



Data Furnisher API Implementation Guide

Introduction

Purpose e-OSCAR API Services is a Representational State Transfer (REST) service for Data Furnishers to create, update and access AUD, BRR, ACDV and Notifications data.

e-OSCAR API Services is intended for Data Furnishers to receive and respond to Automated Credit Dispute Verifications (ACDVs), submit Automated Universal Data forms (AUDs), receive Notifications from Consumer Reporting Agencies and download Archive data within their own host system of record, eliminating the need to access the e-OSCAR interactive application for transaction processing.

This document provides the Data Furnisher with detailed information regarding the e-OSCAR API Services for use during the following:

- Technical Specification Review
- User Acceptance Preparation and Testing
- API Production Implementation
- Ongoing API Processing Support

NOTE: Technical specification documents have been provided separately. Please refer to the technical specifications as needed in the SwaggerHub application.

Technical Prerequisites The following prerequisites must be met by the Data Furnisher to utilize API Services:

- Ability to access the document repository www.SwaggerHub.com
- Establish a web service client software that can support RESTful web service calls



Introduction, continued

Contact Information

Role	Contact Name	Email Address	Phone Number
Director of Client Relations	Joel Strickland	jstrickland@newmgtservices.com	(904) 394-1479
Client Relations Manager	Christine Macdonald	cmacdonald@newmgtservices.com	(904) 289-8559
Technical Engineer	Charles Williams	cwilliams@newmgtservices.com	(904) 394-1476

Contacts for Ongoing API Support Post Implementation

e-OSCAR API Support is available via our Jira ticketing platform. Upon provisioning access to the UAT environment for testing, an invite will be sent to all users listed in the Support Contact section of the API Onboarding form that is required to be completed by representatives at your organization. Users listed in the approved contacts section for support will be able to initiate service tickets for any support related issues.



API Implementation Stages

API Services Implementation Stages

Implementation of API Services occurs in the following stages:

- Access to the technical documentation provided for review
- Data Furnisher development
- Billing of Onboarding fee and creation of Pricing Tier Analysis
- User Acceptance Testing
 - UAT IP Address whitelisting
 - UAT API Key Generation
 - Certification of successful testing in UAT
- Production Go Live
 - Production IP Address Whitelisting
 - Production API Key Generation
 - Go-Live Worksession
- Billing and Invoicing

Access to Technical Specifications

API Services Technical Specifications are housed in SwaggerHub. OLDE will provision access to the documentation library upon request. The following guidelines apply when requesting access:

- Release of this documentation requires a fully executed Non-Disclosure Agreement
- The Data Furnisher will be governed by the existing API User Agreement. This is not an executable document and is available in the www.e-OSCAR.org informational website. This usage agreement will govern the access to the documentation in SwaggerHub and is agreed to upon accessing the SwaggerHub documentation.
- Access to SwaggerHub is limited to no more than three Data Furnisher users. The Data Furnisher must provide the name and email address of the individual(s) that will require access to the technical documentation.
- Once granted, users may access the documentation using the following link: <https://app.SwaggerHub.com/home>
- It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).

NOTE: Users will have 14 calendar days from receipt of the registration request email to register in SwaggerHub. If the user does not register within 14 calendar days of receiving the registration request, access will be revoked.



API Implementation Stages, continued

Data Furnisher Development To achieve the benefits of API Services, the Data Furnisher’s development effort will vary depending on the Data Furnisher’s internal business and compliance requirements.

NOTE: The internal development effort is the responsibility of the Data Furnisher.

Pricing and Fees Upon request by the Data Furnisher to participate in UAT, the Data Furnisher will be invoiced for the API Onboarding fee.

OLDE will generate an invoice for the Onboarding fee for API Services and submit to the billing contact on record for payment. Data Furnisher must review the Pricing Matrix available at www.e-OSCAR.org.

OLDE will complete a Pricing Tier Analysis to identify the Data Furnisher billable tier for the use of API Services.

User Acceptance Testing The User Acceptance Testing has the following objective:

To ensure that the Data Furnisher’s internal build can successfully accept, process and submit transactions that e-OSCAR sends to or receives from the Data Furnisher.

OLDE will:

- Generate and provide an API Key to the Data Furnisher that must be used for each API call in the UAT environment. The API key is unique to the Registration ID. This will be provided to the Data Furnisher via encrypted email.
- Load data to the UAT environment so that the Data Furnisher may complete the “Get” calls for testing purposes. The Data Furnisher will need to test the “Submit” calls directly from their host system of record.
- Enable the API Feature flag within the UAT environment to allow the API User Management screen to become available.

Data Furnisher must:

- Provide the external IP Address for the Machine(s) that will talk to the services for UAT.
 - Create an API User in the UAT environment.
-



API Implementation Stages, continued

Certification of Successful UAT

UAT will be considered “successful” when the Data Furnisher can:

- Authenticate with the Service
- Send and receive data with the Services absent of repeated technical errors
- Send data that may fail for Metro2 validation and/or field validation rules but can successfully process upon correction of the errors
- All Services that will be used by the Data Furnisher must be tested in UAT and confirmed via the API Final Acceptance Form
- Multiple records have been successfully submitted with different indicators and response codes

e-OSCAR will provide access to our UAT environment to allow you to test the connectivity of the API endpoints you choose to utilize in the production environment. The testing effort for API Services is intended only to ensure connectivity to the endpoints; testing of your internal processing system(s), user interface(s), automation, etc., is outside the scope of Endpoint Connectivity Testing. If you wish to perform additional, non-connectivity testing, and you require additional testing data (i.e. additional ACDV records, specific Dispute Codes, specific data points, etc.), you must specifically request additional data loads. e-OSCAR reserves the right to charge for additional data loads ACDV requests for endpoint connectivity testing will be loaded to the UAT environment daily based on the pricing tier e-OSCAR anticipates that you will be assigned to in production.

The number of ACDV requests will be loaded as follows based on tier:

Tier 1 – 10 ACDV Requests per CRA (maximum of 40 total daily)

Tier 2 – 50 ACDV Requests per CRA (maximum of 200 total daily)

Tier 3 – 100 ACDV Requests per CRA (maximum of 400 total daily)

Tier 4 – 250 ACDV Requests per CRA (maximum of 1000 total daily)

Data Furnishers will need to supply a Subscriber code in UAT for each CRA that they currently report to. NOTE – Please generate a fake Subscriber Code in UAT if you do not already have fake Subscriber Codes existing in the UAT environment. Please do not provide Subscriber Codes currently utilized in the production environment. Given the way AUD transactions are processed, OLDE will not provide any data for testing AUD related endpoints. Please note that e-OSCAR will not be able to provide application guidance or technical support for any testing efforts related to internal Data Furnisher processes, procedures, or systems.

Implementation to Production

OLDE strongly recommends that Go Live is within 2 weeks after the successful conclusion of UAT. The API Deployment Form will be provided to the Data Furnisher. This form will identify all of the steps and the timeline of those steps to complete a successful launch of API Services to Production.

API Implementation Stages, continued

Go Live Worksession

Once the Production IP address(s) has been whitelisted, the Production API Key has been generated and the API User has been created in Production, a worksession will be held to ensure that the API Services can be successfully implemented in the Production environment.

The Worksession must include the following teams:

- Data Furnisher end users
- Data Furnisher project manager / coordinator
- Data Furnisher technical team members required to troubleshoot

Data Furnishers must confirm that all required Data Furnisher team members are available for the scheduled implementation / Go Live.

In preparation for the Go-Live session, please review the following:

- The Production IP address(s) has not changed
- The API User has been created in the e-OSCAR interactive application

Billing and Invoicing

Upon successful implementation of API Services to the Production environment, OLDE will update the Data Furnisher account to include invoicing for API Services.

API Services will require invoices to be paid monthly. Data Furnishers on quarterly billing cycles will be transitioned to monthly billing cycles subsequent to implementation of API Services in the Production Environment.

Current Monthly bill Data Furnisher – API Fee will appear on the next monthly invoice.

Current Quarterly bill Data Furnisher – If the Data Furnisher is billed quarterly, they will be converted to a monthly bill at the end of the current quarter. The API fees will then appear on the first monthly invoice:

January – March: Transition to monthly billing in April. API fee starts in April

April – June: Transition to monthly billing in July. API Fee starts in July

July – September: Transition to monthly billing in October. API fee starts in October

October – December: Transition to monthly billing in January. API Fee starts in January



API Services Process Specifications

Access to API Environments

e-OSCAR Services are available to the Data Furnisher in the following environments:

UAT:

Host Name

api-uat.e-oscar-web.net/eoscar-services

Base URL

<https://api-uat.e-oscar-web.net/eoscar-services>

Production:

Host Name

api.e-oscar-web.net/eoscar-services

Base URL

<https://api.e-oscar-web.net/eoscar-services>

NOTE: Technical specification documents have been provided separately. Please refer to the technical specifications as needed in the SwaggerHub database.

Creation of API User in e-OSCAR

OLDE requires that an API User is created in the e-OSCAR interactive application prior to performing any API calls regardless of environment (UAT – <https://uat.e-oscar-web.net> or Production – <https://e-oscar-web.net>).

- The e-OSCAR User Administrator for the UAT or Production environment will log into the e-OSCAR interactive application and access the Administration screen.
- The navigation bar on the left -hand side of the screen will populate with an option for API User Management.
- Select the option for API User Management and select the link to Create User in the Upper right-hand corner of the page.
- A User Role will need to be selected:
 - API Receiver – Completes “get” calls only
 - API Initiator – Completes both “get” and “submit” calls

NOTE: e-OSCAR interactive users cannot access API and e-OSCAR API users cannot access e-OSCAR interactively.

The first time that the API user accesses API Services, they must change the password using the following call: **`/auth/v1/changePwd`**.

The API user must then complete the Auth call to create a token for that user using the following call: **`/auth/v1`**.



API Services Process Specifications, continued

API Key and Authorization Token

OLDE will require that the following information is included in each API Call for API Services:

- **API Key** (generated by OLDE and provided via encrypted email to the Data Furnisher)
 - Ex. Http header: *x-api-key: f7c42aa3-0891-4ee4-9a6f-5594a3ea0c9c*
- **Authorization Token** (Generated by completing the /auth/v1 call)
 - Ex. Http header: *Authorization: Bearer eyJ0eXAioiJKV1QjLChbGiOiJSUzI1NiJ9.eyJzdWl0OiJBUeIU0VSIiwiaXNzIjoibExWSHJMUIZCeTNvY2N4Y2NWeTlrNTFoR1FSY2U3VSIsImhkdCI6MTU4MjEzNTc2NCwianRpljoiMzg3OTc4NiJ9.BKrnU8kcqsCsOoRfVsnzicsmNLNmHGzxGzTqzE9fwCOFbl_nX5zBYNfSr6zQ8EbWMSbDxHi5qcZri9TgVHyxfCYDXSbg-CXannh1IslyoVsDataFurnisher0eN4tl_CoREbsgQJmu99v7IYz0q8BmciadWOZkIWalaC73p8qUPTnOuu7bck_8Wt5SJ786DsWy_hfIVeXNY5oEtzgsaTzgiUbobN_k7PawOpv1gao6-lqSiwckkztDrbqBgVDjJn1JGblvV7UE-UstAULbPDp WXQERuzpvMoyv9Ee05hlFeCurSPx46TKRUHC72RtAyTFG8wBVKkhC7ZmDpdmQ68ZmSSRZhlBxVA*

Authorization Services

e-OSCAR Authorization Services provide authentication for all e-OSCAR Services.

Auth Services expose 2 endpoints:

- **/auth/v1/changePwd**
 - Accepts a Username, password (created in the e-OSCAR interactive application), new password and registration ID.
 - This is the first call that must be completed prior to completing any other calls within e-OSCAR Services.
- **/auth/v1**
 - Accepts a Username, password (new password created from the change password call) and registration ID.
 - This call will provide the Authorization Token required on all remaining calls in e-OSCAR Services.
 - The Authorization Token will expire after 12 hours of continuous activity or 30 minutes of inactivity.

These endpoints require the API Key to be passed in the request header.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Process Specifications, continued

Request Services e-OSCAR Request Services allow Data Furnishers to receive ACDV Request transactions as well as the associated images into the Data Furnishers own solution for response processing which will be sent back to e-OSCAR in the form of an ACDV Response.

Request Services expose multiple endpoints, the endpoints below are required to ingest the ACDV requests in Services:

`/acdvreq/v1/find`

- Accepts a date range, account number, SSN, status, Date Type, and returns ACDV Request Control Numbers matching the given search criteria. ACDV Control Numbers obtained via this call must have the details received using the `/acdvreq/v1/view/{acdVCtrlNum}` call prior to responding to the ACDV.

`/acdvreq/v1/view/{acdVCtrlNum}`

- Using the resulting data from the `/acdvreq/v1/find`, retrieves transaction data details and images for a single ACDV by ACDV Control Number. The status of the transaction must be Requested or Pending. If the status of the ACDV is Requested Status, it will be updated to Pending Status and the image accessed flag will be updated to Y.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Process Specifications, continued

Validation Services e-OSCAR Validation Services allow Data Furnishers to validate the records being sent to e-OSCAR prior to submission to check the data for valid values as well as Metro2 Compliance.

Validation Services are available for the following transaction types:

- ACDV Response
- AUD
- BRR

Validation Services expose 3 endpoints:

/acdvresp/v1/validate

- Provide the ACDV Response details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will not be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.

/aud/v1/validate

- Provide the AUD details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will not be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.

/brr/v1/validate

- Provide the BRR details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will not be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Process Specifications, continued

Submission Services e-OSCAR Submission Services allow Data Furnishers to submit a valid record to e-OSCAR so that it may be received by the CRAs. The submitted record will be checked for valid values and Metro2 Compliance before being accepted by the Service.

Submission Services are available for the following transaction types:

- ACDV Response
- AUD
- BRR

Submission Services expose 3 endpoints:

`/acdvsp/v1/submit`

- Provide the ACDV Response details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.
- ACDV transactions will also be reviewed for Response Code validations prior to submission to confirm that the Response Code sent meets the criteria of the response data provided.
- ACDV Control Numbers obtained using the `/acdvreq/v1/find` must have the details for the Control Number accessed by completing the `/acdvreq/v1/view/{acdCtrlNum}` prior to completing the `/acdvsp/v1/submit` call.

`/aud/v1/submit`

- Provide the AUD details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.

`/brr/v1/submit`

- Provide the BRR details and this call will return feedback regarding the validity of the data.
- Data sent using this endpoint will be stored in the e-OSCAR database.
- Up to 50 records may be sent for validation in a single validation call.
- All errors and successes will be returned at once.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Process Specifications, continued

Notification Services e-OSCAR Notification Services allow Data Furnishers to retrieve all Notifications generated by the CRAs and transmitted through e-OSCAR.

Notification Services are available for the following transaction types:

- AUD
- Block/DR (Dispute Response)

AUD Notifications expose 2 endpoints:

- **/audnotif/v1/find**
 - Accepts a date range, account number, SSN, status, Date Type and returns AUD Notification IDs matching the given search criteria.
- **/audnotif/v1/getList**
 - Using the resulting data from the **/audnotif/v1/find**, retrieves AUD Notifications for the Notification IDs entered.
 - Up to 50 AUD Control Numbers may be entered in each call.
 - AUD Notifications are returned in the order that the AUD Control Numbers are entered.

Block/DR Notifications expose 3 endpoints:

- **/notification/v1/find**
 - Accepts a doc type, date range, account number, SSN, status, Date Type and returns Block/DR Notification IDs matching the given search criteria.
- **/notification/v1/getList**
 - Using the resulting data from the **/notification/v1/find**, accepts a maximum of 50 Block/DR Control Numbers and returns the Block/DR Notifications in the same order.
- **/notification/v1/getImages**
 - Using the resulting data from the **/notification/v1/find**, accepts a maximum of 50 Block/DR Control Numbers and returns the processed requests in the same order.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Process Specifications, continued

Archive Services e-OSCAR Archive Services allow Data Furnishers the ability to retrieve transaction data for their unique registration ID and ingest it into their own reporting tools for research and archive purposes.

Archive Services are available for the following transaction types:

- AUD
- BRR
- ACDV Request
- ACDV Response

AUD Archive Services expose 3 endpoints:

- **`/aud/v1/find`**
 - Accepts a data range, account number, SSN and status as search criteria and returns AUD IDs matching the given criteria.
- **`/aud/v1/get/{id}`**
 - Using the resulting data from the **`/aud/v1/find`** call, enter a single AUD ID and the corresponding AUD Details are returned in the response.
- **`/aud/v1/getList`**
 - Using the resulting data from the **`/aud/v1/find`** call, accepts a maximum of 50 AUD IDs and returns the processed AUDs in the same order.

BRR Archive Services expose 3 endpoints:

- **`/brr/v1/find`**
 - Accepts a data range, account number, SSN and status as search criteria and returns BRR IDs matching the given criteria.
- **`/brr/v1/get/{id}`**
 - Using the resulting data from the **`/brr/v1/find`** call, enter a single BRR ID and the corresponding BRR Details are returned in the response.
- **`/brr/v1/getList`**
 - Using the resulting data from the **`/brr/v1/find`** call, accepts a maximum of 50 BRR IDs and returns the processed BRRs in the same order.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).

API Services Process Specifications, continued

Archive
Services,
continued

ACDV Request Archive Services expose 4 endpoints:

- **`/acdvreq/v1/find`**
 - Accepts a date range, account number, SSN, status, Date Type, and returns ACDV Request Control Numbers matching the given search criteria.
- **`/acdvreq/v1/get/{acdvCtrlNum}`**
 - Using the resulting data from the `/acdvreq/v1/find`, enter a single ACDV Control Number and the corresponding ACDV Request is returned in the response.
 - There is no change to the ACDV status resulting from this request.
- **`/acdvreq/v1/getList`**
 - Using the resulting data from the `/acdvreq/v1/find`, enter up to 50 ACDV Control Numbers and the corresponding ACDV Requests are returned in the same order.
 - There is no change to the ACDV status resulting from this request.
- **`/acdvreq/v1/getImages`**
 - Using the resulting data from the `/acdvreq/v1/find`, enter the ACDV Control Number and the corresponding image for that ACDV is returned.
 - There is no change to the ACDV image status resulting from this request.

ACDV Response Archive Services expose 3 endpoints:

- **`/acdvresp/v1/find`**
 - Accepts a date range, account number, SSN, status, Date Type, and returns ACDV Response Control Numbers matching the given search criteria.
- **`/acdvresp/v1/get/{acdvCtrlNum}`**
 - Using the resulting data from the `/acdvresp/v1/find`, enter a single ACDV Control Number and the corresponding ACDV Response is returned in the response.
 - There is no change to the ACDV status resulting from this request.
- **`/acdvresp/v1/getList`**
 - Using the resulting data from the `/acdvresp/v1/find`, enter up to 50 ACDV Control Numbers and the corresponding ACDV Responses are returned in the same order.
 - There is no change to the ACDV status resulting from this request.

NOTE: It is recommended that users download the existing YAML files created in swagger to the web service client software (it is not required to use the existing YAML files to create your client, but it does make it much easier to create one by using it. You can always handcraft a JSON or XML to talk to API Services).



API Services Versioning

Application Versioning: API Services

Versioning in e-OSCAR Services is applicable in 2 areas. 1. API Services: This is what is called the service version. This is clearly visible in the end point of any service. For ex: In <https://api-uat.e-oscar-web.net/eoscar-services/aud/v1/get/86614191> ; V1 identifies what version of the service the DF is using to complete the call.

- a. This version will be specific to the area that is being served. For example: AUDs may run on v1 while ACDVs may run on v12.
- b. A new version will not be created/available unless the DF must change their code in order to get the latest updates or bug fixes.
- c. Whenever a bug fix or enhancement requires an update to request/response schema; e-OSCAR will create a new version.
- d. Service endpoints will not get updated with minor versions releases like 1.2; they will still be /aud/v1/get/xxxx. These updates will be communicated to the DFs, however they will not require schema updates and will not require code updates by the DF.

API services: When a new version of Services is released (for example v2); the previous version of the service (v1) will be available in production for up to 6 months from the release of v2. Once V2 is released there will be no more patch releases done to V1.

API Services Versioning, continued

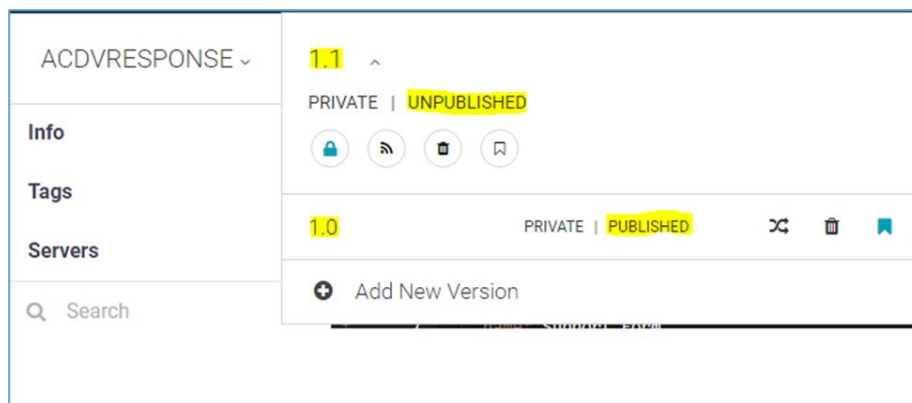
Application Versioning: API Documentation

API documentation (Swagger): This is the version of API documentation. Ideally this will correspond to the version of the service. For ex: In <https://app.swaggerhub.com/apis/New-Mgt-Services/ACDVRESPONSE/1.0> . 1.0 determines what version of service documentation the DF is looking at. This version will be specific to the area that is being served.

a. Latest version of AUDs may be on 1.3 and ACDVs may be on 2.5. Unlike API services, Swagger documentation can have minor versions.

1. 1.0 vs 1.1: What is the difference between 1.0 and 1.1 or 1.13? Each represent the same version of API with minor enhancements or bug fixes that will not require DFs to make any changes on their side. Every time a software release is complete by e-OSCAR that does not require DFs to make any changes, a new minor version of the documentation will be created. Notification will be provided to the DF advising of such minor version.

2. Published vs unpublished: Once the Production environment is updated with the latest software, the corresponding version of the documentation is published. For example: The latest version of documentation that is published is 1.4 meaning that the services are on v1 and the 4th minor release is in production now. Version 1.5 of documentation can be in an unpublished state and DFs may still access it. The code corresponding to the unpublished version will be available in non-prod environments, prior to going live in Production to allow DFs an opportunity to review and prepare for any potential impact.



API documentation: At least one published version (ex: 1.5) and one unpublished version (ex: 1.6) may be available for DFs in Swaggerhub. When a new version of API Services is released (ex: v2); the latest published version of v1 and any available published and unpublished versions of v2 are available.

NOTE: The latest published version (ex: v1) will be available for 6 months after the new version (ex: v2) has been published.