INSURANCE TECHNOLOGIES

Illustration Integrations APIs (SDK)

FIRELIGHT BASE

FireLight®

Platform

ILLUSTRATION INTEGRATIONS APIS (SDK)

Draft: 1.0

Published: August 09, 2022

Insurance Technologies, LLC

Insurance Technologies, LLC
Two South Cascade Avenue
Colorado Springs, CO 80903
USA
Phone: 719.442.6400
FAX: 719.442.0600

Internet E-Mail: info@insurancetechnologies.com
Website: http://www.insurancetechnologies.com

Table of Contents

Design Approach - Illustration Integrations APIs (SDK)

Current implementations of the IS-API that FireLight communicates with are all ForeSight implementations. Insurance Technologies would like to broaden this capability to include calculation and report generation applications that are not ForeSight implementations. This would require developers outside of Insurance Technologies to be able to implement IS-API applications that FireLight could interact with. By formally defining the API that FireLight would call in an industry standard OAS document, Insurance Technologies would enable external developers to develop these IS-API applications. Further effort by Insurance Technologies that would enhance the ability for external developers to implement IS-API applications would include providing a site for documentation that could further explain the IS-API requirements, and to provide a working example of such an IS-API those outside developers could reference as a guide for their own IS-API implementations. Collectively, the OAS document, the additional IS-API documentation site, and the working example of an IS-API constitute the SDK that Insurance Technologies would provide outside developers.

1 Firelight Admin Tool

1.1 Illustration Services - Admin Tool - Self-Help for Organization Settings and External Integrations

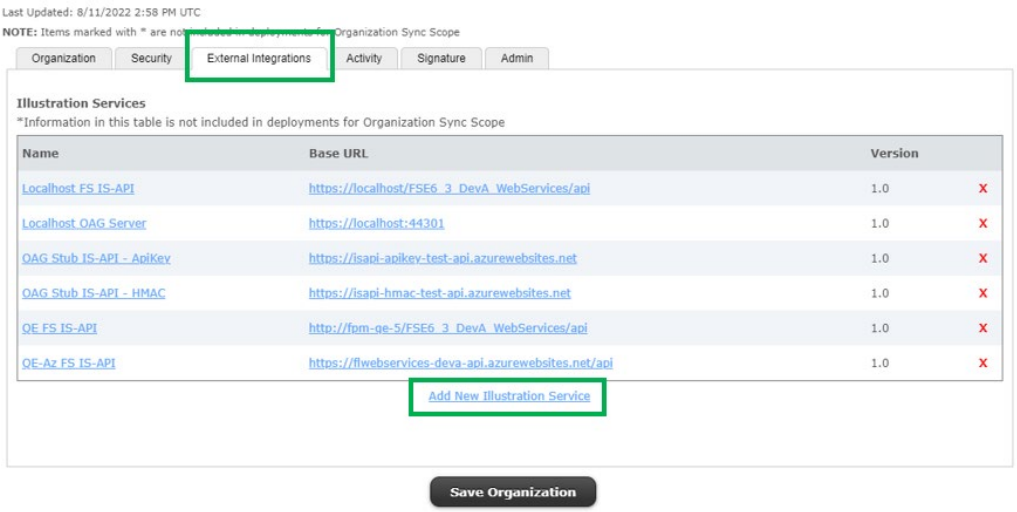
New additions have been added to the Admin Tool to set-up a new “External Integration” for a New Illustration Service API. This new option has been created to be able to take on new IllustServices Endpoints for every product.

When in the Admin Tool, under the Organization Settings we will be adding a new tab between the "Security" and "Activity" tabs, labeled "External Integrations".

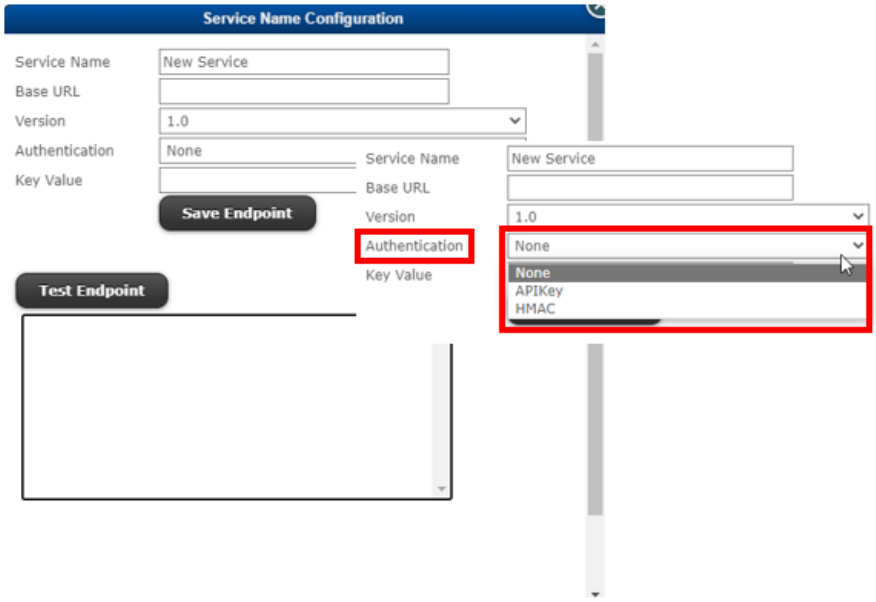
When the tab, “External Integrations” has been clicked on, there will be a new table that will be displayed. The table will display the columns for: Name, Base URL, Version, and a red “X” for the option to delete the service.

Under this newly created table, it lists the available Services.

- Data in the tab stays in synch with the product selected, under the Product Configuration > Products

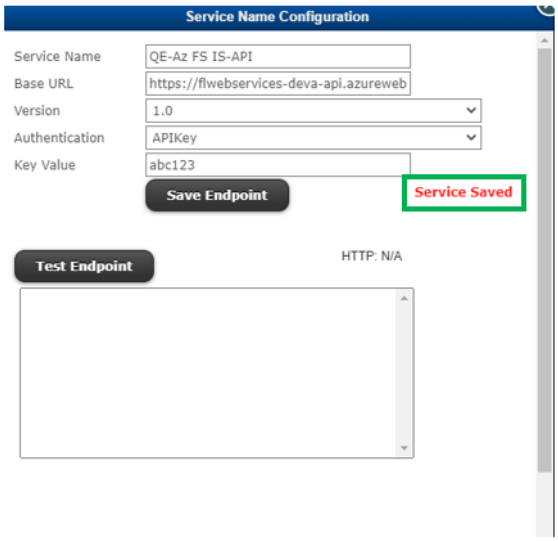


The "Add New Illustration Service" link will pop-up a new input window, labeled "Service Name Configuration" with new input fields:

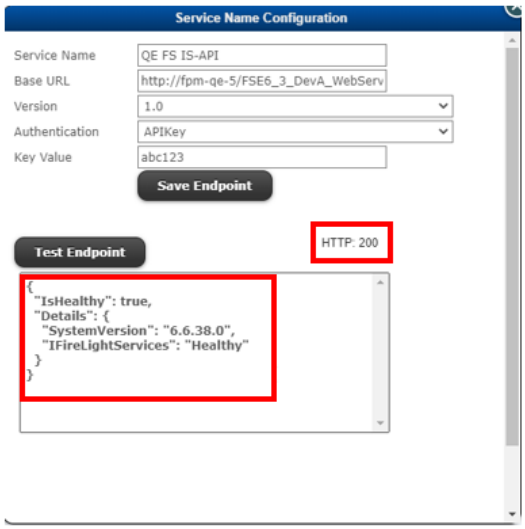


- Service Name – (text box) - once a new one has been created, it will be available under the newly created table displayed on the "External Integrations" tab page
- Base URL - (seleURL - text box) - this will be data that the user will input based on their URL
- Version - (select from dropct from drop-down - ex. 1.0) - Day 1 we are going on only have 1.0
- Authentication - (select from dropct from drop-down - ex. API key or HMAC) - select from the drop-down
- Key Value (text box) – currently contains: None, APIKey, and HMAC

- "Save Endpoint" button - which will save the new or updated input data. Once the new service is saved, a “Service Saved” notice will be displayed next to the button. It will show up on the list in the box under the "External Integrations" tab.
    - After the data has been input, click on the "Save Endpoint" to save the newly created or updated service. A message will display in "Red" stating "Service Saved".



- "Test Endpoint" webservice button - this button is utilized to test the New Service set-up
- Text box that will receive the test results back - this will return the JSON results to let the user know if the set-up is "Healthy" or “Unhealthy”
- -HTTP: N/A or a Code



Acceptance Criteria

Select from "External Integrations" - click on "Add New Illustration Service" beneath the new box that lists Illustration Services that have already been configured. (See mock-ups)

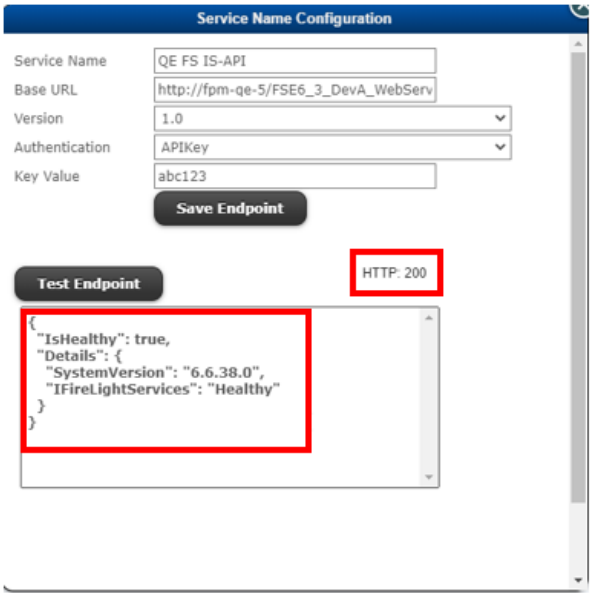
A new pop-up window will appear for the "Service Name Configuration" data to be input:

- Input in the "Service Name" - this is provided by the client.
- Input in "Base URL" - this is provided by the client.
- Select from "Version" from the drop-down list. On Day 1, we are going to populate this dropdown with “1.0”.
- Select from "Authentication" drop-down - APIKey or HMAC.
- Input the "Key Value" - the client will define this - for testing you will use "abc123" (which will test against ForeSight)
- Click on "Save Endpoint" - to save your data that you have input. You will see a message appear in Red that states "Service Saved" and you will see it listed in the box under the "External Integrations" tab (see mock-ups)
- Click on "Test Endpoint" - you will receive and HTTP: 200, or HTTP: 401, or HTTP: 500 type message back (see examples below).
- Once the set-up is complete and an HTTP: 200 and the message returned shows, “IsHealthy”: true, click on the "X" in the upper right-hand corner to close the window.
- Application returns an unhealthy response for the health check during some defined interval.

Results from the "Test Endpoint" - will be shown in the box.

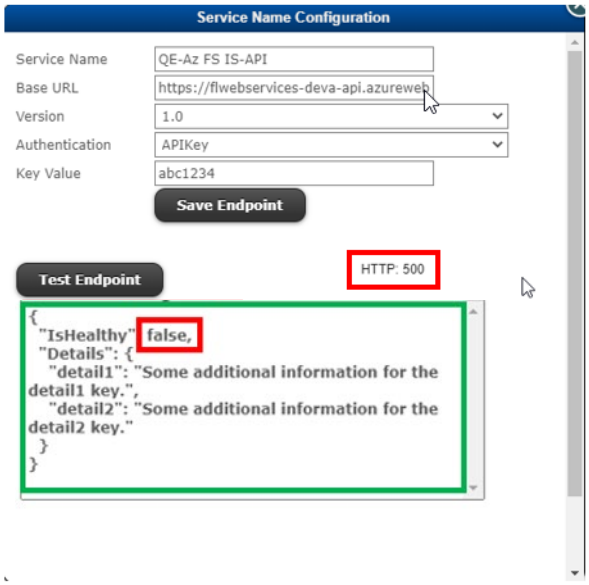
Example of a “Healthy” Return:
Valid: HTTP: 200


```
{
 "isHealthy": true,
 "details": {
  "database": "Healthy",
  "cache": "Healthy"
 }
```



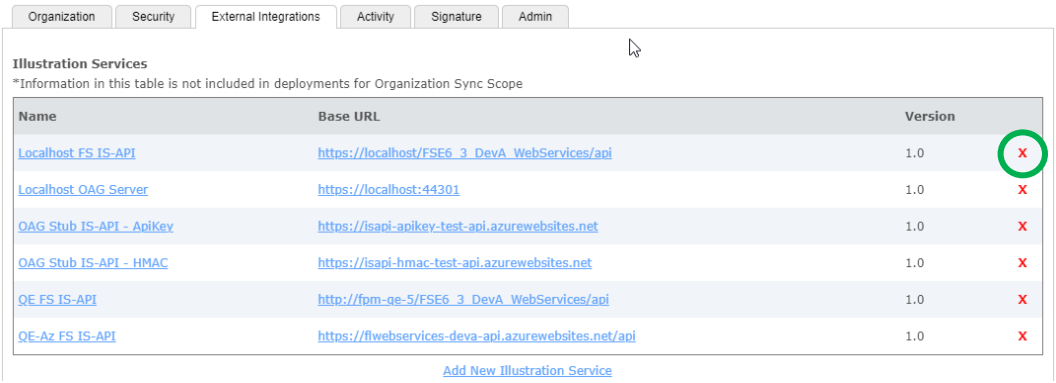
Example of a “UnHealthy” Return:

If an error, you would receive a response such as : HTTP: 401 or HTTP: 500

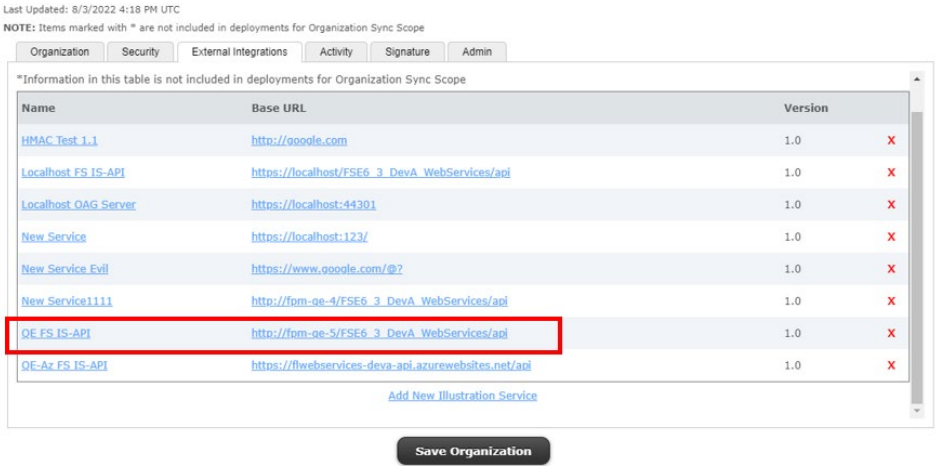


1.2 Deletion or Update of an Existing Service Name Configuration

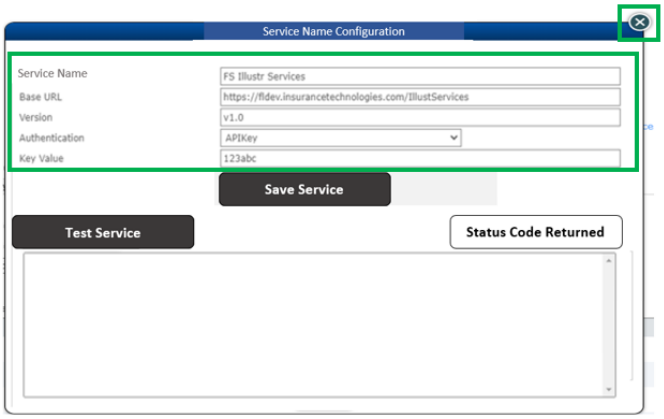
When there is a need to Delete and existing Illustration Service, click on the red “X” next to the corresponding “Name”.



When there is a need to Update an existing Service Name Record, click on the Service Name in the table.



Once you have clicked on the Service Name Record, the Service Name Configuration window will pop-up for editing. Change up the fields that need to be updated, click on the "Save Service" button to save the changes. If you click on the "X" in the upper right-hand corner before clicking on "Save Service" your changes will not be saved.



Acceptance Criteria

Updating an Existing Service Name configuration:

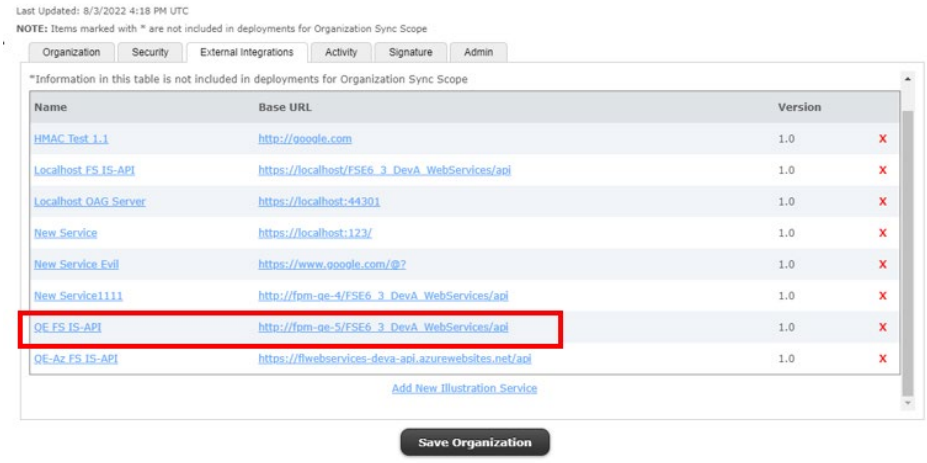
- 1. Click on the "Name" from the IllustServices Application Name table. Example in mock-up: FS Illust Service
- 2. The Service Name Configuration box will then pop-up
- 3. Make any necessary changes.
- 4. Click on "Save Service Name" button to save the changes.
- 5. Run the "Test Service Name" button to make sure that the new changes work. You will see a "Status Code Returned" in the text box. You will also see the JSON results returned in the text box below.
- 6. If completed, click the "X" button in the upper right-hand corner of the pop-up window.
- If you click on the "X" to close the pop-up window prior to clicking on the "Save Service Name" button, the window will close and no changes will be saved.

1.3 Additional Item to the Product Config > Products > Products Details Tab

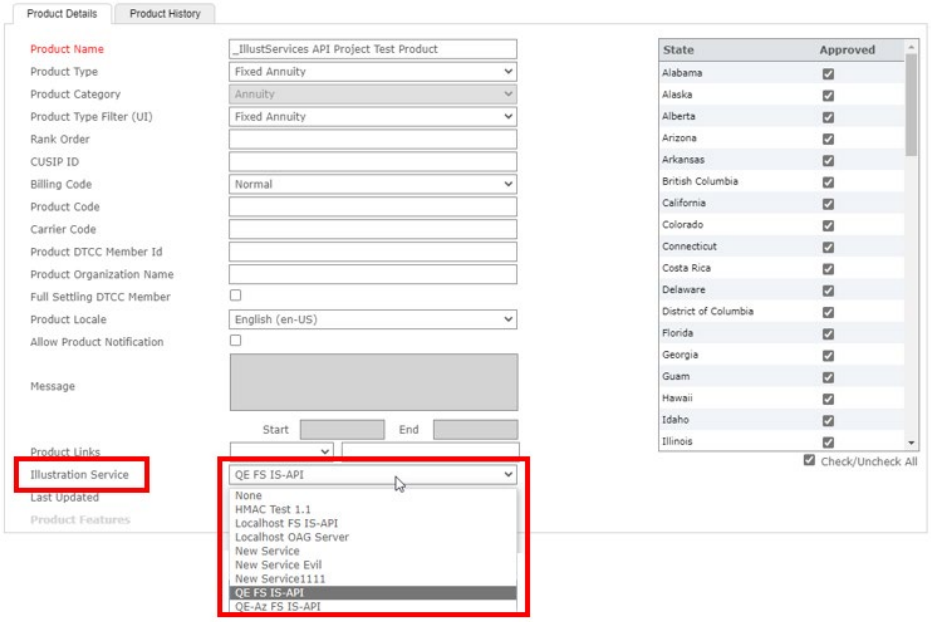
Additional item to be added to the Product Details tab. The new item will be "Illustration Service" and will have a drop-down listed associated with it. The drop-down list will contain the list of Service Names that have been created on the Organization Config > Organization Settings > External Integrations > Name

Once a new Illustration Service is set-up under the Organization Configuration and External Integrations tab, it will now be available as a selection under the Product Details tab (see mock-ups below). This will now create the link between the Product and calling the associated Illustration API.

Organization Configuration > External Integrations Tab



Product Configurations > Products > Illustration Service > Drop-Down List:



Acceptance Criteria

- When setting up New Product or updating an existing Product, the user will now be able to associate an Illustration Service to that Product.
- A new field, "Illustration Service" has been added to and is visible on the Product Details tab. This is found under: Product Configuration > Products > Product Details Tab
- Click on the drop-down list, next to "Illustration Service", to select the service that has been set-up.
- If the Illustration Service that the user is wanting to link to this Product is not showing on the list, then the Illustration Service was not set-up or saved under the Organization Settings > External Integrations tab. The user will need to go back and set-up and save the "New Illustration Service" that they would like to use.
- If the Illustration Service name is listed, then it has been set-up and ready to be linked to the product and used.
- After the New Product or Existing Product has been updated, click on the "Save Product Config" button.

1.4 Database Tasks for IllustServices API Settings

New database tasks will be created for the Illustration API Settings:

- Tied to Authentication
- Running Queries
- Pulling in data

2 SDK Documentation Site

2.1 Create Working Example Content

Create the documentation files for the part of the documentation site that explains the interaction of the example API (ForeSight). The intent is not to modify ForeSight, but to simply document what is already implemented.

Acceptance Criteria

- 1. Input map for FA and FIA products documented.
- 2. Example JSON request files that could be used for testing.
- 3. Verification that steps to interact with the API are working as described in the documentation.

2.2 Create Custom Docfx Template

Using example templates like darkfx, discordfx, etc, create a custom template for Docfx that can be used for the color scheme, styling of HTML elements, and client side scripting functionality.

Acceptance Criteria

- All relevant template files are created and added to the repo. The Docfx.json file is updated to use the new template. The Docfx build completes without any errors or warnings with the custom template.

2.3 Create General Content

Create the general content that applies to the API and is not product specific.

Acceptance Criteria

- Content to allow access to the OAS document.
- Additional description of the API Key for authentication and authorization.
- All model and endpoint documentation.
- Description of the use of common tags and a reference to the complete list of common tags.

2.4 Create DevOps Project

Create the necessary requirements for the DocFX static site.

Acceptance Criteria

All resources created:

- Create DocFX Repo – Creation of the Repo for the DocFX files. Existing files from the IT-FLT-RD/Illustration-SDK project can be used.

- Create Service Connection to Azure Subscription – Create a Service Connection to the Azure Subscription that will hold the storage account that the DocFX static site is hosted in.
- Create CI Build – Create the build pipeline that will build the DocFX site and copy the static site files to the storage account that will host the static website. Use the Service Connection to copy the files to the storage account
- Create Pull Request Build – Create the build definition that will be used to validate pull requests.
- Define Main Branch Policy – Define the branch policy for the main branch. Require pull requests and code reviews.

2.5 Create Azure Resources

Create all resources that will be used by the SDK documentation site.

Acceptance Criteria

All resources created:

- Storage account – this will hold the static website blob.
- Resource group(s).

3 FireLight EGApp changes

3.1 Detect IllustServices API Version and Use Appropriate Models

When calling an IllustServices API, determine what version of the API is defined for that endpoint(s), and use the appropriate model class definitions to serialize and deserialize the request and response objects.

Acceptance Criteria

- 1. The desired IllustServices API endpoint version has been determined based on the product's configuration data.
- 2. The version 0 and version 1 API model classes are defined separately. (Day 1 will be set for version 1).
- 3. The correct model classes are used for the IllustServices API that is getting called.
- 4. Current illustrations run as expected.

3.2 Define Database Queries for IllustServices API Configuration

Define that queries that will be used to retrieve the IllustServices API configuration data that is saved by the Admin tool. Define model classes that will be used to hold the data retrieved from the queries.

Acceptance Criteria

- Query SQL syntax has been defined.
- Data model classes have been defined and can hold the data returned by the query.

3.3 Use Health Endpoint for Version 1 Implementations

Version 0 implementations do not have a standardized Health endpoint. Version 1 implementations do have the Health endpoint defined. If the desired IllustServices API that is getting called implements version 1 of the API, then the Health endpoint should be used to determine if the IllustServices API endpoint is operational and available for illustrations.

The health of the IllustServices API endpoint can be determined by calling the Health endpoint and getting a "healthy" response. Applications that implement the Health endpoint but respond with an non-successful HTTP status code, or return an "unhealthy" response are excluded from illustration.

Acceptance Criteria

- User cannot see illustration values or reports if the system is unhealthy.
- Use PostMan to test but we do not have a app service to connect to this yet, so no address to point to.

3.4 Define Illustration Configuration Data for Existing Implementations

Move the configuration data that currently exists in the CalcRouter config file into the new product configuration tables for the endpoint URLs and the API version. All existing IllustServices API configurations will be configured as version 0.

Acceptance Criteria

- All existing illustration products are running as version 0 API implementations.
- CalcRouter config files have been removed for all products.

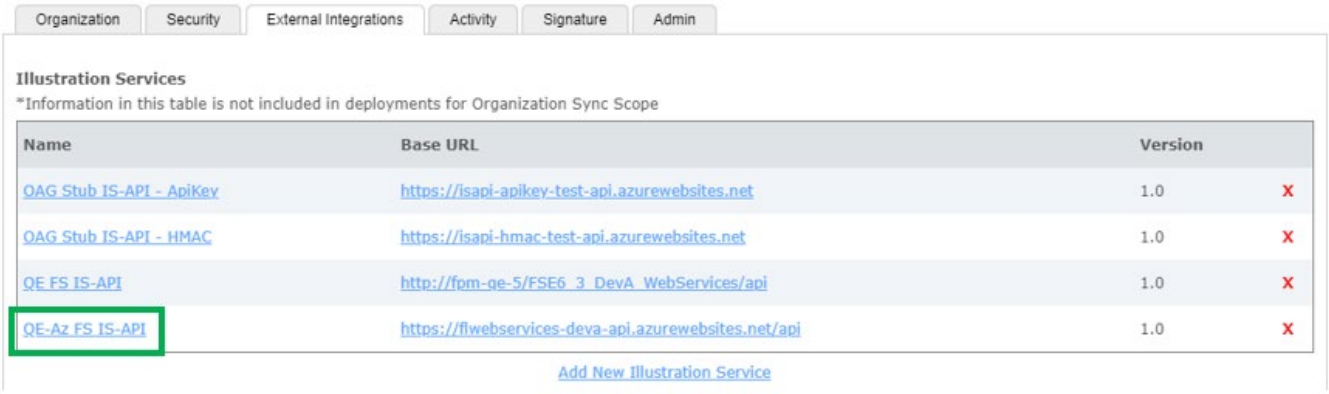
3.5 Create Server Stub Pipeline

Need to create ForeSight Test Site Endpoints. For testing purposes, we need to be able to receive both a "healthy/successful" and "unhealthy/unsuccessful" responses.

New Service has been created in both the Dev and QA environments.

A new Product has been created for testing. The new Service Name (also set-up) is: QE-Az FS IS-API that will be utilized.
Under the QA environment - several new endpoints have been created. In order to receive both the "healthy" and " unhealthy" responses, these would need to be tested at different times within the hour.

- First 45 minutes of the hour, when testing, you will receive a "healthy" response back.
- Last 15 minutes of the hour, when testing, you will receive a "unhealthy" response back.



Acceptance Criteria

- `aspnetcore` generator template files modified so the OAG generation will generate code for the unhealthy response.
- Changes to the OAS document trigger a new build and deploy of the application.
- CI pipeline is modified to build the application, and the build artifacts are published for deployment.
- Steps for generating calculations and reports in the App:
    - Start New Page:
    - Click on: "Illustration"
    - Product Type: "Fixed Annuity"
    - Select: FS IS-API Fixed Annuity
    - FS IS-API Wizard screen will display.
        - Auto-populated: CUSIP, Issue State (corresponds to the "Jurisdiction", Product Line of Business = 2 (Fixed Annuity)
        - Input: Owner DOB, Primary Owner First Name, Primary Owner Last Name, Owner Gender
    - Click on "Illustrate" for a "Quick View" of the calculation values that are returned
    - Click on "Print" and the PDF of the report will be displayed

4 Base ForeSight IS-API Upgrade

Encompasses all of the work that is required to upgrade the base ForeSight IS-API to use the new API standard that is part of the SDK.

4.1 Add Health Endpoint

Add the Health endpoint, as specified in the OAS document.

Acceptance Criteria

- 1. The {Base URL}/Health GET operation returns a 200 status code and relevant health data in a response.
- 2. The {Base URL}/Health GET operation returns a 500 status code if an exception is generated, and a relevant error response is generated.

4.2 Change FireLight Controller to Illustration Controller

Move the existing /FireLight/* endpoints to the new /Illustration/* controller. Implement correct exception handling.

Acceptance Criteria

- 1. The Illustration/
- endpoints can be called and return appropriate HttpStatus codes and response models.

4.3 Update Model Definitions

Update all model classes that are being used for the IS-API to use new models that are based on the new OAS document. Use code generated classes as a reference.

Acceptance Criteria

- 1. All model classes have been updated.
- 2. Model data is getting populated correctly.

4.4 Add API Key Authentication

Add the functionality to validate the API key that is passed in the header, per the OAS document. Use a hard-coded API key value for testing.

Acceptance Criteria

- 1. The base ForeSight IS-API will generate HttpStatus codes as specified in the OAS document.
- 2. The base ForeSight IS-API will correctly validate the passed API Key value.

4.5 Update ForeSight WebServicesTester Tool

Update the WebServicesTester tool to use the API Key authentication/authorization.

Acceptance Criteria

- The WebServicesTester can authenticate against the ForeSight implementation of the IllustServicesAPI using the API Key defined in the IllustServicesApiSdk OAS document.

5 HMAC Authentication for SDK

5.1 HMAC Authentication Documentation Content

Documentation site can be found here: IS-API SDK Documentation Site | IllustServices API SDK (windows.net)

5.2 Code Generator Template for HMAC

Testing for this story will require the OAG server stub to be generated and configured to use HMAC.

Acceptance Criteria

- FLT user can use the OAG server stub application with HMAC as an authentication method. (Create application that uses the HMAC authentication.)
- Service Name: OAG Stub IS-API – HMAC
- Base URL: https://isapi-hmac-test-api.azurewebsites.net
- Version: 1.0
- Authentication: HMAC
- Key Value = my-secret-value