INFORMATION SYSTEMS

IPI WORKING GROUP

IPI Pocket Edition API

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Version</th>
<th>Comments / Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>02.03.2023</td>
<td>Peter Klauser</td>
<td>0.1</td>
<td>Initial Draft.</td>
</tr>
<tr>
<td>27.03.2023</td>
<td>Peter Klauser</td>
<td>0.2</td>
<td>Bulk data calls – via FTP</td>
</tr>
<tr>
<td>12.05.2023</td>
<td>Peter Klauser</td>
<td>0.3</td>
<td>Added getDomain calls</td>
</tr>
</tbody>
</table>
Table of contents :

IPI QUERY SYSTEM - POCKET EDITION API ......................................................................................... 3

API Operations ........................................................................................................................................ 3
  API Operation: searchIP .......................................................................................................................... 5
  Operation: getIPNameDetail .................................................................................................................... 7
  Operation: getSocieties .......................................................................................................................... 8
  Operation: getCreationClasses ............................................................................................................ 9
  Operation: getRoles ............................................................................................................................... 9
  Operation: getRights ............................................................................................................................ 10
  Operation: getTerritories ..................................................................................................................... 10
  Operation: getTerritoryHierarchies .................................................................................................... 11
  Operation: resolveTerritories ............................................................................................................ 12

Security and Authentication .................................................................................................................... 13

Service Levels ....................................................................................................................................... 15

Application Architectures ...................................................................................................................... 15

Requesting Access ................................................................................................................................. 16

Endpoints .................................................................................................................................................. 16

Information about Building Client Software ...................................................................................... 17
  Postman ................................................................................................................................................ 17
  Java 11 .................................................................................................................................................. 17

IPI QUERY SYSTEM – BULK DATA ...................................................................................................... 18

SFTP data bundles ................................................................................................................................. 18

Agreement Data Format ....................................................................................................................... 18

Name Data Format ............................................................................................................................... 19
IPI QuerySystem - PocketEdition API

The IPI QuerySystem - PocketEdition API is a “read only” interface to search and retrieve data from IPI. The

No changes can be made in the IPI system by using the APIs.

Publishers can use the PocketEdition API as part of the PocketEdition subscription.

The PocketEdition API does not contain the relationship between names of an interested party. No name type information ( PA, PP, .. ) is displayed nor is the IpBaseNumber made available.

The search and lookup data ( not bulk ) provided by the PocketEdition APIs is based on a realtime replica of the IPI master system. The real-time replica may “lag” behind the master system at times where bulk operations are taking place, but is eventually consistent with the master. The lag is in the order of minutes and not days.

The PocketEdition provides bulk datasets which are made available via (S)FTP. Publishers requesting access to the bulk datasets are requested to sign a data protection agreement and incur a CHF 600.- annual service charge (subject to change).

API Operations

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchIP</td>
<td>Search for IP names given some combination of name, date of birth/death, society affiliations.</td>
</tr>
<tr>
<td>getIPNameDetail</td>
<td>Lookup of name and agreement data given an IpNameNr found via the search or known in advance.</td>
</tr>
<tr>
<td>getSocieties</td>
<td>Retrieve the list of currently defined society codes and the associated society names and country.</td>
</tr>
<tr>
<td>getTerritories</td>
<td>Territory codes and their names ( TISN, validFrom )</td>
</tr>
<tr>
<td>getTerritoryHierarchies</td>
<td>Territory hierarchy ( geographic / political / economic country groups )</td>
</tr>
<tr>
<td>resolveTerritories</td>
<td>Mechanism to resolve include/exclude lists into explicit lists of countries.</td>
</tr>
<tr>
<td>getCreationClasses</td>
<td>A listing of all creation classe codes and their names.</td>
</tr>
<tr>
<td>getRoles</td>
<td>A listing of all roles.</td>
</tr>
<tr>
<td>getRights</td>
<td>A listing of all rights.</td>
</tr>
</tbody>
</table>
**API Operation: searchIP**

Example Request

```
{
  "ipNameNumber": "461733556",
  "ipName": "+Winehouse +Amy",
  "type": "N",
  "dateOfBirthOrFoundation": "2001",
  "dateOfDeathOrDissolution": "2010",
  "agreementCcRoRi": "+MW +LY",
  "agreementSociety": "052",
  "start": 0,
  "rows": 20
}
```

Search parameters for finding an Interested Party.

<table>
<thead>
<tr>
<th><strong>Property</strong></th>
<th><strong>Datatype</strong></th>
<th><strong>Example</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ipNameNumber</td>
<td>string</td>
<td>461733556</td>
<td>Interested Party Name Number. If an Interested Party Name Number is provided, the other parameters are ignored.</td>
</tr>
<tr>
<td>ipName</td>
<td>string</td>
<td>+Winehouse +Amy</td>
<td>Name of the Interested Party.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>N</td>
<td>Type of the Interested Party.</td>
</tr>
<tr>
<td>dateOfBirthOrFoundation</td>
<td>string</td>
<td>2001-10</td>
<td>Birth/foundation of the Interested Party.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2001-10-19</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2014 TO 2016]</td>
<td></td>
</tr>
<tr>
<td>dateOfDeathOrDissolution</td>
<td>string</td>
<td>2010-12-01</td>
<td>Death/dissolution of the Interested Party.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2010-12-19</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2014 TO 2016-12-01]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[* TO 2014-12-01]</td>
<td></td>
</tr>
<tr>
<td>agreementCcRoRi</td>
<td>string</td>
<td>+MW +LY</td>
<td>Creation class, role and right of the agreement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MW-EM-PR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>+OB</td>
<td></td>
</tr>
<tr>
<td>agreementSociety</td>
<td>string</td>
<td>052</td>
<td>Society code of an agreement.</td>
</tr>
<tr>
<td>start</td>
<td>integer ($int32)</td>
<td>0</td>
<td>The start row of the found records..</td>
</tr>
<tr>
<td>rows</td>
<td>integer ($int32)</td>
<td>20</td>
<td>Number of rows to be returned by the search.</td>
</tr>
</tbody>
</table>

Information about the search syntax:

The parameters ipName, type, dateOfBirthOrFoundation, dateOfDeathOrDissolution, agreementCcRoRi and agreementSociety are used in logical conjunction (AND) if provided together.

String syntax (Matching Rules):
Wildcard '*': The search 'tes*' would match 'test', 'testing', and 'tester'.
Wildcard '?': The search string 'te?t' would match both 'test' and 'text'.
Boolean Operator '+': The search string '+Winehouse' matches strings which include 'Winehouse'.
Boolean Operator '-': The search string '+Winehouse -Amy' matches strings with 'Winehouse' but not 'Amy'.

Date syntax:

Single date examples:
2000: The entire year 2000
2000-11: The entire month of November, 2000
2000-11-05: The Fifth of November

Date Range examples:
[2014 TO 2014-12-01]: From the start of 2014 till the end of the first day of December.
[* TO 2014-12-01]: From the earliest representable time till the end of the day on 2014-12-01.

Example Response

```
{
    "start": 0,
    "numFound": 1,
    "errorCode": 0,
    "errorMessage": "string",
    "ipSearchDetailList": [
        {
            "ipNameNumber": 123456789,
            "ipName": "WINEHOUSE AMY JADE",
            "type": "N",
            "sex": "F",
            "dateOfBirthOrFoundation": "1983-09-14",
            "dateOfDeathOrDissolution": "2011-07-23",
            "currentState": 1,
            "agreementSummary": [
                "MW-LY*-ER*-052"
            ]
        }
    ]
}
```

Result of an 'Interested Party' search

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
<th>Example(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>integer ($int64)</td>
<td>0</td>
<td>Offset of the search result (used for paging).</td>
</tr>
<tr>
<td>numFound</td>
<td>integer ($int64)</td>
<td>1</td>
<td>Number of results found. (total)</td>
</tr>
<tr>
<td>errorCode</td>
<td>integer ($int32)</td>
<td>0</td>
<td>0 – success, else some code.</td>
</tr>
<tr>
<td>errorMessage</td>
<td>string</td>
<td>null</td>
<td>null if no error, else descriptive text.</td>
</tr>
<tr>
<td>ipSearchDetailList</td>
<td>IPSearchDetail</td>
<td>see below</td>
<td>One or more search results representing a summary description of the IpName.</td>
</tr>
</tbody>
</table>
Individual IPSearchDetail record description:

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
<th>Example(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipNameNumber</td>
<td>Integer</td>
<td>461733556</td>
<td>Interested Party Name Number.</td>
</tr>
<tr>
<td>ipName</td>
<td>string</td>
<td>SMITH JOHN</td>
<td>Name of the Interested Party.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>N</td>
<td>Type of the Interested Party.</td>
</tr>
<tr>
<td>sex</td>
<td>string</td>
<td>M, F, null</td>
<td>Sex of the Interested Party.</td>
</tr>
<tr>
<td>dateOfBirthOrFoundation</td>
<td>string</td>
<td>1983-09-14</td>
<td>Birth/foundation of the Interested Party in ISO format. Can be 'YYYY', 'YYYY-MM' or 'YYYY-MM-DD'.</td>
</tr>
<tr>
<td>dateOfDeathOrDissolution</td>
<td>string</td>
<td>2011-07-23</td>
<td>Death/dissolution of the Interested Party in ISO format. Can be 'YYYY', 'YYYY-MM-MM-DD'.</td>
</tr>
<tr>
<td>currentState</td>
<td>integer</td>
<td>1, 2, 3, 4</td>
<td>Current state of the Interested Party.</td>
</tr>
<tr>
<td>agreementSummary</td>
<td>List - string</td>
<td>[ &quot;MW-LY*-ER*-052&quot; ]</td>
<td>Summary of the agreements. Lists all CC/RO/RI/SOC for each CreationClass (only the first combination).</td>
</tr>
</tbody>
</table>

Open Questions:

Should we provide back the placeOfBirth, stateOfBirth, countryOfBirth, and Nationalities in the search result overview per IP to help in identification? These can be looked up later with the getIpNameDetail.

There is no search provision for “FirstName” vs “Name” – by convention we state the Name, then FirstName as the “ipName”.

Phonetic search is not forseen, nor is the search

**Operation: getIPNameDetail**

Example Request: `/getIPNameDetail?ipNameNumber=<IpNameNr>`

Parameters for retrieving IP details.

```plaintext
<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipNameNumber</td>
<td>string</td>
<td>461733556</td>
<td>Interested Party Name Number.</td>
</tr>
</tbody>
</table>
```

Example Response:
All agreement data is returned past and present.

**Operation: getSocieties**

Example Request: /getSocieties

Example Response:

```json
[
  {
    "code": "001",
    "name": "ACUM",
    "country": "ISRAEL"
  },
  {
    "code": "002",
    "name": "ADDAF",
    "country": "BRAZIL"
  },
  {
    "code": "003",
    "name": "AEPI",
    "country": "GREECE"
  }
]
Operation: getCreationClasses

Example Request: /geCreationClasses

Example Response:

```json
[
  {
    "code": "AC",
    "description": "ARCHITECTURAL WORK"
  },
  {
    "code": "AD",
    "description": "DOCUMENTARY AUDIO-VISUAL WORK"
  },
  {
    "code": "AF",
    "description": "FICTION AUDIO-VISUAL WORK"
  },
  {
    "code": "AM",
    "description": "MUSIC VIDEOS"
  }
]
```

Operation: getRoles

Example Request: /geRoles

Example Response:

```json
[
  {
    "code": "AG",
    "description": "GRAPHIC DESIGNER"
  },
  {
    "code": "AP",
    "description": "ANALYST/PROGRAMMER"
  },
  {
    "code": "AS",
    "description": "AUTHOR OF SCREENPLAY/AUTHOR OF DIALOGUE"
  }
]
```
Operation: getRights

Example Request: /getRights

Example Response:

```
[{
  "code": "BT",
  "description": "BLANK TAPE REMUNERATION"
},
{
  "code": "DB",
  "description": "DATABASE RIGHT"
},
{
  "code": "ER",
  "description": "EDUCATIONAL RIGHT"
},
{
  "code": "EX",
  "description": "EXHIBITION RIGHT"
},
{
  "code": "MA",
  "description": "MECHANICAL RIGHT RADIO"
}
]
```

Operation: getTerritories

Example Request: /getTerritories

Example Response:

```
[
  {
    "tisn": 4,
    "tisa": "AF",
    "tisan": "AFGHANISTAN",
    "tisnValidFrom": "1000-01-01",
    "tisnValidTo": "3999-12-31",
    "nameValidFrom": "1000-01-01",
    "nameValidTo": "3999-12-31"
  },
  {
    "tisn": 8,
    "tisa": "AL",
    "tisan": "ALBANIA",
    "tisnValidFrom": "1000-01-01",
    "tisnValidTo": "3999-12-31",
    "nameValidFrom": "1000-01-01",
    "nameValidTo": "3999-12-31"
  }
]
```
"nameValidTo": "3999-12-31"
},
{
  "tisn": 12,
  "tisa": "DZ",
  "tisan": "ALGERIA",
  "tisnValidFrom": "1000-01-01",
  "tisnValidTo": "3999-12-31",
  "nameValidFrom": "1000-01-01",
  "nameValidTo": "3999-12-31"
},
{
  "tisn": 2134,
  "tisa": "2WI",
  "tisan": "WEST INDIES",
  "tisnValidFrom": "1000-01-01",
  "tisnValidTo": "3999-12-31",
  "nameValidFrom": "1000-01-01",
  "nameValidTo": "3999-12-31"
},
{
  "tisn": 2136,
  "tisa": "2WL",
  "tisan": "WORLD",
  "tisnValidFrom": "1000-01-01",
  "tisnValidTo": "3999-12-31",
  "nameValidFrom": "1000-01-01",
  "nameValidTo": "3999-12-31"
}
]

Operation: getTerritoryHierarchies

Example Request: /getTerritoryHierarchies

Example Response:

[ {
  "parentTisn": 2100,
  "childTisn": 12,
  "parentTisnValidFrom": "1000-01-01",
  "childTisnValidFrom": "1000-01-01",
  "hierarchyValidFrom": "1000-01-01",
  "hierarchyValidTo": "3999-12-31"
},
  {
  "parentTisn": 2100,
  "childTisn": 24,
  "parentTisnValidFrom": "1000-01-01",
  "childTisnValidFrom": "1000-01-01",
  "hierarchyValidFrom": "1000-01-01",
  "hierarchyValidTo": "3999-12-31"
},
  {
  "parentTisn": 2100,
  "childTisn": 72,
  "parentTisnValidFrom": "1000-01-01",
  "childTisnValidFrom": "1000-01-01",
  "hierarchyValidFrom": "1000-01-01",
  "hierarchyValidTo": "3999-12-31"
} ]
Operation: resolveTerritories

Example Request

```json
{
  "validAt": "2004-01-01",
  "agreementTerritoryList": [
    {
      "flag": "I",
      "tisn": 2136
    },
    {
      "flag": "E",
      "tisn": 2120
    }
  ]
}
```

Example Response:

```json
{
  "territoryList": [
    {
      "tisn": 4,
      "tisa": "AF",
      "tisan": "AFGHANISTAN",
      "tisnValidFrom": "1000-01-01",
      "tisnValidTo": "3999-12-31",
      "nameValidFrom": "1000-01-01",
      "nameValidTo": "3999-12-31"
    },
    {
      "tisn": 12,
      "tisa": "DZ",
      "tisan": "ALGERIA",
      "tisnValidFrom": "1000-01-01",
      "tisnValidTo": "3999-12-31",
      "nameValidFrom": "1000-01-01",
      "nameValidTo": "3999-12-31"
    },
    {
      "tisn": 887,
      "tisa": "YE",
      "tisan": "YEMEN",
      "tisnValidFrom": "1990-05-22",
      "tisnValidTo": "3999-12-31",
      "nameValidFrom": "1990-05-22",
      "nameValidTo": "3999-12-31"
    }
  ]
}
```
Security and Authentication

In order to use the PocketEdition API’s, the client must first “log in” to an authentication service and retrieve a JWT access token (see [https://jwt.io/](https://jwt.io/) ). This access token is then used by the client with each subsequent HTTP call to the PocketEdition API functions. The token is valid for a certain period of time, and once the token is no longer valid, the client must log in again at the authentication service and retrieve a new token. The duration of validity of the token is provided in the response to the client from the authentication service (expires_in seconds). Each API request must include the token in the “Authorization” HTTP Request header, as a “Bearer” style authorization. See example below.

AUTHENTICATION REQUEST

REQUEST HEADERS
Content-Type: application/x-www-form-urlencoded
Content-Length: 144

POST https://auth.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token

REQUEST BODY

```
client_id=ipiqs-api&client_secret=<cs>&username=<username>&password=<pwd>
&scope=openid&grant_type=password
```

RESPONSE HEADERS

Content-Type: application/json
Content-Length: 2843

RESPONSE BODY

```
{"access_token":"eyJhbGciOiJ…jJyV2Ds7m1x36jGh2esdKY31czFXW9w","expires_in":300,
"refresh_expires_in":1800,
"refresh_token":"eyJhbGciOiJ…zRlYi1iOWM3LTQyYmQt…DgifQ.9G0PqT-K2Pt","token_type":"Bearer",
"id_token":"eyJhbG…J5ZE95N0hqTHJON3FGQdzzoUznIS_t4YdHveQgQeA",
```
"not-before-policy":1675178258,
"session_state":"ace22da5-8720-4114-bf18-fc32fa1af348",
"scope":"openid"
}

AUTHENTICATED REQUEST

REQUEST HEADERS
Content-Type: application/json
Accept: application/json
Authorization: Bearer eyJpeyJjyV2Ds7m1x36jGh2esdKY31czFw9w
Host: ipiqsentw.ipisystem.org
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 59

POST https://ipiqsentw.ipisystem.org/services/api/public/limited/v1/search

REQUEST BODY
{
  "ipName": "John Smith",
  "start": 0,
  "rows": 20
}

RESPONSE HEADERS
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Keep-Alive: timeout=60
Connection: keep-alive
Strict-Transport-Security: max-age=31536000
content-length: 293

RESPONSE BODY
{
  "ipNameNumber": 123456789,
  "name": "SMITH",
  "firstName": "JOHN",
  "type": "N",
  "sex": "F",
  "dateOfBirthOrFoundation": "1983-09-14",
  "agreementList": [
    {
      "validFrom": "2004-07-01",
      "validTo": "9999-12-31",
      "society": "052",
      "creationClass": "MN",
      "roles": ["LY", "MC"],
      "rights": ["ER", "MP", "OB", "OD", "PC", "PR", "PT", "RB", "RT", "TB", "TO", "TP", "TV"],
      "share": 100,
      "dateOfSignature": "2004-09-03"
    }
  ],
  "share": 100
}
Service Levels

The service is provided with “Best Effort” support.

Currently there is no foreseen “Rate Limitation” concerning the number of API calls which can be performed. However, if excessive usage is determined to cause a detriment of the service to others, then the access to the system can be restricted indefinitely, by the revocation of access entirely.

Application Architectures

The API and security mechanism allow the API to be used as part of a “Single Page” applications which run mostly in a user’s browser.
Requesting Access

Publishers can request access to the PocketEdition APIs by emailing ipi@suisa.ch. Please provide your full contact information including your (main) IPI name numbers.

Endpoints

<table>
<thead>
<tr>
<th>ENV</th>
<th>Authentication API</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>Swagger Documentation</td>
</tr>
<tr>
<td></td>
<td>Authentication API</td>
</tr>
<tr>
<td></td>
<td><a href="https://auth-test.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token">https://auth-test.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token</a></td>
</tr>
<tr>
<td></td>
<td>Service API</td>
</tr>
<tr>
<td></td>
<td><a href="https://ipiqstest.ipisystem.org/services/api/public/limited/v1/">https://ipiqstest.ipisystem.org/services/api/public/limited/v1/</a></td>
</tr>
<tr>
<td>PREP</td>
<td>Swagger Documentation</td>
</tr>
<tr>
<td></td>
<td>Authentication API</td>
</tr>
<tr>
<td></td>
<td><a href="https://auth-prep.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token">https://auth-prep.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token</a></td>
</tr>
<tr>
<td></td>
<td>Service API</td>
</tr>
<tr>
<td></td>
<td><a href="https://ipiqsprep.ipisystem.org/services/api/public/limited/v1/">https://ipiqsprep.ipisystem.org/services/api/public/limited/v1/</a></td>
</tr>
<tr>
<td>PROD</td>
<td>Swagger Documentation</td>
</tr>
<tr>
<td></td>
<td>Authentication API</td>
</tr>
<tr>
<td></td>
<td><a href="https://auth.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token">https://auth.ipisystem.org/auth/realms/ipi/protocol/openid-connect/token</a></td>
</tr>
<tr>
<td></td>
<td>Service API</td>
</tr>
<tr>
<td></td>
<td><a href="https://ipiqs.ipisystem.org/services/api/public/limited/v1/">https://ipiqs.ipisystem.org/services/api/public/limited/v1/</a></td>
</tr>
</tbody>
</table>
**TLS / Cipher support**

We do not support TLSv1.0. TLSv1.1 is “deprecated” and currently clients should use a TLSv1.2 with a cipher supported from the list below, preferably from the “Perfect Forward Secrecy” set.

<table>
<thead>
<tr>
<th>TLSv1.1</th>
<th>TLS_RSA_WITH_AES_128_CBC_SHA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_3DES_EDE_CBC_SHA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TLSv1.2</th>
<th>( Perfect Forward Secrecy – ciphers )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td></td>
</tr>
</tbody>
</table>

( the following are considered WEAK )

| TLS_RSA_WITH_AES_128_GCM_SHA256 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 |

If operating with JAVA – you should install the unlimited strength cryptography extensions. See [https://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html](https://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html)

**Information about Building Client Software**

**Postman**

Describe how to

**Java 11**

Describe how to
IPI Query System – Bulk Data

Data bundles are prepared weekly and provided via SFTP.

The files are organized in a directory structure starting with the IPI_PocketEdition folder.

SFTP data bundles

<table>
<thead>
<tr>
<th>Directory</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPI_PocketEdition</td>
<td>Empty except for directories {names, agreements}</td>
</tr>
<tr>
<td>/names</td>
<td>Empty except for directories { full, delta }</td>
</tr>
<tr>
<td>/full</td>
<td>names-full-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/delta</td>
<td>names-delta-YYYYMMDD-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/agreements</td>
<td>Empty except for directories { full, delta }</td>
</tr>
<tr>
<td>/full</td>
<td>Empty except for directories { all, music, nonmusic }</td>
</tr>
<tr>
<td>/music</td>
<td>agreements-full—all-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/nonmusic</td>
<td>agreements-full—all-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/full</td>
<td>Empty except for directories { all, music, nonmusic }</td>
</tr>
<tr>
<td>/all</td>
<td>agreements-delta—all-YYYYMMDD-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/music</td>
<td>agreements-delta—music-YYYYMMDD-YYYYMMDD.tsv.gz</td>
</tr>
<tr>
<td>/nonmusic</td>
<td>agreements-delta—nonmusic-YYYYMMDD-YYYYMMDD.tsv.gz</td>
</tr>
</tbody>
</table>

The contents of the gzip file are streamed to the client.

Agreement Data Format

Agreement data is a TAB separated line for each IpNameNr, SocietyCode, CreationClass, ValidFrom, ValidTo, SignatureDate, Share, Territories (I/E csv), Roles (csv), Rights (csv)

```
471719049 010 MW 2005-03-04 2005-03-01 100.00 I:2136; LY,MC
471719147 010 MW 2005-03-04 2005-03-01 100.00 I:2136; LY,MC
471719245 010 MW 2005-03-04 2005-03-01 100.00 I:2136; LY,MC
471719343 010 MW 2005-03-04 2005-03-01 100.00 I:2136; LY,MC
471719441 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
471719539 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
471719637 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
471719735 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
471719833 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
471719931 010 MW 2005-03-04 2005-03-01 100.00 I:2136; EM
```
Name Data Format

The name data is a TAB separated line for each name – with fields for IpNameNr, Name, FirstName, NameUsages, TypeOfEntity,

<table>
<thead>
<tr>
<th>Name</th>
<th>FirstName</th>
<th>TypeOfEntity</th>
<th>Gender</th>
<th>BirthDate</th>
<th>City</th>
<th>Zip</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOU GIRONES PASCUAL</td>
<td></td>
<td>AV-AS;AV-DS;AV-RE;CW-CG;DM-LY;DM-MC;DW-PW;DW-WP;LW-WP;MW-LY;MW-MC</td>
<td>N M</td>
<td>1978.12.17</td>
<td>BARCELONA</td>
<td>724</td>
<td>724</td>
</tr>
<tr>
<td>SANCHEZ GONZALEZ JAVIER</td>
<td></td>
<td>AV-AS;AV-DS;AV-RE;CW-CG;DM-LY;DM-MC;DW-PW;DW-WP;MW-LY;MW-MC</td>
<td>N M</td>
<td>1978.12.11</td>
<td>SALAMANCA</td>
<td>724</td>
<td>724</td>
</tr>
<tr>
<td>GALLEGO CARRO CARLOS</td>
<td></td>
<td>AV-AS;AV-DS;AV-RE;CW-CG;DM-LY;DM-MC;DW-PW;DW-WP;LW-WP;MW-LY;MW-MC</td>
<td>1</td>
<td>1977.06.02</td>
<td>MADRID</td>
<td>724</td>
<td>724</td>
</tr>
<tr>
<td>PINO LASSO DE LA VEGA ANGEL</td>
<td></td>
<td>AV-AS;AV-DS;AV-RE;CW-CG;DM-LY;DM-MC;DW-PW;DW-WP;MW-LY;MW-MC</td>
<td>1</td>
<td>1977.09.22</td>
<td>MADRID</td>
<td>724</td>
<td>724</td>
</tr>
<tr>
<td>PERRET ANNE MARIE MARINA RAYMONDE</td>
<td></td>
<td>MW-LY;MW-MC</td>
<td>N</td>
<td>1950.07.19</td>
<td>ARGENTEUIL</td>
<td>95</td>
<td>250</td>
</tr>
<tr>
<td>PERRET ARGENTEUIL</td>
<td></td>
<td>MW-LY;MW-MC</td>
<td>N</td>
<td>1950.07.19</td>
<td>ARGENTEUIL</td>
<td>95</td>
<td>250</td>
</tr>
</tbody>
</table>