

**Integrating With Voyant**

*A Guide*

Updated: May 16, 2022 11:30 AM

**Table of Contents**

[**Introduction** 2](#_Toc103592310)

[Single Sign-On 3](#_Toc103592311)

[Enterprise User Provisioning 3](#_Toc103592312)

[Client Plan Data Integration and Migration 4](#_Toc103592313)

[Appendix A: Voyant REST API Service Specification 6](#_Toc103592314)

[Appendix B: Client Plan Data Model 10](#_Toc103592315)

[Appendix C: Voyant REST Services 15](#_Toc103592316)

[Appendix D: Legacy Plan Data Import 21](#_Toc103592317)

[Appendix E: Hierarchy and User Provisioning 23](#_Toc103592318)

[Appendix F: Launch in Context of a Client via Integration 24](#_Toc103592319)

[Appendix G: Details on Secure Connectivity 27](#_Toc103592320)

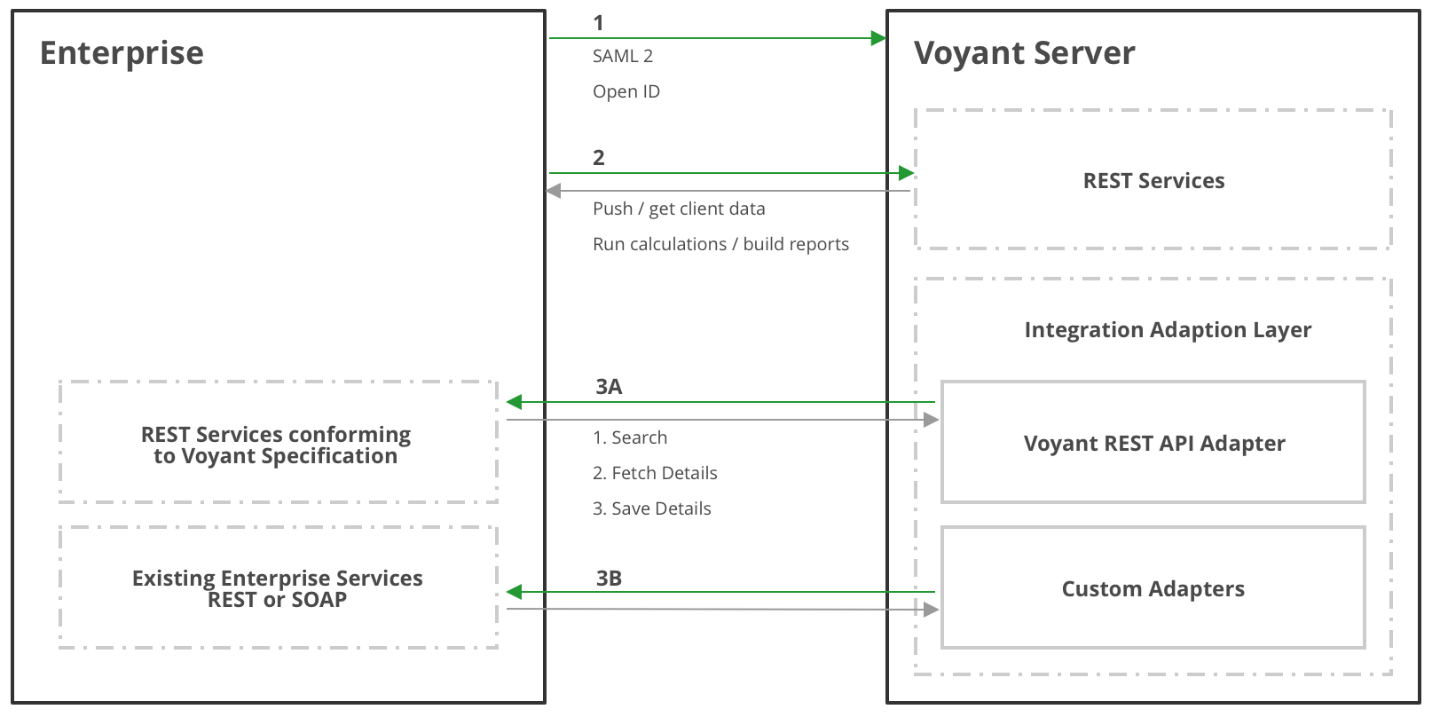
**Introduction**

Voyant supports integration with enterprise sites and partner application in numerous manners, including:

* Single sign-on (SSO)
* User provisioning
* Client plan data import/export
* Client reports
* Management Information

This document gives a high-level overview of the currently supported options for SSO and client data integration.

**Voyant Integration Overview**



1. Single Sign-On
2. Enterprise Initiated Integration (Push)
3. Voyant Initiated Integration (Pull)
   1. Rest services confirming to Voyant specification
   2. Existing enterprise services – REST or SOAP

Single Sign-On

Voyant supports the standards based SAML and Oauth/OpenId protocols, which are available to enterprise clients utilizing a siloed Voyant AWS environment. Additionally, custom authentication filters may be implemented for JWT (JSON Web Token) or other custom authentication approaches as required.

*SAML*

Voyant supports both IDP initiated and Service initiated flows with browser POST binding.

*OAuth/OpenID Connect*

Voyant supports OAuth based approaches with OpenId as a configurable option.

Enterprise User Provisioning

Voyant utilizes a hierarchical role and permission-based system to enable authorization or client data and features within Voyant. As such, Voyant requires a user’s “account” to be provisioned within the Voyant system with the appropriate roles. We provide service endpoints that allow enterprises to programmatically manage the hierarchy and user provisioning.

**Adviser & Admin User Hierarchy Management in Voyant**

Appendix E includes a sample of the group hierarchy file that can be uploaded to Voyant via the admin console or service.

In typical engagements, due to complexity this file is *not* created directly by the enterprise. Rather, Voyant creates the file through a bespoke automated processing step from enterprise provided data files.

The hierarchy structure (LOBs, groups, etc.) is not expected to change frequently and thus does not require a nightly automated update process.

**Adviser User Provisioning**

Adviser users can be incrementally added and assigned to their respective roles within the hierarchy by uploading a .csv file via the admin console or service endpoint. A sample file with documentation is attached. It is expected that CIBC would manage this process directly via the service endpoints or admin console.

Client Plan Data Integration and Migration

Voyant provides several options for initializing and updating client data within the system through service calls initiated by Voyant and/or partner systems.

1. **Voyant REST Service Spec** – Partners may implement services conforming to Voyant’s Integration Service specification, enabling seamless configuration within the Voyant system.

* User interaction initiated from with the Voyant AdviserGo application.
* Partners implement three simple REST services – search, get client details, and save client details (if client details entered in Voyant should be saved back to the partner system)
* Configurable authentication methods – Basic Auth or HTTP Session w/ proxied user credentials, OAuth (preferred), or custom authentication handlers (available to enterprise partners)

1. **Integration with existing partner services** – For partners with existing client data services, Voyant can develop custom adapters to orchestrate the integration of these existing services into the standard Voyant initiated user experience.

Partner services must generally provide the following:

* Search capability based on user entered data such as first/last name, DOB, province/state, etc.
* Services providing household data (i.e. financial data for all parties in a plan) are preferred. However, Voyant can provide orchestrate the inclusion of multiple clients’ data into a single plan.
* OAuth authentication is preferred, but custom authentication and authorization methods may be supported. Services implemented at the partner system MUST provide for authentication and authorization based on the current user.

1. **Voyant REST API** – For partner-initiated service requests, Voyant provides a full suite of REST APIs for everything from creating clients to updating client data, running simulations, and retrieve calculation/chart data.   
     
   There are two authentication options depending on the calling system’s “ownership” of the data within Voyant:

* **OAuth Client Credentials Grant flow** – For enterprises who have purchased Voyant licenses and control user access to Voyant, the client credentials grant method of authentication/authorization. This enables seamless access to the enterprise of all data in Voyant and all services to which the enterprise has subscribed.
* **OAuth Authorization code flow** – For partners who do not own the Voyant license of the mutual user, the authorization code flow method allows Voyant users to control the partner’s access to Voyant data and services.

1. **Launch in context** – Voyant provides a url endpoint for integration partners wishing to add a link in the partner application which will launch Adviser Go with the indicated client in context.

* If partner provided services are available as defined #1 or #2 above then Voyant will handle the requests to retrieve for the first time and/or update the client records with data provided by the partner.
* Alternatively, the partner can use the “load” REST endpoint to preload client plan data into Voyant prior to directing the user to the launch in context URL.

Please note: technical details for above integration options can be found in the included appendixes.

**Legacy Data Migration**

Voyant provides services Enterprises may also migrate client plan data from existing client data from legacy planning systems can be posted in bulk to the Voyant migration service for conversion.

Appendix A: Voyant REST API Service Specification

Supported Authentication Protocols:

* OAuth – Authorization code and Client Credentials Grant flows supported
* Proxied end user credentials – Basic Auth or HTTP session-based authentication
* Custom authentication handlers with header injected authentication keys/tokens

Client Search:

**Request:**

Method: GET

Path: https://[host and app context]/clientSearch?filter={search filter string}

With the exception of an “id” for each search result, the partner system may define the dataset to be displayed in the search screen by specifying the column names and returning the appropriate column data. The example below illustrates a response containing name, email address, and client acquisition data.

**Response Example:**

*<?xml version="1.0" encoding="utf-8"?>*

*<clientSearch>*

*<columns>*

*<column>Name</column>*

*<column>Email</column>*

*<column>Date Acquired<column>*

*</columns>*

*<searchResults>*

*<searchResult id="ABDC12345">*

*<column>Smith, Joe</column>*

*<column>*[*joesmith@gmail.com</column*](mailto:joesmith@gmail.com%3c/column)*>*

*<column>28/08/2010</column>*

*</searchResult>*

*<searchResult id="WXYZ54321">*

*<column>Johnson, Joe</column>*

*<column>*[*joejo@yahoo.com</column*](mailto:joejo@yahoo.com%3c/column)*></column>*

*<column>20/08/2010</column>*

*</searchResult>*

*</searchResults>*

*<clientSearch>*

Client Details

**Request:**

Method: GET

Path: https://[host and app context]/client/{clientReferenceID}

Where clientReferenceID = an id from a searchResult in the search response

**Response:**

XML client details conforming to Voyant dataIntegration.xsd.

**Client Details Save**

**Request:**

Method: POST

Path: https://[host and app context]/client/{clientReferenceID}

Posted message - XML client details conforming to Voyant dataIntegration.xsd.

**Response:**

The service should return a mapping of ids for any new items contained in the posted data message, which are processed and stored by the partner service.

<?xml version="1.0" encoding="utf-8"?>

<savedClient>

<partnerClientId>170e96c0-f133-e211-b14f-782bcb776b8c</partnerClientId>

<idMapList>

<idMapping>

<voyantId>eeeb6fe6c0a801733aa3defcc872a2a8</voyantId>

<partnerId>partnerXYZ:ghshkjashfkjhkjh87687678678hjk</partnerId>

</idMapping>

</idMapList>

</savedClient>

Error Responses

1. Authentication/invalid session errors would be returned as normal HTTP 401 “Unauthorized” errors.
2. All services may also return an error in the following style.

**Example:**

  <?xml version="1.0" encoding="utf-8"?>

  <error>

    <code>500</code>

    <message>Invalid client reference id</message>

  </error>

Appendix B: Client Plan Data Model

XSD Data Schema & Browsable HTML schema available at:

[**https://support.planwithvoyant.com/hc/en-us/articles/360020796592**](https://support.planwithvoyant.com/hc/en-us/articles/360020796592)

The following sample illustrates the minimal plan data required for typical plan elements (Ireland version shown)

*<?xml version="1.0" encoding="UTF-8" standalone="yes"?>*

*<client calcType="IE" planId="3aefc383c0a8002901c4800e08afd601" xmlns="urn:voyant:integration">*

*<people>*

*<primaryPerson id="3aefc393c0a800290171a67883c64955">*

*<firstName>Mr</firstName>*

*<lastName>Sample-IE</lastName>*

*<birthday>1971-01-01</birthday>*

*<gender>MALE</gender>*

*<personType>PRIMARY</personType>*

*</primaryPerson>*

*<spouse id="3aefc399c0a800290198e2a1ef4c1232">*

*<firstName>Mrs</firstName>*

*<lastName>Sample-IE</lastName>*

*<birthday>1974-01-01</birthday>*

*<gender>FEMALE</gender>*

*<personType>SPOUSE</personType>*

*</spouse>*

*<child id="3aefc39bc0a800290091139eadff93ad">*

*<firstName>Child1</firstName>*

*<lastName>Sample-IE</lastName>*

*<birthday>2003-01-01</birthday>*

*<gender>MALE</gender>*

*<personType>CHILD</personType>*

*</child>*

*<child id="3aefc39cc0a800290106a9c94bcf16c2">*

*<firstName>Child2</firstName>*

*<lastName>Sample-IE</lastName>*

*<birthday>2006-01-01</birthday>*

*<gender>MALE</gender>*

*<personType>CHILD</personType>*

*</child>*

*</people>*

*<events>*

*<event id="3aefc39bc0a8002901d614cf679f1378">*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*<timeType>AGE\_RELATIVE</timeType>*

*<ageOrYear>65</ageOrYear>*

*<type>RETIREMENT</type>*

*<icon>retirement</icon>*

*<name>Retirement</name>*

*</event>*

*<event id="3aefc39bc0a800290121653e4a26e6b9">*

*<owner>3aefc399c0a800290198e2a1ef4c1232</owner>*

*<timeType>AGE\_RELATIVE</timeType>*

*<ageOrYear>65</ageOrYear>*

*<type>RETIREMENT</type>*

*<icon>retirement</icon>*

*<name>Retirement</name>*

*</event>*

*</events>*

*<details>*

*<ieEmployment id="3aefcccdc0a8002901306e01af8cf50e">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<employmentSource>EMPLOYED</employmentSource>*

*<name>Jones &amp; Roberts</name>*

*<salary>75000</salary>*

*<isMonthly>false</isMonthly>*

*</ieEmployment>*

*<otherIncome id="3af16ed5c0a8002901f4f932056fc687">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<name>rental income</name>*

*<income>12000</income>*

*<isTaxable>true</isTaxable>*

*</otherIncome>*

*<ieSavings id="3af29659c0a8002901c19c35a1f2f034">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*<owner>3aefc399c0a800290198e2a1ef4c1232</owner>*

*</owners>*

*<name>Joint Savings</name>*

*<balance>25000</balance>*

*<growthRate>0.03</growthRate>*

*<growByAssetAllocation>false</growByAssetAllocation>*

*<type>SAVING</type>*

*<purchaseValue>0</purchaseValue>*

*<feeRate>0.005</feeRate>*

*<appliedEventList/>*

*</ieSavings>*

*<ieInvestment id="3af2fac0c0a800290135d5fe14eceee2">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<name>GIA</name>*

*<balance>120000</balance>*

*<growthRate>0.06</growthRate>*

*<growByAssetAllocation>false</growByAssetAllocation>*

*<type>PORTFOLIO</type>*

*<purchaseValue>86000</purchaseValue>*

*<feeRate>0.005</feeRate>*

*<assetAllocation>*

*<class name="Cash">1</class>*

*</assetAllocation>*

*<appliedEventList/>*

*</ieInvestment>*

*<ieMoneyPurchase id="3af350bac0a80029016f0d313ab866cf">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<name>Occupational Pension</name>*

*<balance>300000</balance>*

*<growthRate>0.06</growthRate>*

*<growByAssetAllocation>false</growByAssetAllocation>*

*<type>OCP</type>*

*<feeRate>0.005</feeRate>*

*<assetAllocation>*

*<class name="Cash">1</class>*

*</assetAllocation>*

*<appliedEventList/>*

*</ieMoneyPurchase>*

*<ieRetirementFund id="3af3b66cc0a8002901e457025c4f289c">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<name>ARF</name>*

*<balance>125000</balance>*

*<growthRate>0.06</growthRate>*

*<growByAssetAllocation>false</growByAssetAllocation>*

*<initialBalance>100000</initialBalance>*

*<feeRate>0.005</feeRate>*

*<isAMRF>false</isAMRF>*

*<assetAllocation>*

*<class name="Cash">1</class>*

*</assetAllocation>*

*<appliedEventList/>*

*</ieRetirementFund>*

*<ieFinalSalary id="3af4147fc0a8002900436a010e03e5ba">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<status>DEFERRED</status>*

*<name>Deferred Final Salary</name>*

*<amount>55000</amount>*

*<isMonthly>false</isMonthly>*

*<escalationType>NONE</escalationType>*

*<survivorPercent>0</survivorPercent>*

*<appliedEventList/>*

*</ieFinalSalary>*

*<ieAnnuity id="3af477b3c0a8002901ee301402edbd26">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<status>IN\_PAYMENT</status>*

*<name>Annuity</name>*

*<paymentAmount>12000</paymentAmount>*

*<isMonthly>false</isMonthly>*

*<lifetime>true</lifetime>*

*<qualified>true</qualified>*

*<capitalElement>0</capitalElement>*

*<isJoint>false</isJoint>*

*</ieAnnuity>*

*<property id="3af4b1f8c0a800290066bf40270583b7">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<type>PRIMARY\_RESIDENCE</type>*

*<name>House</name>*

*<marketValue>450000</marketValue>*

*<purchaseValue>200000</purchaseValue>*

*<newPurchase>false</newPurchase>*

*<ownershipType>JOINT\_TENANCY</ownershipType>*

*</property>*

*<debt id="3af4fb31c0a8002901d135be988b8986">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*</owners>*

*<name>Loan</name>*

*<balance>300000</balance>*

*<interestRate>0.05</interestRate>*

*<paymentAmount>24000</paymentAmount>*

*<isMonthly>false</isMonthly>*

*<interestOnly>false</interestOnly>*

*</debt>*

*<basicExpense id="3af5b7bcc0a8002901bb7dc8194e9af9">*

*<owners>*

*<owner>3aefc393c0a800290171a67883c64955</owner>*

*<owner>3aefc399c0a800290198e2a1ef4c1232</owner>*

*</owners>*

*<name>Core expense</name>*

*<group></group>*

*<amount>75000</amount>*

*<isMonthly>false</isMonthly>*

*</basicExpense>*

*</details>*

*</client>*

Appendix C: Voyant REST Services

The following endpoints are available for consumption by enterprise partners. Full documentation to be provided.

**User and client config services**

Return plan config details and default assumptions such as capital market assumptions for the current user or specified client record:

*/voyant/services/rest/user/config  
/voyant/services/rest/client/config*

**Client data services**

These are the primary endpoints to creating, fetching, and updating client plans:

*/voyant/services/rest/client/  
/voyant/services/rest/client/search  
/voyant/services/rest/client/copy/  
/voyant/services/rest/client/*

**Calc services**

Invoke plan calculations and simulations:

*/voyant/services/rest/calc/*

**Specialised services**

*/voyant/services/rest/client/collegeData/  
/voyant/services/rest/socialSecurity/calc/  
/voyant/services/rest/canadaPensionCalc/*

**Financial security info and allocation services**

*/voyant/services/rest/allocation/  
/voyant/services/rest/secInfo/save  
/voyant/services/rest/secInfo/search  
/voyant/services/rest/secInfo/price  
/voyant/services/rest/secInfo/allocation*

**PDF report building services**

*/voyant/services/rest/clientReport/buildReport/  
/voyant/services/rest/clientReport/status/  
/voyant/services/rest/clientReport/download/*

**Authenticating with OAuth**

Currently Voyant supports two different types of OAuth2 Flows:

Authorization Code:

Used when individual users in a system will be required to log in to Voyant to grant the integrator access to their data.

Client Credentials:

Used when an integrator has full access over a range of data in Voyant. This set of credentials may be used to pull data from Voyant without having to have individual users grant access to their data.

The following are the basic steps in order to integrate using OAuth2:

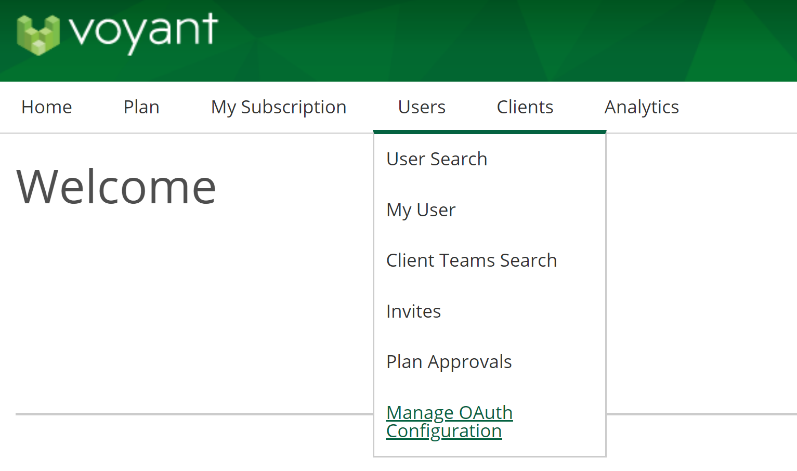
1. Register and configure the integrating application in the Voyant system
2. For users of the Authorization Code Flow, redirect the user to the Authorization Endpoint. The user will be required to log in to Voyant and approve the integration’s request for data access.
3. For users of the Authorization Code Flow, a code will be returned to the designated redirect url configured for the integration.
4. Using the returned code (Authorization Code Flow) or the corresponding credentials (Client Credential Flow) the integration will request an Access Token from the Token Endpoint.
5. Once a token is retrieved by the integration it may be used to access the protected resources in the Voyant system (Rest API).
6. Once a token expires it may be refresh automatically by integration without having prompt the user to reapprove the integration with Voyant. Once the refresh token expires, the user will be required to go back through the original flow to reapprove.

**Set up**

* In order to integrate with Voyant using OAuth2 the client application must first be registered in the Voyant System.
* An app that is registered with Voyant will have a unique OAuth client id and client secret generated. Additionally, for the Authorization Code Grant Flow, a redirect uri must be configured.
* A user may be elected as a ‘manager’ for a set of OAuth credentials. This user may log in to the Voyant admin console to change certain options on the configuration.

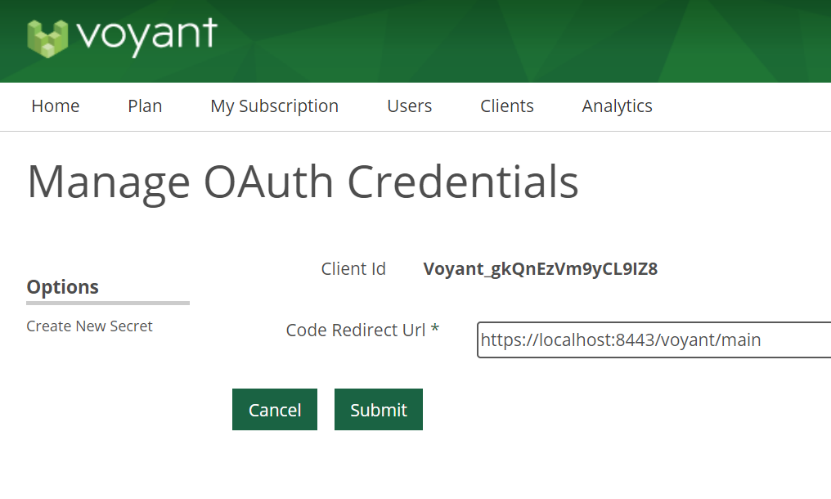
**Configuring OAuth Details**

The Credential Manager can access their OAuth Configuration through the Users -> Manage OAuth Configuration menu.



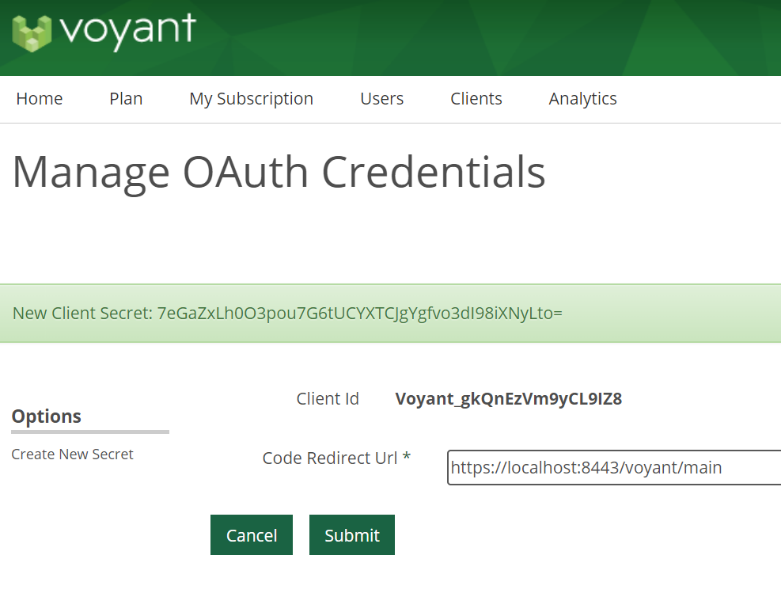
This page allows a user to:

* View the Client Id
* Set the desired redirect Uri for the credentials to return the code to.
* Generate a Client Secret.



Please note when generating a new Client Secret to copy down the value for use when accessing Voyant. The secret serves as the integrations password when accessing Voyant and should be securely stored and not shared.

This is the only time that the secret will appear in plain text. If the secret is forgotten or otherwise lost then a new secret may be generated at any time.



**Client Credential Configuration**

For enterprises using the Client Credential Flow. A user in the Voyant system will need to be associated with a given set of OAuth Credentials. This user will act as the Authorization source when using tokens returned for the credentials. This association can only be established by an administrator.

**Authorization**

For the Authorization Code Flow only.

The authorization endpoint is: https://{baseUrl}/voyant/main/oauth/authorize

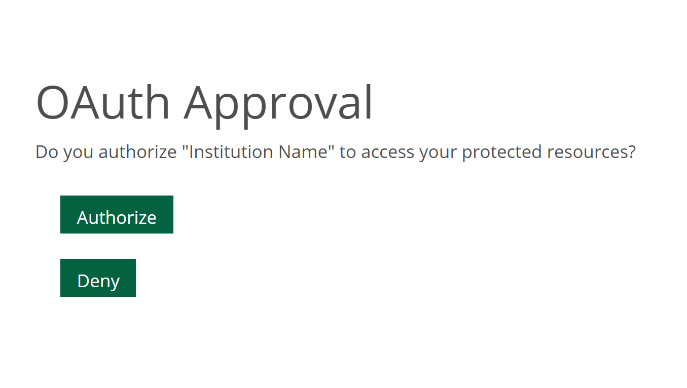
The query params to include are:

* client\_id – the client id configured in the credentials
* response\_type – code
* redirect\_uri – the redirect uri value configured in the credentials
* state – a nonce value which will be returned back to the integration uri. For security purposes.

Ex.

https://planwithvoyant.co.uk/voyant/main/oauth/authorize?response\_type=code&client\_id=Voyant\_gkQnEzVm9yCL9IZ8&redirect\_uri=https://institution.com/oauth/authorize&state=d43lkj2o

Upon redirect the user will be prompted to login, once they have authenticated in the Voyant system they will be required to approve the integration for data access:



After the user approves the integration, a code will be sent to the configured redirect uri ex:

https://institution.com/oauth/authorize?code=S9IsDK&state= d43lkj2o

**Token**

To receive the OAuth Access token the integration must POST to the token endpoint:

https://{baseUrl}/voyant/main/oauth/token

The integration must authenticate to this end point by sending the client id and client secret as a Basic encoded Authorization header value.

Required headers:

|  |  |
| --- | --- |
| Header | Value |
| Content-Type | application/x-www-form-urlencoded |
| Accept | application/json |
| Authorization | Basic {client\_id:client\_secret} |

The request body will depend on which OAuth Flow Type is being used.

Authorization Code:

|  |  |
| --- | --- |
| Param | Value |
| grant\_type | authorization\_code |
| code | Code sent to redirect uri in Authorization step |
| redirect\_uri | The redirect uri specified in the OAuth configuration |

Ex.

|  |  |
| --- | --- |
| grant\_type | authorization\_code |
| code | S9IsDK |
| redirect\_uri | https://institution.com/oauth/authorize |

Client Credentials:

|  |  |
| --- | --- |
| grant\_type | client\_credentials |

Sample Response:

{

"access\_token": "3724deb8-be8e-4404-b986-ce1a8a369b99",

"token\_type": "bearer",

"refresh\_token": "7226291a-c852-45fa-99b0-0f8eb0246dda",

"expires\_in": 3599,

"scope": "all"

}

**Refresh Token**

A token may be refresh for a certain period of time by calling the Token endpoint with grant type ‘refresh\_token’ and providing the corresponding refresh token found in the original response.

Ex.

|  |  |
| --- | --- |
| grant\_type | refresh\_token |
| refresh\_token | 7226291a-c852-45fa-99b0-0f8eb0246dda |

**Accessing Resources**

Once a valid access token is retrieved for a user it can be used in API requests to the Voyant system by providing the token in the http Authorization header.

Ex.

|  |  |
| --- | --- |
| Authorization | bearer 3724deb8-be8e-4404-b986-ce1a8a369b99 |

**OAuth Client**

Voyant can also act as the OAuth client to access data from integration partners for use within the Voyant system.

For the Authorization Code Flow Voyant’s code redirect uri is: https://{baseUrl}/voyant/main/oauth/authorize/callback

Ex. https://planwithvoyant.co.uk/voyant/main/oauth/authorize/callback

Appendix D: Legacy Plan Data Import

**Overview**

* REST API – Provides endpoint for enterprises to submit legacy client data and connect with associated advisers
* Multi-part post w/ authentication by token
* Voyant client conversion occurs interactively when first accessed by a user

Request:



Response Codes:

|  |  |  |
| --- | --- | --- |
| **Status** | **Message** | **Reason** |
| 401 (Unauthorized) | User not found for provided userId. | The userId provided in the request does not exist in the voyant system. |
| 401 (Unauthorized) | User not authorized to view provided userId. | The userId provided in the request exists, but the credentials provided to the API are not authorized to view the userId. |
| 400 (Bad Request) | Attachment Content-Id=userInfo not found in request. | The multi-part request does not contain the userInfo section. |
| 400 (Bad Request) | Attachment Content-Id=userData not found in request. | The multi-part request does not contain the userData section. |
| 400 (Bad Request) | Provided data source is not a supported external data source. | The request contains a data source that is not recognized (i.e. not NAVIPLAN) |
| 400 (Bad Request) | Request cannot be processed for the following reason: | The request can be parsed correctly but contains some invalid data condition which prevents a minimal plan from being created (i.e. no people in the plan, XML provided is not the expected format, etc.) |
| 500 (Server Error) |  | General uncaught exceptions, most should come back as 400 (Bad Request) |

Appendix E: Hierarchy and User Provisioning



Appendix F: Launch in Context of a Client via Integration

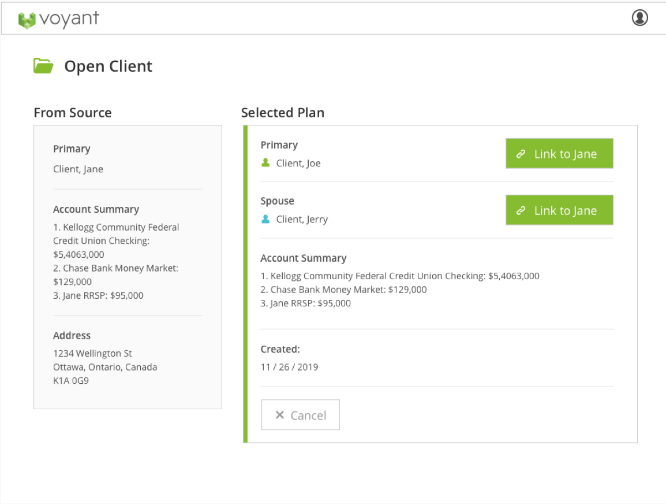
**Partners wanting a more seamless experience can use Voyant’s Launch-in-Context flows to transition directly from the partner’s system into AdviserGo.**

Depending on how your data is organized, Voyant has multiple integrations journeys. Supported options include launching directly as family unit, or an individual person from another system. Additionally, partners who do not have publicly accessible web services, but can map their data to Voyant’s data schema, can launch directly from a file.

* Person ID
  + Integration with data for multiple people in a household / family unit.
  + Multiple Integration IDs are linked to a Voyant client
* Household / Family Unit ID
  + Integration Source has data for the entire household.
  + One integration ID, one Voyant client
* Import from a client File
  + Create / Update a client directly from a file source

**Person ID Integration flow**

Voyant Plan can be created directly from integration’s person details. Existing Voyant plans can be updated by connecting people already in the plan.



**URL & Required Parameters**

Person ID must be provided

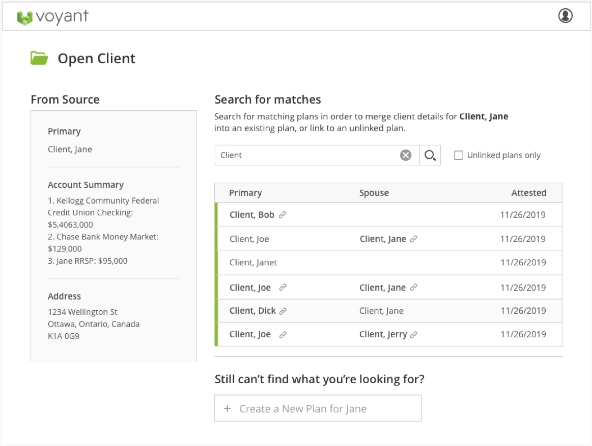


adviserGo Domain + Launch-in-Context Person URL + integration person\_id

*E.g.* [*https://planwithvoyant.com/advisergo/#/launch/person/bd72e1ea-230d-4ee1-b74e-c4dd56d954e3*](https://planwithvoyant.com/advisergo/#/launch/person/bd72e1ea-230d-4ee1-b74e-c4dd56d954e3)

**Household ID Integration flow**

Voyant Plan can be created directly from integration’s household / family details. Existing Voyant plans can be updated by connecting to an integration household ID.



**URL & Required Parameters**

Family ID must be provided

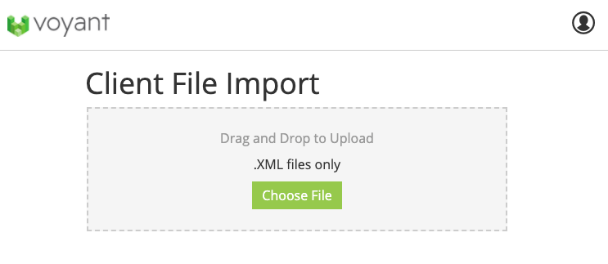


adviserGo Domain + Launch-in-Context Family URL + integration family\_id

*E.g.* [*https://planwithvoyant.com/advisergo/#/launch/family/bd72e1ea-230d-4ee1-b74e-c4dd56d954e3*](https://planwithvoyant.com/advisergo/#/launch/family/bd72e1ea-230d-4ee1-b74e-c4dd56d954e3)

**Launch from a File**

Voyant Plan can be created directly from a client export file. Existing Voyant plans can also be updated directly from client file exports.



**URL**



adviserGo Domain + Launch-in-Context File URL

*E.g.* [*https://planwithvoyant.com/advisergo/#/launch/file*](https://planwithvoyant.com/advisergo/#/launch/file)

Appendix G: Secure Connectivity

**Connections initiated by the Voyant Server (Outgoing Connections from Voyant)**

Connections made by Voyant to services hosted by outside partners are made using https and the TLS 1.2/1.3 protocol. 2-way SSL or so-called **Client certificates are supported for these connections**.   
  
These services are most often secured via OAuth 2.0, with client credentials flow or authorization grant flow being the most common grant types.

**Connections initiated by external services (Incoming Connections to Voyant)**

Connections initiated by partners to Voyant APIs must use https connections in accordance with the latest AWS TLS Policy. As of now this is, ELBSecurityPolicy-FS-1-2-Res-2020-10. (more details can be found at[**https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html**](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html))  
**These connections cannot be secured with client certificates currently**.

For enterprise siloed environments, IP whitelisting is usually employed to limit the source of incoming connections to partner controlled CIDR blocks.

These connections are mostly secured via OAuth 2.0 client credentials grant type.

**Static IPs for Voyant Environments**

With the use of AWS Application Load Balancers (ALBs), Voyant environments do not **currently** have a mechanism to provide for fixed IP addresses for incoming connections.

For outgoing connections, Voyant environments use AWS NAT Gateways, which provide for a set a fixed IP addresses (2 or 4 depending on the presence of a Voyant client facing portal infrastructure) for a given Voyant environment. This allows partners to secure their services used by the Voyant application, not only with client certificates, but also via IP whitelisting.

**VPN Connections for Enterprise Siloed Environments**

AWS does supports site-to-site VPN connections between Virtual Private Clouds (VPCs) and customer VPN devices. While Voyant has employed these in an ad hoc fashion for specific environments, this is not a standard part of the Voyant application environment. The use of such a setup requires manual confirmation for each Voyant environment. The implementation of a site-to-site VPN requires significant additional fees on top the Voyant standard siloed environment hosting fees.