fSeries API

fSeries API makes it possible to share data and documents with external systems. Any fSeries DSD or fDocs/fSheets template may be made accessible and subject to the same security and permission structures as are available to internal users of fSeries.

Contents

Making an API

Making an API is a simple process of specifying the entity (DSD or fDocs/fSheets template) concerned together with some additional details such as the format required and the URL route used to access it.

Consuming an API

There are two types of API consumers:

- External systems
- Other fSeries systems

In both cases the “user” accessing the API is set up on the fSeries ‘server’ system (marked as an fSeries API User). It is then subject to all of the same security and permission set up as any other user of fSeries.

External systems first obtain a short term token based on their designated fSeries API User's id and password. They can then access any API available to them.

Other fSeries systems simply specify the fSeries 'server' system as an API source and specify the fSeries API User credentials. fSeries then takes care of accessing documentation and requesting tokens and data. The fSeries 'server' system becomes another source of data as well as a source of documents that may be added to menus.

Setting Up fSeries API

Once the APIs module is activated in your fSeries system go to fAdmin > Account API to set up the account's API. This is where your API is defined. The contents of this page are used in generating the OpenAPI documentation used by other systems (including other fSeries systems) to be aware of the options made available from your fSeries system.

Note that there are two ways in which data may be returned. The first, Data Only returns raw fSeries content, the second, JSON:API returns it in the JSON:API format which includes a "data" element, which is the same as the Data Only content.

If you wish to use a service such as Postman to test the fSeries API, you can download the OpenAPI documentation from this page either as text in the clipboard or saved as a file.

fSeries API User

Each consumer of fSeries API must be represented as a user in the system, marked as an fSeries API User. It is recommended that these users have only the 'User' role, but this is not essential.

Once such a user is added go to the 'More Options' on the user display page and select 'API Credentials' to enter a password. This password is only used to obtain a short term access token. It, together with the User Id must be shared with the calling system in order for them to access your fSeries API.

It is recommended that each consuming system has at least one fSeries API User for both security and audit. In addition, it is recommended that you consider multiple fSeries API Users for each consuming system if it would assist in security and audit, for example by giving different permissions and access roles to different fSeries API Users. This enables consuming systems to have some control of the data and documents available to different callers from their systems or in different areas of their application.

Creating APIs

To create an API to be made available to your consumers, go to the APIs module and click "New API" and add the general entity properties as with all fSeries entities. Save and Design to set up the details of your API.

There are three API entity types:

- fData – return data from an fSeries DSD either as data or as a spreadsheet
- fDocs – return a document generated from an fDocs template
- fSheets – return a document generated from an fSheets template

fData API

Select the DSD on which to base the API. You may then select a specific Data Group from within the DSD to limit the data returned to just that Data Group.

There are a number of formats available for this type of API. Use the publicly available fAPI dll (from nuget) for class definitions of each format.

fData Internal Content

This is the default format for an fData API and returns the data in the same format as fData uses internally. This is a collection of data groups each with a list of rows. Each row has a collection of columns with id, value and formattedValue attributes. The content definition is in fSeries.API.Response.fData class.

Spreadsheet

The spreadsheet format option generates an xlsx file based on the data generated (full DSD or just one data group). The content of the response is defined in the fSeries.API.Response.Document class.

Data Group Table

Available for single data group APIs, this format is a simple table structure with a list of rows, each with a list of columns containing id and value. The content of the response is defined in the fSeries.API.Response.Table class.

fDocs / fSheets API

Select the template (fDocs or fSheets) on which to base your API. For fDocs templates you may further specify the format to be generated. If not specified it will either use the default (docx) or may be specified by the calling system as a parameter.

Use the publicly available fAPI dll (from nuget) for class definitions of each format. The fSeries.API.Response.Document class defines the response, which contains:

- Output – a byte array of the generated document
- FileName – the name of the file generated
- Format – e.g. pdf for an fDocs document where pdf was selected
- MimeType – e.g. application/vnd.openxmlformats-officedocument.wordprocessingml.document for docx

Route

Once you have created and published your API design you need to specify the route by which consumer system will access it.

By specifying the route of each API you can design a RESTful structure to your APIs. The route consists of a series of values that represent classifications and parameters. For example:

/Customer/Orders/{Id}

Could return a specific customer order ({Id}).

Optionally you can prefix your route with a version number in order to provide version control over your APIs.

The parameters may be any value that the API's underlying DSD permits, plus f_Format if your API allows for the response format to be specified by the consumer.

When the consumer accesses you API they will use the fSeries API Server followed by “Api” then optionally the version and then the route you specify with parameters replaced. For example:

https://www.myapi.com/Api/1.0/Customer/Order/2361

Discovery

External system builders need to have access to documentation about your APIs in order to build their own connections and applications. fSeries API provides OpenAPI documentation for this purpose (it is also used by other fSeries systems to connect to yours).

First, the external system will need an access token. See the section below on obtaining and using an access token for more information.

To obtain the documentation, they must use the /openapi.json endpoint (e.g. https://www.myapi.com/openapi.json). This will return an OpenAPI Json document will full details of the server and available paths, including parameters available.

Some custom values are also included. These are used by other fSeries systems but may be useful to other consumers as well:

x-fDocs. x-fSheets and x-fData are true/false values at the top level indicating whether your fSeries API includes any fDocs, fSheets or fData responses respectively.

For each path (route) there are two values:

- x-output indicates the format of response
- x-entity identifies the APIs entity in the fSeries system

Access Token

External systems need to obtain an access token in order to make use of the APIs or documentation. This is obtained from the POST endpoint /Security/RequestToken where the form is as follows:

```
{
	{“UserId”:”[fSeries API User Id]”},
	{“Password”:”[fSeries API User password]”},
	{“CallerId”:”[optional value to record the user from the calling system]”}
}
```

The token returned must then be used as a Bearer token in the Authorization header when calling any fSeries API endpoint.

fSeries to fSeries

A significant purpose of fSeries API is to connect two or more fSeries systems. One fSeries system can access data and/or documents in another fSeries system simply by adding the ‘server’ fSeries API as a data source.

It is then possible to configure fSeries act as a data hub, document hub or in an open network of data/document sharing systems.

fSeries API Data Source

To obtain access to another fSeries system, add a new API source to your system. You will need the server address (e.g. https://www.myapi.com and then check the "fSeries API" option to enable adding the fSeries API User id and password provided by the administrator of the other fSeries.

Your fSeries will automatically discover the available APIs in the other fSeries and make them available within you fSeries.

Any number of other fSeries systems may be connected in this way.

Using fSeries API Data

When designer a DSD, if you have any fSeries API sources which contain fData based APIs you can select a Data Group Type of "fSeries API DSD" or "fSeries API Data Group". Select the source and then the Path from within the source (the Paths are the routes set up in the other fSeries API). For the Data Group option, select the specific data group from with the other fSeries API's DSD.

You will also be asked to specify the values (or placeholders) to be used to set the parameters for the API.

For the Data Group option, that's it, the data will be gathered from the other fSeries and added to your DSD data.

For the DSD option all of the data from the other fSeries API's DSD and you can then reference it using a "DSDDG" data group type. This is the same as if you had included one of your own DSDs in the DSD; you simply specify the data group containing the DSD and then the Data Group in that DSD that you want to use.

Using fSeries API Documents

You can include fSeries API documents in a menu.

If you have fSeries API sources that contain fDocs or fSheets APIs, the menu builder will include a menu type of "API Document". Click on the type to add a document from another fSeries system.

You will be asked to select the API source and then the document from that source that you wish to include in the meu.

If the document is from an fDocs template and the API has not been set to generate a specific format (e.g. PDF) then you can select the format to be generated. This and all other settings are exactly the same as for fDocs and fSheets menu items.

When the menu is run, the user can click on the option to obtain the document directly from the other fSeries system.