

数据上传、查询测试文档

xcredit wip

说明

请先完成系统的部署、注册、证书生成和上传，再执行以下测试。

测试的目的是验证前置机和中心的联通性。

测试分为数据上传和查询两部分。必须先完成数据上传才能执行查询，这系统为了鼓励共享、保护共享数据机构的利益所设置的规则。

测试结束后请勿删除测试数据，平台将不定期向前置机获取测试数据，以对前置机运行状况加以评估。

上传数据测试

测试数据简介

系统准备了三条数据（另行上传到群文件），对应目前系统中的三种数据类型，为了跟真实数据有区分，这里采用了不同于真实数据的 `dataType`：

- * 9000: 对应系统中的个人借贷。
- * 9001: 对应系统中的企业借贷。
- * 9002: 对应系统中的投资信息。

数据导出为文件 `sample_data.json` 和 `sample_data.sql` 脚本。

上传数据方式简介

前置机现支持三种数据上传方式。

1. 不通过前置机，直接将数据导入 `tb_generic_data`，之后访问前置机接口 `baseUrl/data/dbimport` 激活导入任务。
2. 通过接口，将数据导入。需要写程序，将字段序列化成 json 格式的数据，然后将数据 POST 至前置机接口 `baseUrl/data/uploadApi`。
3. 通过数据文件，将数据导入。将数据按上述格式导出为文件，登录前置机后台界面，访问“数据管理-上传数据”将数据上传。

其中，2、3 需要的数据都为 json 数组（可用紧凑格式），并不适合大批量上传。

文件 `sample_data.json` 的方式适合 2、3 两种上传，脚本 `sample_data.sql` 适合 1。

查询数据测试

平台搭建了一个模拟机构，性能不高，用于提供数据查询测试。

测试数据简介

系统准备了三组查询参数。

- * indexDigest=e09d4a9f6d60678c1bc78d151b0fa863ad8866d1a728aa9f2c71cc3ec3541f79;indexType=1000
- * indexDigest=d1ebd3fc7f0f6f84d838e0fb73fd2262fc07126750a129cd7a1faa05d9a06f39;indexType=1001
- * indexDigest=1c126681512bf17952abab084a2a79c01afe193da960240f8a0f8361727e9569;indexType=1000

使用以上参数，按以下方法，来获取数据

获得数据，从业务上来说，分为两步，首先获得数据索引，然后过滤出感兴趣的索引，分别发出查询请求，获得数据。

1. 输入 `indexDigest` `indexType`，获得该身份所有人的全部索引列表
2. 根据索引列表，可获得某条数据的 `dataType` 以及 `cert`，这样加上第一步的两个数据，得到四个查询“坐标”，可获得此条索引所代表的信息。

示例步骤如下：

（以下数据和返回结果仅为提供演示，需要将参数（比如 `indexType` 和 `dataType` 替换成上面提到的值）

1. 获得索引

选择测试数据中的一条，取其 `indexDigest` `indexType`，比如

`indexType=1000`

`indexDigest=e09d4a9f6d60678c1bc78d151b0fa863ad8866d1a728aa9f2c71cc3ec3541f79`

2. GET 请求 `http://前置机地址/index/get` 接口，将参数以 query string 形式拼到 url 上，如：

`http://前置机地址/index/get?`

`indexType=1000&indexDigest=e09d4a9f6d60678c1bc78d151b0fa863ad8866d1a728aa9f2c71cc3ec3541f79`

3. 返回数据是一个 json 字符串

```
1.  {
2.    "result": 0,
3.    "message": null,
4.    "data": [
5.      {
6.        "address": "http://bjwd.bjp2p.com.cn/index/getData",
7.        "data": {
8.          "indexType": 1000,
9.          "dataDigest": "/dDBXQVULVDd35n3Y0bMvO+qQHprI4n35DXSetztIyQ=",
10.         "dataType": 9000,
11.         "cert":
"MIIDDCCAFsGAWIBAgIERW/+2TANBgkqhkiG9w0BAQsFADA/MRYwFAYDVQQDDA14Y3JlZG10IGZyb250MRgwFgYDVQKDA9PUkc9UExBQ0VIT0xERVlxCzAJBGNVBAZTAkNOMB4XDTE2MDIyODAwMTIxMloXDTE2MDAwMTIxMlowPzEWMBGA1UEAwNeGNyZWVpdCBmcm9udDEYMBYGA1UECgwPT1JHIFBMRQUNFSE9MRVSMQSwCQYDVQQGEwJDTjCCASIwDQYJKoIhvcNAQEBBQADggEPADCCAQoCggEBBAJPC9MiDOEDxulU7UhpDO6UtO52LzuJpXHRripq/X5/V1XD5chKme7mFbNFeOYOXNRWS+liLveM/vIxcOy25n30kPJRkKwzRnt2gV3v7c52C1cBTLM195+2hkmtP9zfV0x1uGK+RNuOVf9sKoGf/HRiO16IzlKzeG+trdocXKI8WYfYHumAMD4ZQdETQTimXcySZJ27OVSAuTRTnSgQ5qyTfX5NhXjJQ77jGwEfpUPkCRG9W4xa+qbuoqN5YfQUxutRTYvcKdfPwb77iq5Op3BSetLMruUdd5QrmyBw+HB2zEexFBf4MyW8jKPiX7q/hUcInRAoDqBLndhUfwuECAwEAAAMQMA4wDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAW3R2BTzeOzR6KfRug+cA2LbeIVp+U1sYB6RfOm8U7Oq8TFX2rcc2ZofJ8GwjXerhx34x9snuqgKp+cule8WlfYWpq2d1/zHP1xm3ZOLVYN/y+kPFIRvZP62LR
```

```

mpJwX6hyiCY3vx/zjj3nuqX0f4lcdlw1ZKse6j/nz4f3XFNGMGxjiw7q8+oD8MNlRTkjx6lSbo4o95BkUUfB5i2cA
58cSzwAh4JuHoKv2FflEYukQAEGU1Ph1FliQKoQUqJ/kXrYlI55jBAPVcwYm1Ll9X6XgSxZ5npiYz+L3LAr9GLpdQxw
sNRWLQvpkX7fTj8M8Qki2DXTUvPj/U84QRA==" ,
12.         "price": 0,
13.         "indexSignature":
"DQOzttnECb7v2e+dU/ySdsHq6QxgYW+iGZevVkcOQ4acJx3A0scq2sNw04fYCBTo4TzgyM/2oeB/EzmQgmbeFpSSa5
1IK7kMeQjKTXi/jzIeGnILHSvc1Plc+h3ztd0fq5uIFB+ipTuJdhajaUhNZ2s3nUxM1OgF8liYHejDhiSEsjPGkk1dc
kF1QHWLoZa6YuVVIWh+47QOD+mV8Z4YCKUqXuisjRXYEpU8/9DAH+mxZqaJb9RBptGcM5QZ82d2sx6dJmLbpIwTrdY+
xZ7BfZSkDYhB5FoZ4HCKHpjYROYaSXGCI7VrZYbvYUCWn7NjLSODU9dhkOhNkaRP0hQ==" ,
14.         "indexDigest":
"e09d4a9f6d60678c1bc78d151b0fa863ad8866d1a728aa9f2c71cc3ec3541f79",
15.         "lastUpdateTime": 1484631086000,
16.         "endDate": 1480521600000,
17.         "beginDate": 1448899200000,
18.         "indexSaltDigest": ""
19.     }
20. }
21. ]
22. }

```

4. 根据返回的信息，提取 `dataType` 和 `cert`，联合第一步的两个参数，以 `multipart/form-data` 的 `MediaType` POST 到前置机 `baseUrl/index/getData` 接口。

5. 解析返回数据

Have fun

附录

以下内容与测试本身无关。

了解某些状态字段，有助于 debug。

上传状态

激活上传任务后，系统将分批、分步执行上传任务，更新 `tb_generic_data` `tb_index` `tb_cert` 等表。通过查询 `tb_generic_data` `tb_index` 的某些字段，可以了解上传进行情况。

`tb_generic_data` 表

`is_indexed` 字段有如下状态:

- 0: 初始状态
- 1: 已经生成索引
- 2: 旧数据，说明其被同一个人（或机构）的相同数据类型的数据所取代
- 3: 错误，参考 `note` 字段的错误信息
- 4: 已验证，但未生成索引，说明数据字段完整，通过初步校验
- 5: 验证错误

其中 0 为初始状态，1 为目标状态

tb_index 表

is_synced 字段有如下状态:

- 0: 初始状态
- 1: 已上传到平台，可查询
- 2: 内容已更新，尚未签名
- 3: 内容已签名，待上传
- 4: 错误

其中 0 为初始状态，1 为目标状态

当索引变为 1 时，说明己方数据已经上传成功，可以进行查询测试了。

数据更新逻辑

系统对数据有几个限定条件：

1. 相同 indexDigest + 相同 dataType + 相同 indexType 的索引，只能有一条。换句话说，每个主体在某种数据类型中，只能存在一条索引。
2. 因为 1，所以多条相同 indexDigest + 相同 dataType + 相同 indexType 的业务数据，后面的只能更新原有索引，不能新增一条索引。
3. 索引首次创建时，会自动选择一个已经上传的证书与之关联，并且之后不能更换证书。