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 without the need to access the e-OSCAR User Interface.

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), submit Block Recission Requests (BRRs), 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
- Endpoint documentation
- Resource links

NOTE: Technical specification documents have been provided separately. Please refer to the technical specifications as needed in the SwaggerHub application. Please review the section within this document that outlines ***Access to Technical Specifications***.

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 Customer Success	Joel Strickland	jstrickland@newmgtservices.com	(904) 394-1479
Customer Success Manager	Christine Macdonald	cmacdonald@newmgtservices.com	(904) 289-8559
Onboarding Implementation Manager	T Edwards	tedwards@newmgtservices.com	(904)-479-1840

Contacts for Ongoing API Support Post Implementation

e-OSCAR API Support is available via our Jira ticketing platform. You may access the platform by using this link: https://newmgtservices.atlassian.net/servicedesk/customer/portal/2

When opening a support ticket, please be sure to include as many details as possible. Provide screenshots showing the body of the call you are making as well as the endpoint and URL you are calling. Please provide the error message that you are receiving.

It is recommended that Data Furnishers attempt the offending call in a neutral application (i.e. Postman) prior to opening the ticket in an effort to identify if this is an internal issue or a potential services defect. **NOTE** – By completing this step, Data Furnishers may find that this allows you to share the details required for triage purposes in a more efficient manner.

ALWAYS redact any PII in any evidence provided for triage.

API Implementation Stages

API Services Implementation Stages

Implementation of API Services occurs in the following stages:

- Acceptance of the API License Agreement
- Access to the technical documentation provided for review
- Billing of Onboarding fee and creation of Pricing Tier Analysis
- Data Furnisher development
- User Acceptance Testing
- Production Go Live
- Billing and Invoicing

Acceptance of API License Agreement

The Data Furnisher will be governed by the existing API User Agreement. This document is available here and will require acceptance. This usage agreement will govern the access to the documentation in SwaggerHub.

This document must be accepted prior to release of documentation via the Swaggerhub application.

API Implementation Stages

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 document is available here and will require acceptance. This usage agreement will govern the access to the documentation in Swaggerhub.
- Access to Swaggerhub is limited to no more than three users per Data Furnisher. The Data Furnisher must provide the name and email address of the individual(s) that will require access to the technical documentation. ***NOTE – An invitation email will be sent to the users from SwaggerHub. Users must register with SwaggerHub.***
- Once registered, 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: Swaggerhub is a third-party provider and is not managed by OLDE. Should you become locked out of the application, please contact Swaggerhub directly via their contact information in the application.

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 the Stage testing environment, the Data Furnisher will be invoiced for the API Onboarding fee or $2500.00.

OLDE will generate an invoice for the Onboarding fee for API Services and submit to the individual requesting access to the Stage environment for testing for payment.

Data Furnishers will be provided with a pricing tier analysis identifying the monthly API Service fee that will be charged on their invoice in addition to the existing per transaction fee that they are currently charged. This pricing sheet must be executed and returned to OLDE prior to launching into production.

Information on pricing tiers is available here.

The API pricing tier analysis is reviewed annually.

API Implementation Stages, continued

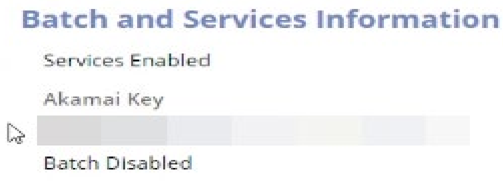
User Acceptance Testing

The User Acceptance Testing is required to confirm connectivity to the e-OSCAR database via the API Service. This is not an end-to-end test for your internal build.

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

OLDE will:

- Turn the API flag on in the Stage environment which will generate an API Key that must be used for each API call in Stage. The API key is unique to the Registration ID. This is located in the View/Update Registration page under the label of Akamai Key in the section titled Batch and Services Information and may be accessed by a user in Stage with a Registration Admin role.



Batch and Services Information

Services Enabled

Akamai Key

[redacted]

Batch Disabled

- Load data to the Stage testing environment so that the Data Furnisher may complete the “Get” calls for testing purposes. The Data Furnisher must test the “Submit” calls directly from their host system of record.

- Testing data requirements:
    - DFs will receive the following data to ensure successful connectivity testing for API Services:
        - Twice weekly data load inclusive of ACDVs from the DFs production inventory and scrubbed of all PII – The amount of data will fluctuate depending on the number of ACDVs available when completing the initial pull from production.
        - ACDVs with images that reflect at least one of each type of allowable image types (JPG, JPEG, GIF, PNG, TIFF, TIF, PDF) will be made available. They do not all need to be provided in each data load; however, all image types will need to be incorporated into the data load throughout the testing process.
        - AUD notifications for any AUDs submitted by the DF.
        - Block and DR notifications.

API Implementation Stages, continued

User Acceptance Testing, cont.

Data Furnisher must:

- Provide the external IP Address(es) for any user attempting to access the Stage environment via API. Please provide the full range of any potential addresses.
- Create an API User in the Stage testing environment.
- Complete testing to confirm connectivity to the API endpoints that will be used by the Data Furnisher in production.

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.

Stage testing 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 the Stage testing environment
- Multiple records have been successfully submitted with different indicators and response codes

Upon launching into production, Data Furnishers will be advised that the data loads will cease. Should a DF require data to continue to be loaded in the Stage environment post Go Live, it will be at a charge of $250.00 per month.

The frequency and volume of data will remain the same as it was for the testing period.

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. OLDE will not accommodate custom data requests.

NOTE – Given the way AUD and BRR transactions are processed, OLDE will not provide any data for testing AUD or BRR related endpoints. All AUDs and BRRs must be originated by the Data Furnisher in the Stage environment for testing.

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.

API Implementation Stages, continued

Implementation to Production

OLDE strongly recommends that Go Live is within 2 weeks after the successful conclusion of testing in the Stage environment. Please keep in mind launch calls for production take place Monday – Thursday. We will not launch to production on Fridays or holidays.

When the Data Furnisher is ready to launch to production, the Data Furnisher must open a ticket using this link: https://newmgtservices.atlassian.net/servicedesk/customer/portal/2

Please include:

- A list of attendees for the launch call
- Dates and times for launch call availability (please include your time zone)
- A final IP address(es) and or ranges that must be whitelisted prior to launching in to production

Go Live Worksession

Worksessions to launch the API Services into production are scheduled on a first come, first served basis.

- e-OSCAR will turn the API flag on in the production environment
- The API key will appear on your View Registration Page upon enablement (A Registration Admin is needed to view the API Key)
- Data Furnishers will need to create the API user in the e-OSCAR User Interface (Please have a resource on the call from your business unit that will be able to log in to the e-OSCAR User Interface with an Operator Admin role so that they may create the API user)
- Data Furnishers will need to complete the change password call in the production API environment
- Data Furnishers will need to complete the Auth Token call in the Production API environment

API Implementation Stages, continued

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 require a monthly fee in addition to the per transaction charge currently billed.

Current Quarterly bill Data Furnisher - Data Furnishers on quarterly billing cycles will **see a line item on their quarterly statement equal to a 3-month total of the Monthly API Services fee.** Based on the go live date of the API service, the fee will be added to the account on the following full quarter invoice. If the customer goes live in the middle of a billing quarter, the fee will be added the next full quarter invoice.

Current Monthly bill Data Furnisher – API Fee will appear on the next monthly invoice. **Monthly customers will not receive a USPS Mailed invoice, all invoices are sent by EMAIL** or accessible through the payment portal.

API Services Process Specifications

Access to API Environments

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

Stage:
Host Name
stage-api.e-oscar-web.net/prweb/api/
Base URL
https://stage-api.e-oscar-web.net/prweb/api/

Production:
Host Name
api.e-oscar-web.net/prweb/api
Base URL
https://api.e-oscar-web.net/prweb/api

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

NOTE: Creation of the API user and retrieval of the API Key is found in the e-OSCAR User Interface:
Stage: https://stage-www.e-oscar-web.net/
Production: https://identity.e-oscar-web.net/

API Services Process Specifications

Creation of API User in e-OSCAR

OLDE requires that an API User is created in the e-OSCAR User Interface prior to performing any API calls regardless of environment.

- The e-OSCAR Operator Manager for the Stage or Production environment will log in to the e-OSCAR User Interface and access the Operator Management Screen
- Select the button in the upper left-hand corner for Create Operator
- A window for Adding Operator will appear:
    - Select API User for Operator Type (This option will only appear once OLDE has enabled API Services for the Data Furnisher)
    - Complete Required fields (marked by a red asterisk)
    - A User Role will need to be selected:
        - API Retriever – 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. Because the API user has no access to the interactive application, there is no email sent to the API user to initiate a log in to the User Interface.

The API user must change the password using the following call: **/changePwd** within 24 hours of the creation of the User ID and temporary password.
The API user must then complete the Auth call to create a token for that user using the following call: **/auth/vX/authRequest**.

The API user in production must be a unique user ID and may not be a replica of the API user ID in Stage.

API Services Process Specifications, continued

API Key and Authorization Token

OLDE will require that the following information is included on all API Calls for API Services except for **/changePwd** and **/authRequest** (these two calls only require the API Key):

- **API Key** (available in the View/Update Registration Page of the e-OSCAR User Interface)
  - Ex. Http header: *x-api-key:* `f7c42aa3-0891-4ee4-9a6f-5594a3ea0c9c`
- **Authorization Token** (Generated by completing the /auth/vX/authRequest call)
  - Ex. Http header: *Authorization: Bearer*
    eyJ0eXAioiJKV1QiLChbGiOiJSUzI1NiJ9. eyJzdWIiOiJBUElVU0VSIiwiaXNzIjoibExWSH
    JMUlZCeTNvY2N4Y2NWeTIrNTFoR1FSY2U3VSIsImlh
    dCI6MTU4MjEzNTc2NCwianRpIjoiMzg3OTc4NiJ9.
    BKrnU8kcqsCsOoRfVsnzicsmNLNmHGzxGzTqzE9fwCOFbl_n
    X5zBYNfSr6zQ8EbWMSbDxHi5qcZri9TgVHyxfCYYDXSbg- CXannh1IslyoVsData
    Furnisher0eN4tl_CoREbsgQJmu99v7IYz0q8BmciadWOZk
    IWaIaC73p8qUPTnOuu7bck_8Wt5SJ786DsWy_hf
    lVeXNY5oEtzgsaTzgiUbobN_k7PawOpv1gao6- lqSiwckkztDrbqBgVDjJn1JGblvV7UE-
    UstAULbPDp WXQERuzpvMoyv9Ee05hlFeCurSPx46TKRUHC
    72RtAyTFG8wBVKkhC7ZmDpdmQ68ZmSSRZhIBxVA

API Services Process Specifications, continued

Authorization Services

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

Auth Services expose 2 endpoints:

- **/changePwd**
  - Accepts a Username, password (created in the e-OSCAR interactive application), new password and registration ID.
  - Password requirements are:
    - No repeated characters(i.e. Bookcase)
    - Minimum of 8 characters
    - Lowercase Letter
    - Uppercase Letter
    - At least one Number
    - At least one Symbol
    - No parts of username
    - Not one of the last 4 passwords
  - This is the first call that must be completed prior to completing any other calls within e-OSCAR Services.

- **/authRequest**
  - 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.
  - This call is not designed to be made before every call in API Services, but rather only made once for every 12 hours of continuous activity or after 30 minutes of inactivity. The Authorization token provided by this call will not change if the call is made multiple times within the timeframe described. Completion of this call prior to all subsequent calls may result in unnecessary errors that OLDE cannot triage as it is not being used as designed.

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.

The endpoints below are required to ingest the ACDV requests in Services with the intent to submit the ACDV Response via API Services:

/find

- Accepts a date range, account number, SSN, status, Date Type, subscriber code, account type, portfolio type, or dispute code 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 **/view/{acdvCtrlNum}** call prior to responding to the ACDV.

/view/{acdvCtrlNum}

- Using the resulting data from the **/find,** retrieves transaction data details and images for a single ACDV by ACDV Control Number. The status of the transaction must be PENDING-SENDREQUEST or PENDING-AWAITINGRESPONSE. If the status of the ACDV is PENDING-SENDREQUEST, it will be updated to PENDING-AWAITINGRESPONSE 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. Please note that this service is not required to submit the response. All validations are included in the Submit Services as well.

Validation Services are available for the following transaction types:

- ACDV Response
- AUD
- BRR

Validation Services expose 3 endpoints:

/acdvresp/vX/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/vX/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/vX/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).

The X in the endpoint identifies that a version number will be available at the time you utilize the Service. Please refer to the section in this document Application Versioning.

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:
/acdvresp/vX/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 submission in a single submit 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/vX/find** must have the details for the Control Number accessed by completing the **/acdvreq/vX/view/{acdvCtrlNum}** prior to completing the **/acdvresp/vX/submit** call.

/aud/vX/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 submission in a single submit call.
- All errors and successes will be returned at once.

/brr/vX/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 submission in a single submit 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).
The X in the endpoint identifies that a version number will be available at the time you utilize the Service. Please refer to the section in this document Application Versioning.

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/vX/find**
  - Accepts a date range, account number, SSN, status, Date Type and returns AUD Notification IDs matching the given search criteria.
- **/audnotif/vX/getList**
  - Using the resulting data from the **/audnotif/vX/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/vX/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/vX/getList**

Using the resulting data from the **/notification/vX/find**, accepts a maximum of 50 Block/DR Control Numbers and returns the Block/DR Notifications in the same order. When the notification is viewed using the GetList call, the status changes from PENDING-REVIEW for **DR NOTIFICATON ID, DR Tradeline and Block NOTIFICATIONID** and **Block Tradeline** to:
RESOLVED-REVIEWED for **DR NOTIFICATON ID, DR Tradeline and Block NOTIFICATIONID**
and RESOLVED-ACCEPTED for **Block Tradeline**

- **/notification/vX/getImages**
  - Using the resulting data from the **/notification/vX/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).
The X in the endpoint identifies that a version number will be available at the time you utilize the Service. Please refer to the section in this document Application Versioning.

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/vX/find**
    - Accepts a data range, account number, SSN and status as search criteria and returns AUD IDs matching the given criteria.
- **/aud/vX/get/{id}**
    - Using the resulting data from the **/aud/vX/find** call, enter a single AUD ID and the corresponding AUD Details are returned in the response.
- **/aud/vX/getList**
    - Using the resulting data from the **/aud/vX/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/vX/find**
    - Accepts a data range, account number, SSN and status as search criteria and returns BRR IDs matching the given criteria.
- **/brr/vX/get/{id}**
    - Using the resulting data from the **/brr/vX/find** call, enter a single BRR ID and the corresponding BRR Details are returned in the response.
- **/brr/vX/getList**
    - Using the resulting data from the **/brr/vX/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).
The X in the endpoint identifies that a version number will be available at the time you utilize the Service. Please refer to the section in this document Application Versioning.

Archive Services, continued

<u>ACDV Request Archive Services expose 4 endpoints</u>:

- **/acdvreq/vX/find**
    - Accepts a date range, account number, SSN, status, Date Type, and returns ACDV Request Control Numbers matching the given search criteria.
- **/acdvreq/vX/get/{acdvCtrlNum}**
    - Using the resulting data from the **/acdvreq/vX/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/vX/getList**
    - Using the resulting data from the **/acdvreq/vX/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/vX/getImages**
    - Using the resulting data from the **/acdvreq/vX/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.

<u>ACDV Response Archive Services expose 3 endpoints</u>:

- **/acdvresp/vX/find**
    - Accepts a date range, account number, SSN, status, Date Type, and returns ACDV Response Control Numbers matching the given search criteria.
- **/acdvresp/vX/get/{acdvCtrlNum}**
    - Using the resulting data from the **/acdvresp/vX/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/vX/getList**
    - Using the resulting data from the **/acdvresp/vX/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).
The X in the endpoint identifies that a version number will be available at the time you utilize the Service. Please refer to the section in this document Application Versioning.

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://stage-api.e-oscar-web.net/prweb/api/aud/v1/get/86614191 ; V1 identifies what version of the service the Data Furnisher 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 Data Furnisher 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 Data Furnishers, however they will not require schema updates and will not require code updates by the Data Furnisher.

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

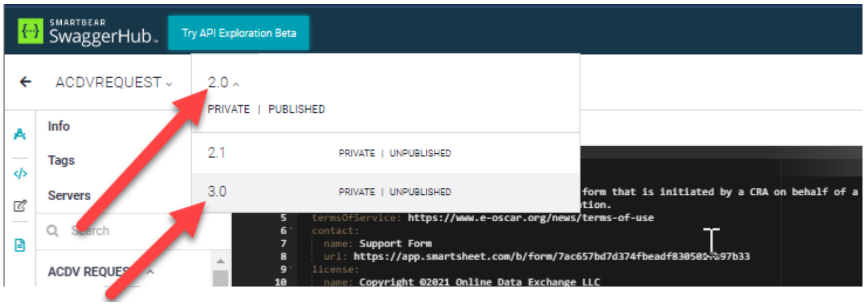
Application Versioning: API Documentation

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 Data Furnisher 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 DATA FURNISHERs to make any changes on their side. Every time a software release is complete by e-OSCAR that does not require Data Furnishers to make any changes, a new minor version of the documentation will be created. Notification will be provided to the Data Furnisher 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 1.4meaning 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 Data Furnishers 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 Data Furnishers an opportunity to review and prepare for any potential impact.



2.0 PRIVATE | PUBLISHED
2.1 PRIVATE | UNPUBLISHED
3.0 PRIVATE | UNPUBLISHED

```
form that is initiated by a CRA on behalf of a
tion.
termsOfService: https://www.e-oscar.org/news/terms-of-use
contact:
  name: Support Form
  url: https://app.smartsheet.com/b/form/7ac657bd7d374fbeadf830502497b33
license:
  name: Copyright ©2021 Online Data Exchange LLC
```

API documentation: At least one published version (ex: 1.5) and one unpublished version (ex: 1.6) may be available for Data Furnishers 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.

Resource Links

Resource Links	• FAQs- https://e2020.my.site.com/support/s/ • Auth call video- https://vimeo.com/810907759 • Create user video- https://vimeo.com/810907486 • API Ticket Portal- https://newmgtservices.atlassian.net/servicedesk/customer/portal/2

Status Definition

ACDV Statuses:

PENDING-SENDREQUEST = An ACDV Request that has been delivered to the Data Furnisher

PENDING-AWAITINGRESPONSE = An ACDV Request that has been opened by the Data Furnisher

RESOLVED-SENDINGTOAGENCY = An ACDV response that has been successfully submitted

RESOLVED-NORESPONSEPROVIDED = An ACDV that has not had a response submitted by the Response Due Date

RESOLVED-RETURNEDTOAGENCY = An ACDV with a submitted response that has been packaged and sent to the CRA

AUD and BRR Statuses:

RESOLVED-SENDINGTOAGENCY = An AUD/BRR that has been Submitted by the DF

RESOLVED-RETURNEDTOAGENCY = An AUD/BRR that has been received by the CRA

AUD Notification Statuses:

PENDING-REVIEW = An AUD Notification has been sent by the CRA to the DF for Review

RESOLVED-REVIEWED = DF affirms the review. (This is an action that may only be completed in the UI by a user with an AUD Notification Viewer (AN) role. Not via API)

Notification (DR & Block) Statuses:

PENDING-REVIEW = A Notification has been sent by the CRA to the DF for Review

RESOLVED-ACCEPTED = DF Accepted the Block TL Notification (this is an action that may only take place in the UI with a Block Notification Viewer (BV) role. This action is not permitted in the API)

RESOLVED-ACKNOWLEDGED = DF Acknowledged the Block TL(this is an action that may only take place in the UI with a Block notification Viewer (BV) role. This action is not permitted in the API)

RESOLVED-RECISSIONED = DF Submitted a BRR for the Block Tradeline

RESOLVED-REVIEWED = DF affirms review (only applicable for Block Notification (ID) and DR Notification (ID and Tradeline). This is an action that may only take place in the UI with either a DR Notification Viewer (DV) or a Block notification Viewer (BV) role. Not via API.)