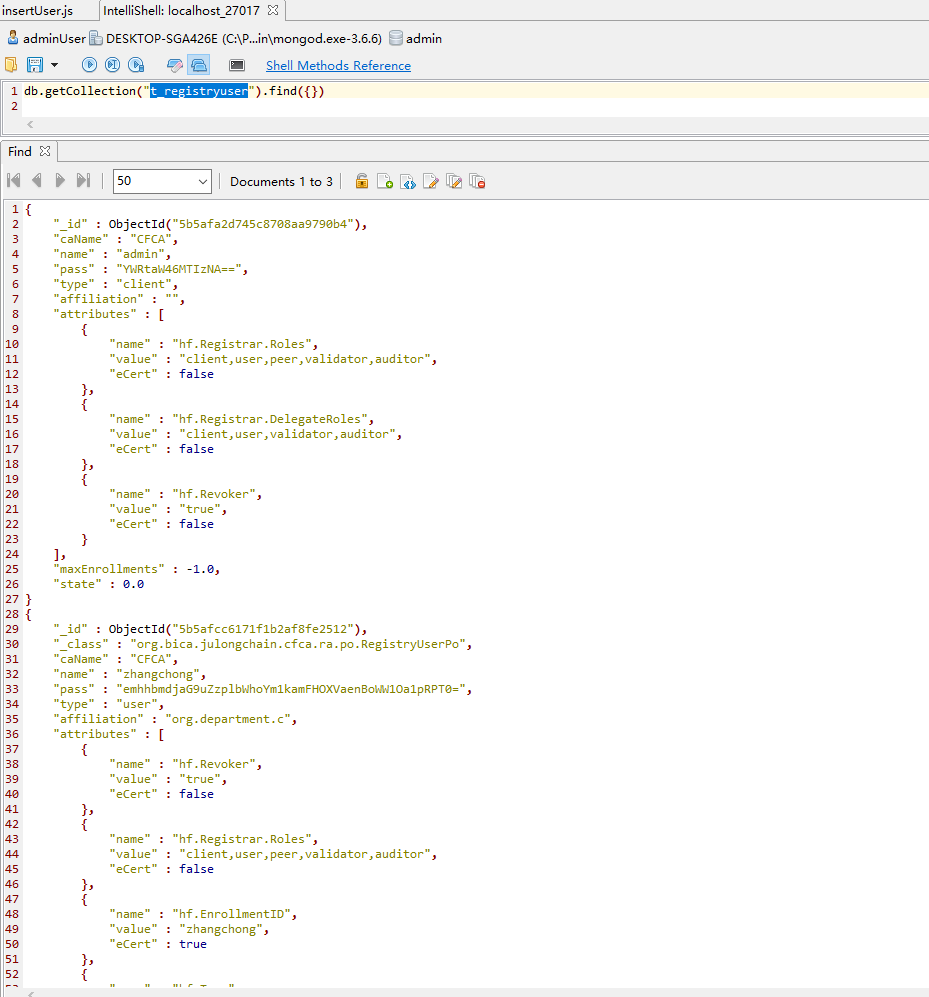


图3‑5

## Revoke服务使用

用户通过命令行客户端或者Restful接口进行访问，吊销制定序列号的证书。

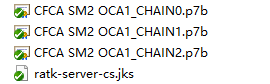
撤销一个用户会撤销他的所有证书，并阻止他再得到新的证书。撤销一个证书只是使一个证书无效。该版本只支持撤销一个证书,并没有管理用户名下申请的证书。

## 更换CA

首先修改config/ application-prod.properties中字段ca.name

ca.name=XXXXX

然后在jar包目录创建对应名称的目录，目录下需要放入证书链文件以及通信证书jks。



请增加admin数据库中表t\_registryuser 中的admin管理员用户名和密码，并且设置caName字段为对应的CA名称。

# **HTTP接口定义**

## 接口概要

CA给应用接入方提供的接口采用http协议。所有参数组装成JSON字符串，其中JSON的key严格遵循请求参数名称，将组装好的JSON放入http body域中用POST方式发送到请求地址。设置http请求的Header为“Content-Type=application/json;charset=UTF-8”。并需新增一个HEAD：“ca-protocol-version=0.0.1”，用来描述本协议内容的版本号。

### 接口协议版本

本接口协议版本号为：0.0.1。

## 获取CA信息接口（cainfo）

获取CA服务的基本信息。

### 接口地址

<http://IP:Port/cainfo>

### HTTP请求方式

Post 方式，请求参数置于报文body中。

### 同步异步

同步。

### 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| caname | ca名称 | String | "caname": "string"指定ca名称 | 不可空 | CFCA |

### 返回参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| Result | 处理结果 | JsonObject | {“caname”:ca名称(string), “cachain”:ca证书链文件的B64编码字符串} | 可空 |  |
| Success | 成功标识 | boolean | 布尔值，指示请求是否成功 | 不可空 | true |
| Errors | 处理错误 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx}  Success为false时,会有值 | 可空 | 一组错误消息（即代码和字符串消息） |
| Messages | 处理信息 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx} | 可空 | 一系列信息性消息（即代码和字符串消息） |

## 签发证书接口（enroll）

接收用户信息，进行用户登记，签发证书接口。

### 接口地址

<http://IP:Port/enroll>

### HTTP请求方式

Post 方式，请求参数置于报文body中。

### 同步异步

同步。

### 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| request | 请求详细信息 | String | 构建的csr的b64字符串 | 不可空 | 样例太长,自行生成 |
| profile | profile信息 | String | 签名模板，暂时不传 | 可空 |  |
| label | label信息 | String | HSM操作中使用的标签 | 不可空 | 暂时可以传空字符串 |
| caname | ca名称 | String | 请求指定的CA名称 | 不可空 | CFCA |

### 返回参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| Result | 处理结果 | String | Base64编码格式的身份证书, Success为true时,会有值 | 可空 | 样例太长,自行生成 |
| Success | 成功标识 | boolean | 布尔值，指示请求是否成功 | 不可空 | true |
| Errors | 处理错误 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx}  Success为false时,会有值 | 可空 | 一组错误消息（即代码和字符串消息） |
| Messages | 处理信息 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx} | 可空 | 一系列信息性消息（即代码和字符串消息） |

## 重新签发证书接口（reenroll）

用户重新登记和签发证书接口。

### 接口地址

<http://IP:Port/reenroll>

### HTTP请求方式

Post 方式，请求参数置于报文body中。

### 同步异步

同步。

### 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| request | 请求详细信息 | String | 构建的csr的b64字符串 | 不可空 | 样例太长,自行生成 |
| profile | profile信息 | String | 签名模板，暂时不传 | 可空 |  |
| label | label信息 | String | HSM操作中使用的标签 | 不可空 | 暂时可以传空字符串 |
| caname | ca名称 | String | 请求指定的CA名称 | 不可空 | CFCA |

### 返回参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| Result | 处理结果 | String | Base64编码格式的身份证书, Success为true时,会有值 | 可空 | 样例太长,自行生成 |
| Success | 成功标识 | boolean | 布尔值，指示请求是否成功 | 不可空 | true |
| Errors | 处理错误 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx}  Success为false时,会有值 | 可空 | 一组错误消息（即代码和字符串消息） |
| Messages | 处理信息 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx} | 可空 | 一系列信息性消息（即代码和字符串消息） |

## 注册用户实体身份接口（register）

接收用户信息，进行新用户注册接口。

### 接口地址

<http://IP:Port/register>

### HTTP请求方式

Post 方式，请求参数置于报文body中。

### 同步异步

同步。

### 请求参数

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | | |
| id | | id | String | 用户id | 不可空 | test |
| type | | 类型 | String | 类型 | 不可空 | user |
| secret | | 密码 | String | 密码 | 不可空 | xxxxxx |
| max\_enrollments | | 数量 | Integer | 最大证书数量 | 不可空 | 4 |
| affiliation\_path | | 路径 | String | 路径 | 不可空 | org.department |
| attrs | | 注册属性 | Json | "attrs": [  {  "name": "string",  "value": "string"  }  ],用数组传入多个属性值 | 不可空 |  |
| caname | | ca名称 | String | ca名称 | 不可空 | CFCA |

### 返回参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| Result | 处理结果 | JsonObject | 新注册身份的base64编码注册密码。 | 可空 |  |
| Success | 成功标识 | boolean | 布尔值，指示请求是否成功 | 不可空 | true |
| Errors | 处理错误 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx}  Success为false时,会有值 | 可空 | 一组错误消息（即代码和字符串消息） |
| Messages | 处理信息 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx} | 可空 | 一系列信息性消息（即代码和字符串消息） |

## 吊销证书接口（revoke）

吊销制定序列号的证书。

### 接口地址

http://IP:Port/revoke

### HTTP请求方式

Post 方式，请求参数置于报文body中。

### 同步异步

同步。

### 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| id | Id | String | 用户名 | 不可空 | test |
| aki | Aki | String | 要撤销的证书的授权密钥标识符（Authority Key Identifier） | 不可空 | “” |
| serial | Serial | String | 证书序列号 | 不可空 | 1032137068 |
| reason | Reason | String | 吊销原因 | 不可空 | expire |
| caname | Caname | String | Ca名称 | 不可空 | CFCA |

### 返回参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可空** | **样例** |
| **业务参数** | | | | | |
| Result | 处理结果 | String | 吊销相关信息 | 可空 |  |
| Success | 成功标识 | boolean | 布尔值，指示请求是否成功 | 不可空 | true |
| Errors | 处理错误 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx}  Success为false时,会有值 | 可空 | 一组错误消息（即代码和字符串消息） |
| Messages | 处理信息 | array | 每个元素是一个对象{ code:xxxxx,message:xxxx} | 可空 | 一系列信息性消息（即代码和字符串消息） |

## 批量查询证书（tcert）

批量查询证书接口。该版本暂不支持

# **HTTP请求示例**

POST http://<host:post>/reenroll/ HTTP/1.1

**Host**: ca.bcia.net.cn

**Accept**: application/json

**Accept-Encoding**: gzip,deflate,sdch

**Accept-Language**: en-US,en;q=0.8,fa;q=0.6,sv;q=0.4

**Cache-Control**: no-cache

**Connection**: keep-alive

**User-Agent**: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36

**Content-Length**: 120

**Content-Type**: application/json

{

"request": "requestExample",

"profile": "profileExample",

"label": "labelExample",

"caname": "canameExample"

}

# **数据库使用说明**

使用mongodb://adminUser:adminPass@localhost:27017/admin 这个url进行连接后,可查看admin数据库中表t\_enrollcert内容，其中使用caname-dn作为主键，每一条数据内容如下图：

