

# FOIAXpress

## API User Manual

v11.10.0

January 2025



OPEXUSTECH.COM

© AINS LLC, 2025

# FOIAExpress v11.10.0 API User Manual

## Notice of Rights

Copyright © 2025, OPEXUS, LLC d/b/a OPEXUS. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of the publisher: OPEXUS, LLC. For information on obtaining permission for reprints and excerpts, contact [info@opexustech.com](mailto:info@opexustech.com).

Additionally, all copyrights, confidential information, patents, design rights and all other intellectual property rights of whatsoever nature contained herein are, and shall remain, the sole and exclusive property of the publisher.

## Notice of Liability

The information in this publication is believed to be accurate and reliable. However, the information is distributed by the publisher (OPEXUS, LLC.) on an “As Is” basis without warranty for its use, or for any infringements of patents or other rights of third parties resulting from its use.

While every precaution has been taken in the preparation of this publication, neither the author (or authors) nor the publisher will have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused, directly or indirectly, by the information contained in this publication or by the computer software and hardware products described in it.

## Notice of Trademarks

The publisher’s company name, company logo, company patents, and company proprietary products are trademarks or registered trademarks of the publisher: OPEXUS, LLC. All other trademarks or registered trademarks are the property of their respective owners.

## Non-Disclosure Statement

This document’s contents are confidential and proprietary to OPEXUS, LLC. This document cannot be released publicly or outside the purchasing agency without prior written permission from OPEXUS, LLC.

Images in this manual are used as examples and may contain data and versioning that may not be consistent with your version of the application or information in your environment.

## Additional Notice

Information in this documentation is subject to change without notice and does not represent a commitment on the part of OPEXUS, LLC.

Notwithstanding any of the foregoing, if this document was produced as a Deliverable or other work for hire under a contract on behalf of a U.S. Government end user, the terms and conditions of that contract shall apply in the event of a conflict



# Contents

|       |                                             |    |
|-------|---------------------------------------------|----|
| 1     | FOIAXpress API Introduction .....           | 6  |
| 1.1   | About the API .....                         | 6  |
| 1.2   | About this Manual.....                      | 6  |
| 2     | Accessing the FOIAXpress API.....           | 7  |
| 2.1   | Authentication using Access Token Tool..... | 7  |
| 2.2   | How to Access the FOIAXpress API.....       | 7  |
| 3     | Request Methods.....                        | 8  |
| 3.1   | Get Request Status .....                    | 8  |
| 3.2   | Get Request .....                           | 9  |
| 3.3   | Get Requests.....                           | 10 |
| 3.3.1 | Parameters .....                            | 11 |
| 3.3.2 | Using the Get Requests Method .....         | 12 |
| 3.4   | Requests Count .....                        | 14 |
| 3.4.1 | Parameters .....                            | 14 |
| 3.4.2 | Output.....                                 | 14 |
| 3.5   | Request/Updaterequest .....                 | 15 |
| 3.5.1 | Parameters .....                            | 15 |
| 3.5.2 | Output.....                                 | 20 |
| 3.6   | Request/Createrequest.....                  | 21 |
| 3.6.1 | Parameters .....                            | 21 |
| 3.6.2 | Output.....                                 | 25 |
| 4     | Requesters Methods .....                    | 27 |
| 4.1   | Requesters/get/requesterId/LocaleId.....    | 27 |
| 4.1.1 | Parameters .....                            | 27 |
| 4.1.2 | Output.....                                 | 27 |
| 4.2   | Requesters/create .....                     | 29 |



## Contents

|       |                                                               |    |
|-------|---------------------------------------------------------------|----|
| 4.2.1 | Parameters .....                                              | 29 |
| 4.2.2 | Output.....                                                   | 31 |
| 4.3   | Requesters/Update.....                                        | 31 |
| 4.3.1 | Parameters .....                                              | 31 |
| 4.3.2 | Output.....                                                   | 33 |
| 5     | Audit Methods .....                                           | 34 |
| 5.1   | User Actions.....                                             | 34 |
| 5.2   | User Logins.....                                              | 35 |
| 6     | EmailService API .....                                        | 38 |
| 6.1   | Mail/api/getmail.....                                         | 38 |
| 6.1.1 | Parameters .....                                              | 38 |
| 6.1.2 | Output.....                                                   | 38 |
| 6.2   | Mail/api/getmailcontent.....                                  | 39 |
| 6.2.1 | Parameters .....                                              | 39 |
| 6.2.2 | Output.....                                                   | 40 |
| 7     | EntityRecognition API.....                                    | 41 |
| 7.1   | Comprehend/api/detectentities.....                            | 41 |
| 7.1.1 | Parameters .....                                              | 41 |
| 7.1.2 | Output.....                                                   | 41 |
| 7.2   | Comprehend/api/detectentities.....                            | 42 |
| 7.2.1 | Parameters .....                                              | 42 |
| 7.2.2 | Output.....                                                   | 42 |
| 8     | MediaRedactions API .....                                     | 44 |
| 8.1   | Mediareductions/apiex/mediafile/getsignedurl .....            | 44 |
| 8.1.1 | Parameters .....                                              | 44 |
| 8.2   | Mediareductions/apiex/mediafile/getsignedurl/signedToken..... | 45 |
| 8.2.1 | Parameters .....                                              | 45 |
| 8.2.2 | Output.....                                                   | 46 |



## Contents

|       |                                |    |
|-------|--------------------------------|----|
| 9     | Reports API.....               | 47 |
| 9.1   | Report/api/isaccuratexml ..... | 47 |
| 9.1.1 | Parameters .....               | 47 |
| 9.1.2 | Output.....                    | 47 |



# 1 FOIAXpress API Introduction

## 1.1 About the API

We've created the FOIAXpress API to allow for integration with external data monitoring applications. This is a standalone API that is included in the FOIAXpress installation package. This API must be installed separately from the FOIAXpress application, and users of the API must be authenticated application users with permissions to access the data being queried.

## 1.2 About this Manual

This manual covers instructions on accessing and using the included API methods. The following major topics are included:

- *Accessing the FOIAXpress API:* Token creation, accessing the API, and viewing logs
- *Request Methods:* Methods to query Request data
- *Audit Methods:* Methods to query user actions and logins



## 2 Accessing the FOIAXpress API

### 2.1 Authentication using Access Token Tool

Each API user must be authenticated before they are able to use the API. Your administrator has an Access Token Tool which can be used to create a token for you as an API user. These tokens are time limited and must be provisioned again after expiration. Please contact your administrator to request an access token.

### 2.2 How to Access the FOIAXpress API

To verify access to the API, navigate to the following link in your browser:

<https://EnterAgencyDNS/GatewayAPI/swagger>

**Note: Make sure to replace EnterAgencyDNS with your agency's DNS.**

You can also check access to Request Methods, Audit Methods, and more via the following links:

|                    |                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Request Methods    | <a href="https://EnterAgencyDNS/GatewayAPI/Requests/swagger/">https://EnterAgencyDNS/GatewayAPI/Requests/swagger/</a>                   |
| Audit Methods      | <a href="https://EnterAgencyDNS/GatewayAPI/AuditLog/swagger/">https://EnterAgencyDNS/GatewayAPI/AuditLog/swagger/</a>                   |
| Reports            | <a href="https://EnterAgencyDNS/GatewayAPI/Reports/swagger/">https://EnterAgencyDNS/GatewayAPI/Reports/swagger/</a>                     |
| Media Redactions   | <a href="https://EnterAgencyDNS/GatewayAPI/MediaRedactions/swagger/">https://EnterAgencyDNS/GatewayAPI/MediaRedactions/swagger/</a>     |
| Jobs               | <a href="https://EnterAgencyDNS/GatewayAPI/Jobs/swagger/">https://EnterAgencyDNS/GatewayAPI/Jobs/swagger/</a>                           |
| Entity Recognition | <a href="https://EnterAgencyDNS/GatewayAPI/EntityRecognition/swagger/">https://EnterAgencyDNS/GatewayAPI/EntityRecognition/swagger/</a> |
| Email Service      | <a href="https://EnterAgencyDNS/GatewayAPI/emailservice/swagger/">https://EnterAgencyDNS/GatewayAPI/emailservice/swagger/</a>           |

**Note: You must have Request Type and Audit permissions within the application to use the Request and Audit methods.**



## 3 Request Methods

The following sections provide details and steps to use the Request methods to query request data.

### 3.1 Get Request Status

You can use the Get Request Status method to retrieve the status of a specific request. This requires the exact ID of the request you are querying.

**Note: You must have Request Type permissions within the application to access this method.**

To use the Get Request Status function:

1. Under *Requests*, locate the *id string* field and enter the Request ID exactly as it appears in the application, then click **Execute**:

The screenshot shows the 'Requests' interface. At the top, there's a 'GET' method and a URL template '/requests/{id}/status'. Below this is a 'Parameters' section with a table. The table has two columns: 'Name' and 'Description'. The first row has 'id \* required' in the 'Name' column and 'string (path)' in the 'Description' column. The value '24-FOI-00001' is entered in the 'id' field. A red arrow points to this field. Below the table is a blue 'Execute' button, which is circled in red, and a 'Clear' button. At the bottom, there's a 'Responses' section.

2. After clicking **Execute**, the request data appears in the *Response Body* as shown below.

The screenshot shows the 'Response Body' after clicking 'Execute'. It displays the 'Request URL' as 'https://localhost:7003/requests/24-FOI-00001/status'. Below this is the 'Server response' section. The 'Code' is '200' and the 'Details' are 'Assigned'. The 'Response body' is highlighted with a red box. To the right of the response body are two red circular buttons labeled 'A' and 'B', and a 'Download' button. Below the response body is the 'Response headers' section, which lists: 'api-supported-versions: 1.0', 'content-type: text/plain; charset=utf-8', 'date: Wed, 18 Oct 2023 16:22:41 GMT', and 'server: Kestrel'. At the bottom, there's a 'Responses' table with columns 'Code', 'Description', and 'Links'. The first row shows '200' for 'Code' and 'Success' for 'Description'. The 'Links' column shows 'No links'.

3. There are options to (A) **Copy** this to your clipboard, or (B) **Download** the response body.





To use this function via URL, use the following string with the application URL in place of <fx-api-url>, and replacing the {id} with the request ID you are querying:

<fx-api-url>/requests/{id}/status

An example URL is shown below:

#### Request URL

```
https://localhost:7003/requests/24-FOI-00001/status
```

## 3.2 Get Request

You can use the Get Request API to query for data on a specific request. This requires the exact Request ID.

**Note: You must have Request Type permissions within the application to access this method.**

This method returns the following Request details:

- *visibleRequestID*: ID of the request
- *requestStatus*: Status of the request
- *receivedDate*: Date when the request was received.
- *closedDate*: Date when the request was closed. It is null if request is active
- *requestTypeName*: Request Type
- *actionOffice*: Action Office
- *deliveryMode*: Mode of delivery of response to the requester
- *feeWaiverStatus*: Fee waiver status of the request
- *paymentStatus*: Payment Status of the request
- *priority*: Priority of the request
- *receivedMode*: How the request was received
- *requesterCategory*: Category to which the requester belongs to
- *reviewStatus*: Review status of the request
- *targetDate*: Target date to close the request

To use the Get Request Status function:



1. Under *Requests*, locate the *id string* field and enter the Request ID exactly as it appears in the application, then click **Execute**:

GET

/requests/{id}

Parameters

Cancel

| Name                                             | Description                                                           |
|--------------------------------------------------|-----------------------------------------------------------------------|
| id * required<br>string<br><small>(path)</small> | <div>24-FOI-00001</div> <div><div>Execute</div><div>Clear</div></div> |

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7003/requests/24-FOI-00001' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIodHRwOiIvZvd3d3lnczlm9yZyBMDAxL2A0L3htbGRzaWctbnV9ZSNoZWYtJLNxOYTUxMIsInR5cCI6IkpXVCJ9.eyJodHRwOiIvc2NoZk1hcyc54bWxzczFwcm9yZy93cyBMDAxL2A0L3htbGRzaWctbnV9ZSNoZWYtJLNxOYTUxMIsInR5cCI6IkpXVCJ9.eyJodHRwOiIvc2NoZk1hcyc54bWxzczFwcm9yZy93cyBMDAxL2A0L3htbGRzaWctbnV9ZSNoZWYtJLNxOYTUxMIsInR5cCI6IkpXVCJ9.'
```

Request URL

https://localhost:7003/requests/24-FOI-00001

2. After clicking **Execute**, the request data appears in the *Response Body* as shown below.

Request URL

https://localhost:7003/requests/24-FOI-00001

Server response

| Code | Details                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200  | <div><div>Response body</div><div><pre>{   "visibleRequestID": "24-FOI-00001",   "requestStatus": "Assigned",   "receivedDate": "2023-10-16T13:37:05.54",   "closedDate": null,   "requestTypeName": "FOIA",   "actionOffice": "HQ",   "deliveryMode": null,   "feeWaiverStatus": "Fee Waiver Not Requested",   "paymentStatus": "No Charges",   "priority": "Normal",   "receivedMode": null,   "requesterCategory": "Commercial Organization",   "reviewStatus": null,   "targetDate": "2023-11-14T00:00:00" }</pre></div><div><div>A</div><div>B</div><div>Download</div></div></div> |

Response headers

```
api-supported-versions: 1.0
content-type: application/json; charset=utf-8
date: Wed, 18 Oct 2023 16:24:25 GMT
server: Kestrel
```

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | Success     | No links |

3. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body.

### 3.3 Get Requests

You can use the [Get Requests API](#) to query for requests with matching attributes.

**Note:** You must have Request Type permissions within the application to access this method.



### 3.3.1 Parameters

The parameters available within the Get Requests method are outlined in the following tables.

Request Status parameter options:

| Parameter | Attribute                       |
|-----------|---------------------------------|
| Need      | On Hold-Need Info/Clarification |
| CWithheld | On Hold-Fee Related             |
| DAdded    | Documents Added                 |
| DD        | Documents Delivered             |
| DF        | Delivery Failed                 |
| DP        | Delivery Pending                |
| Amended   | Amended                         |
| Assigned  | Assigned                        |
| Canceled  | Canceled                        |
| Closed    | Closed                          |
| Completed | Disposition Accepted            |
| Received  | Received                        |
| Screened  | Perfected                       |



| Parameter   | Attribute                     |
|-------------|-------------------------------|
| ReqforDocs  | Request for Docs Sent         |
| RvwPending  | Review Pending                |
| RvwAprvd    | Review Approved               |
| RvwDisAprvd | Review Disapproved            |
| DAddRvwLog  | Documents Added to Review Log |

Request Type Name Parameters:

| Parameter                                                          | Example            |
|--------------------------------------------------------------------|--------------------|
| Request Type name, as found in Administration > Request Management | "FOIA", "FOIA,APP" |

Action Office parameter options:

| Parameter                                                     | Example       |
|---------------------------------------------------------------|---------------|
| Office Codes, as found in Administration > Organization Setup | "HQ", "HQ,DC" |

### 3.3.2 Using the Get Requests Method

Follow the steps below to use the Get Requests method:

1. Under *Post/requests*, use the *Request body* to customize the query. Beside each parameter you are using, enter the Request data that you would like returned in the results:



POST

/requests

Parameters

No parameters

Request body

```

{
  "pageSize": 100,
  "pageNumber": 1,
  "visibleRequestIDs": [
    "24-FOI-00001", "24-FOI-00002"
  ],
  "recievedDateFrom": "2023-01-01",
  "recievedDateTo": "2024-01-01",
  "closedDateFrom": "2023-01-01",
  "closedDateTo": "2023-01-01",
  "requestStatus": [
    "Assigned", "Closed"
  ],
  "requestTypeName": [
    "FOIA"
  ],
  "actionOffice": [
    "HQ"
  ]
}

```

- The `pageSize` and `pageNumber` fields are required and configure the formatting for the returned results. `pageSize` indicates the number of items returned per page, and `pageNumber` dictates the current page number.
- If you are using the `receivedDateFrom` or `closedDateFrom` fields, you must also use the `receivedDateTo` and `closedDateTo` fields, respectively.

**Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.**

- For details on each parameter, see the *Get Request Parameters* section. Remove any parameters you are not using in the query.
- After executing the method, the output returns all Requests matching the parameters used in the Request Body.



## 3.4 Requests Count

This method returns the total number of requests found for the requested ID.

### 3.4.1 Parameters

```
{  
  "visibleRequestIDs": [  
    "string"  
  ],  
  "receivedDateFrom": "2024-04-09T15:56:23.647Z",  
  "receivedDateTo": "2024-04-09T15:56:23.647Z",  
  "closedDateFrom": "2024-04-09T15:56:23.647Z",  
  "closedDateTo": "2024-04-09T15:56:23.647Z",  
  "requestStatus": [  
    "string"  
  ],  
  "requestTypeName": [  
    "string"  
  ],  
  "actionOffice": [  
    "string"  
  ]  
}
```

### 3.4.2 Output

The number of requests that match with the filter



**POST** /requests/count

**Parameters**

No parameters

**Request body**

```

{
  "visibleRequestIDs": [
    "string"
  ],
  "receivedDateFrom": "2024-04-08T19:08:35.454Z",
  "receivedDateTo": "2024-04-08T19:08:35.454Z",
  "closedDateFrom": "2024-04-08T19:08:35.454Z",
  "closedDateTo": "2024-04-08T19:08:35.454Z",
  "requestStatus": [
    "string"
  ],
  "requestTypeName": [
    "string"
  ],
  "actionOffice": [
    "string"
  ]
}

```

Execute

## 3.5 Request/Updaterequest

Update a request with the information that the user sent via JSON.

### 3.5.1 Parameters

```

{
  "description": "string",
  "requestedDate": "2024-04-09T15:57:02.503Z",
  "receivedDate": "2024-04-09T15:57:02.503Z",
  "officeCode": "string",
  "requesterFirstName": "string",
  "requesterLastName": "string",
  "requesterEmail": "string",

```



```
"requesterCountry": "string",
"requesterZipCode": "string",
"requestUserEmail": "string",
"priority": "string",
"deliveryMode": "string",
"receiveMode": "string",
"requesterCategory": "string",
"address1": "string",
"address2": "string",
"city": "string",
"state": "string",
"homePhone": "string",
"workPhone": "string",
"requestCustomFields": [
  {
    "fieldName": "string",
    "dataType": "string",
    "value": "string"
  }
],
"requestDescriptionAttachments": [
  {
    "contentType": "string",
    "fileData": "string",
    "fileName": "string",
    "fileSize": "string"
  }
]
```





```
],  
"expediteRequested": true,  
"expediteDescription": "string",  
"expediteStartDate": "2024-04-09T15:57:02.503Z",  
"expediteDescriptionAttachments": [  
  {  
    "contentType": "string",  
    "fileData": "string",  
    "fileName": "string",  
    "fileSize": "string"  
  }  
],  
"feeWaiverRequested": true,  
"feeWaiverDescription": "string",  
"feeWaiverStart": "2024-04-09T15:57:02.503Z",  
"feeWaiverDescriptionAttachments": [  
  {  
    "contentType": "string",  
    "fileData": "string",  
    "fileName": "string",  
    "fileSize": "string"  
  }  
],  
"linkedRequests": [  
  {  
    "id": 0,  
    "visibleRequestID": "string",
```



```

    "requestStatus": "string",
    "requestDisplayStatus": "string",
    "linkedRequestId": 0,
    "linkedAction": 0
  }
],
"sentToCommissioner": {
  "requestId": 0,
  "isSentToCommissioner": "string",
  "sentDate": "2024-04-09T15:57:02.503Z",
  "responseDate": "2024-04-09T15:57:02.503Z",
  "approvalStatus": "string",
  "sentReason": "string",
  "section32Notice": "string",
  "subsectionCeased": "string",
  "section35Formal": "string",
  "section37Reports": "string",
  "recommendations": "string",
  "orders": "string",
  "paSection31": "string",
  "paSection33": "string",
  "paSection35": "string",
  "paCourtAction": "string"
},
"requestType": "string",
"visibleRequestId": "string",
"feeWaiverEnd": "2024-04-09T15:57:02.503Z",

```



```
"expediteEndDate": "2024-04-09T15:57:02.503Z",  
"lastPerfectedDate": "2024-04-09T15:57:02.503Z",  
"perfectedDate": "2024-04-09T15:57:02.503Z",  
"originalReceivedDate": "2024-04-09T15:57:02.503Z",  
"shippingAddress1": "string",  
"shippingAddress2": "string",  
"shippingCity": "string",  
"shippingState": "string",  
"shippingCountry": "string",  
"shippingZipCode": "string",  
"billingAddress1": "string",  
"billingAddress2": "string",  
"billingCity": "string",  
"billingState": "string",  
"billingCountry": "string",  
"billingZipCode": "string"  
}
```



## 3.5.2 Output

**POST** /requests/updaterequest

**Parameters**

No parameters

**Request body**

**Example Value** | **Schema**

```
{
  "description": "string",
  "requestedDate": "2024-04-08T19:09:33.448Z",
  "receivedDate": "2024-04-08T19:09:33.448Z",
  "officeCode": "string",
  "requesterFirstName": "string",
  "requesterLastName": "string",
  "requesterEmail": "string",
  "requesterCountry": "string",
  "requesterZipCode": "string",
  "requestUserEmail": "string",
  "priority": "string",
  "deliveryMode": "string",
  "receiveMode": "string",
  "requesterCategory": "string",
  "address1": "string",
  "address2": "string",
  "city": "string",
  "state": "string",
  "homePhone": "string",
  "workPhone": "string",
  "requestCustomFields": [
    {
      "fieldName": "string",
      "dataType": "string",
      "value": "string"
    }
  ]
},
```



**POST** /requests/updaterequest

**Parameters**

No parameters

**Request body**

Example Value | Schema

```

{
  "requestDescriptionAttachments": [
    {
      "contentType": "string",
      "fileData": "string",
      "fileName": "string",
      "fileSize": "string"
    }
  ],
  "expediteRequested": true,
  "expediteDescription": "string",
  "expediteStartDate": "2024-04-08T19:09:33.448Z",
  "expediteDescriptionAttachments": [
    {
      "contentType": "string",
      "fileData": "string",
      "fileName": "string",
      "fileSize": "string"
    }
  ],
  "feeWaiverRequested": true,
  "feeWaiverDescription": "string",
  "feeWaiverStart": "2024-04-08T19:09:33.448Z",
  "feeWaiverDescriptionAttachments": [
    {
      "contentType": "string",
      "fileData": "string",
      "fileName": "string",
      "fileSize": "string"
    }
  ]
}

```

## 3.6 Request/Creatererequest

Create a request with the information that the user has specified.

### 3.6.1 Parameters

```

{
  "requestType": "string",
  "description": "string",
  "requestedDate": "2024-04-09T16:01:59.757Z",
  "receivedDate": "2024-04-09T16:01:59.757Z",
  "officeCode": "string",
  "requesterFirstName": "string",

```



```
"requesterLastName": "string",
"requesterEmail": "string",
"requesterCountry": "string",
"requesterZipCode": "string",
"requestUserEmail": "string",
"priority": "string",
"deliveryMode": "string",
"receiveMode": "string",
"requesterCategory": "string",
"address1": "string",
"address2": "string",
"city": "string",
"state": "string",
"homePhone": "string",
"workPhone": "string",
"requestCustomFields": [
  {
    "fieldName": "string",
    "dataType": "string",
    "value": "string"
  }
],
"requestDescriptionAttachments": [
  {
    "contentType": "string",
    "fileData": "string",
    "fileName": "string",
```



```
"fileSize": "string"
}
],
"expediteRequested": true,
"expediteDescription": "string",
"expediteStartDate": "2024-04-09T16:01:59.757Z",
"expediteDescriptionAttachments": [
{
  "contentType": "string",
  "fileData": "string",
  "fileName": "string",
  "fileSize": "string"
}
],
"feeWaiverRequested": true,
"feeWaiverDescription": "string",
"feeWaiverStart": "2024-04-09T16:01:59.757Z",
"feeWaiverDescriptionAttachments": [
{
  "contentType": "string",
  "fileData": "string",
  "fileName": "string",
  "fileSize": "string"
}
],
"linkedRequests": [
{
```



```

    "id": 0,
    "visibleRequestID": "string",
    "requestStatus": "string",
    "requestDisplayStatus": "string",
    "linkedRequestId": 0,
    "linkedAction": 0
  }
],
"sentToCommissioner": {
  "requestId": 0,
  "isSentToCommissioner": "string",
  "sentDate": "2024-04-09T16:01:59.757Z",
  "responseDate": "2024-04-09T16:01:59.757Z",
  "approvalStatus": "string",
  "sentReason": "string",
  "section32Notice": "string",
  "subsectionCeased": "string",
  "section35Formal": "string",
  "section37Reports": "string",
  "recommendations": "string",
  "orders": "string",
  "paSection31": "string",
  "paSection33": "string",
  "paSection35": "string",
  "paCourtAction": "string"
}
}

```





## 3.6.2 Output

**POST** /requests/createrequest

Parameters

No parameters

Request body

Example Value | Schema

```
{
  "requestType": "string",
  "description": "string",
  "requestedDate": "2024-04-08T19:11:45.730Z",
  "receivedDate": "2024-04-08T19:11:45.730Z",
  "officeCode": "string",
  "requesterFirstName": "string",
  "requesterLastName": "string",
  "requesterEmail": "string",
  "requesterCountry": "string",
  "requesterZipCode": "string",
  "requestUserEmail": "string",
  "priority": "string",
  "deliveryMode": "string",
  "receiveMode": "string",
  "requesterCategory": "string",
  "address1": "string",
  "address2": "string",
  "city": "string",
  "state": "string",
  "homePhone": "string",
  "workPhone": "string",
  "requestCustomFields": [
    {
      "fieldName": "string",
      "dataType": "string",
      "value": "string"
    }
  ]
}
```



## Request Methods

**POST** /requests/updaterequest

### Parameters

No parameters

Request body

Example Value | Schema

```
{
  "section37Reports": "string",
  "recommendations": "string",
  "orders": "string",
  "paSection31": "string",
  "paSection33": "string",
  "paSection35": "string",
  "paCourtAction": "string"
},
"requestType": "string",
"visibleRequestId": "string",
"feeWaiverEnd": "2024-04-08T19:09:33.448Z",
"expediteEndDate": "2024-04-08T19:09:33.448Z",
"lastPerfectedDate": "2024-04-08T19:09:33.448Z",
"perfectedDate": "2024-04-08T19:09:33.448Z",
"originalReceivedDate": "2024-04-08T19:09:33.448Z",
"shippingAddress1": "string",
"shippingAddress2": "string",
"shippingCity": "string",
"shippingState": "string",
"shippingCountry": "string",
"shippingZipCode": "string",
"billingAddress1": "string",
"billingAddress2": "string",
"billingCity": "string",
"billingState": "string",
"billingCountry": "string",
"billingZipCode": "string"
}
```



## 4 Requesters Methods

### 4.1 Requesters/get/requesterId/LocaleId

Retrieve the requester's information based on the ID and the requested language.

### Requesters

**GET** /requesters/get/{requesterID}/{localeID}

**Parameters**

| Name                                                        | Description                              |
|-------------------------------------------------------------|------------------------------------------|
| <b>requesterID</b> * required<br>integer(\$int32)<br>(path) | <input type="text" value="requesterID"/> |
| <b>localeID</b> * required<br>integer(\$int32)<br>(path)    | <input type="text" value="localeID"/>    |

#### 4.1.1 Parameters

RequesterID: Requester ID

localeID: Locale ID(language)

#### 4.1.2 Output

```
{  
  "requesterTypeId": 0, requester type  
  "prefix": 0, prefix type  
  "suffix": 0, suffix type  
  "lastName": "Default", requester last name  
  "firstName": "Default", requester first name  
  "middleName": "", requester middle name  
  "jobTitle": "", requester title job
```



"workPhone1": "", requester phone  
"workPhone2": "", requester phone  
"homePhone": "", requester phone  
"mobile": "", requester phone  
"fax": "", requester fax  
"email": [default@example.com](mailto:default@example.com), requester email  
"company": "", requester company  
"login": "default", requester login  
"notes": null, requester notes  
"loginActive": true, status of requester login active or not  
"isLoginActive": true, status of requester login active or not  
"deactiveReason": 0, reason of being deactivate the login  
"createdById": 0, requester created by  
"modifiedById": 0, requester modified by  
"address1": "Default Address", requester address  
"address2": "", requester address  
"city": "Default City", requester city  
"stateId": 0, requester state  
"countryId": 0, requester country  
"zipCode": "00000", requester zip code  
"createdBy": "Default", requester created by name  
"modifiedBy": "Default", requester modified by name  
"delinquent": false, requester is delinquent  
"isDelinquent": false, requester is delinquent  
"syncFlag": true, requester is sync flag on  
"isSyncFlag": true, requester is sync flag on  
"createdDate": "0000-00-00T00:00:00.000",



```

"modifiedDate": "0000-00-00T00:00:00.000",
"requesterType": "Default", requester type
"stateName": "Default State", requester state name
"countryName": "Default Country", requester country name
"prefixName": null, requester prefix name
"sufixName": null, requester suffix name
"localeId": 0, requester language
"updateRequestAddresses": 0, requester update address
"updateAddresses": 0, requester update address
"otherState": null, requester has another state
"customFields": [], requester has custom fields
"id": 0 requester id
}

```

## 4.2 Requesters/create

Create a requester based on the information provided by the user.

### 4.2.1 Parameters

```

{
  "id": 0,
  "requesterTypeId": 0,
  "prefix": 0,
  "suffix": 0,
  "lastName": "string",
  "firstName": "string",
  "middleName": "string",
  "jobTitle": "string",

```



```
"workPhone1": "string",
"workPhone2": "string",
"homePhone": "string",
"mobile": "string",
"fax": "string",
"email": "string",
"company": "string",
"login": "string",
"notes": "string",
"address1": "string",
"address2": "string",
"city": "string",
"stateId": 0,
"countryId": 0,
"zipCode": "string",
"localeId": 0,
"createdDate": "2024-04-09T16:05:22.627Z",
"modifiedDate": "2024-04-09T16:05:22.627Z",
"createdById": 0,
"modifiedById": 0,
"stateName": "string",
"otherState": "string",
"customFields": [
  {
    "customFieldId": "string",
    "customFieldValue": "string",
    "customFieldType": 1
```



```

    }
  ]
}

```

## 4.2.2 Output

After executing the method the requester is created, and an output appears as shown below:

POST /requesters/create

Parameters

No parameters

Request body

Example Value | Schema

```

{
  "id": 0,
  "requesterTypeId": 0,
  "prefix": 0,
  "suffix": 0,
  "lastName": "string",
  "firstName": "string",
  "middleName": "string",
  "jobTitle": "string",
  "workPhone1": "string",
  "workPhone2": "string",
  "homePhone": "string",
  "mobile": "string",
  "fax": "string",
  "email": "string",
  "company": "string",
  "login": "string",
  "notes": "string",
  "address1": "string",
  "address2": "string",
  "city": "string",
  "stateId": 0,
  "countryId": 0,
  "zipCode": "string",
  "localeId": 0,
  "createdAt": "2024-04-08T19:25:36.046Z",
  "modifiedAt": "2024-04-08T19:25:36.046Z",
  "createdById": 0,
}

```

## 4.3 Requesters/Update

Update a requester based on the information provided by the user

### 4.3.1 Parameters

```

{
  "id": 0,
  "requesterTypeId": 0,
  "prefix": 0,

```



```
"suffix": 0,  
"lastName": "string",  
"firstName": "string",  
"middleName": "string",  
"jobTitle": "string",  
"workPhone1": "string",  
"workPhone2": "string",  
"homePhone": "string",  
"mobile": "string",  
"fax": "string",  
"email": "string",  
"company": "string",  
"login": "string",  
"notes": "string",  
"address1": "string",  
"address2": "string",  
"city": "string",  
"stateId": 0,  
"countryId": 0,  
"zipCode": "string",  
"localeId": 0,  
"createdDate": "2024-04-09T16:05:55.310Z",  
"modifiedDate": "2024-04-09T16:05:55.310Z",  
"createdById": 0,  
"modifiedById": 0,  
"stateName": "string",  
"otherState": "string",
```





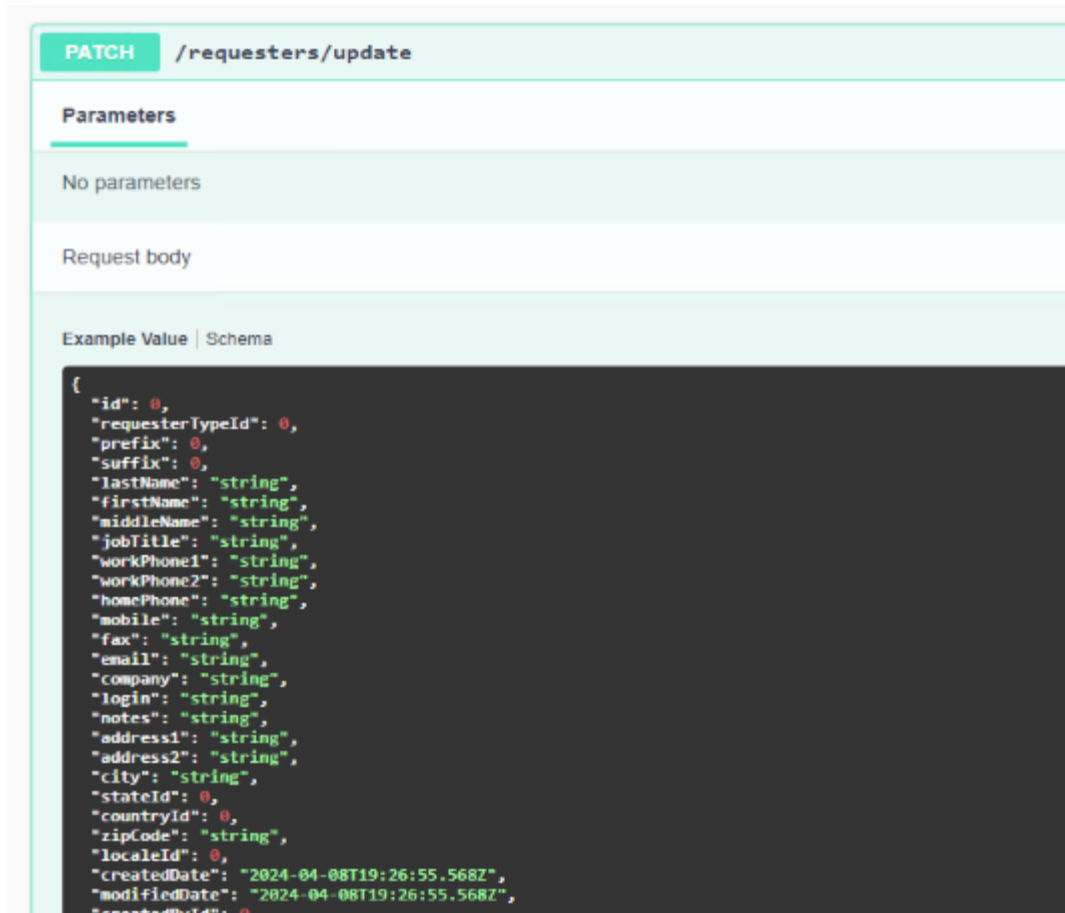
```

"customFields": [
  {
    "customFieldId": "string",
    "customFieldValue": "string",
    "customFieldType": 1
  }
]
}

```

### 4.3.2 Output

After executing the requesters/update method, an output appears as shown in the following example:



**PATCH** `/requesters/update`

**Parameters**

No parameters

**Request body**

Example Value | Schema

```

{
  "id": 0,
  "requesterTypeId": 0,
  "prefix": 0,
  "suffix": 0,
  "lastName": "string",
  "firstName": "string",
  "middleName": "string",
  "jobTitle": "string",
  "workPhone1": "string",
  "workPhone2": "string",
  "homePhone": "string",
  "mobile": "string",
  "fax": "string",
  "email": "string",
  "company": "string",
  "login": "string",
  "notes": "string",
  "address1": "string",
  "address2": "string",
  "city": "string",
  "stateId": 0,
  "countryId": 0,
  "zipCode": "string",
  "localeId": 0,
  "createdDate": "2024-04-08T19:26:55.568Z",
  "modifiedDate": "2024-04-08T19:26:55.568Z",
  "createdBy": 0
}

```

## 5 Audit Methods

You can use the Audit Methods to query for User Actions and User Logins in the application.

### 5.1 User Actions

The UserActions method gets all user actions between two dates.

**Note: You must have Audit permissions within the application to access this method.**

To audit user actions using the API:

1. Locate the UserActions section of the API, as shown below:

### UserActions

GET

/useractions

Parameters

| Name                                       | Description                             |
|--------------------------------------------|-----------------------------------------|
| FromDate<br>string(\$date-time)<br>(query) | <input type="text" value="2023-01-01"/> |
| ToDate<br>string(\$date-time)<br>(query)   | <input type="text" value="2024-01-01"/> |
| PageSize<br>integer(\$int32)<br>(query)    | <input type="text" value="10"/>         |
| PageNumber<br>integer(\$int32)<br>(query)  | <input type="text" value="1"/>          |

2. Enter the dates you'd like to audit User Actions between using the *FromDate* and *ToDate* fields.

**Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following**



**date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.**

3. The `pageSize` and `pageNumber` fields are required and configure the formatting for the returned results. *pageSize* indicates the number of items returned per page, and *pageNumber* dictates the current page number.
4. Execute the method to view matching results, which display in the *Response body* field. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body

The screenshot shows an API response interface. At the top, there's a 'Code' section with '200'. Below it, the 'Response body' is displayed in a dark box with a red border, containing a JSON object. To the right of the response body are two buttons: 'A' (Copy) and 'B' (Download). Below the response body, the 'Response headers' are listed. At the bottom, there's a 'Responses' table with columns 'Code', 'Description', and 'Links'.

```

{
  "userActionId": 1,
  "actionTime": "2023-10-16T13:31:57.437",
  "actionPerformed": "'Fiscal year' updated from 2023 to 2024, with Start Date as '10/1/2023' and End Date as '9/30/2024'",
  "actionBy": "admin",
  "actionOffice": "HQ"
}

```

```

api-supported-versions: 1.0
content-type: application/json; charset=utf-8
date: Thu, 19 Oct 2023 17:15:38 GMT
server: Kestrel

```

| Code | Description | Links    |
|------|-------------|----------|
| 200  | Success     | No links |

## 5.2 User Logins

You can use the `UserLogins` method to get all user logins between two dates.

**Note: Users utilizing Audit methods must have Audit permissions within the application.**

To audit user logins using the API:

1. Locate the `UserActions` section of the API, as shown below:



## UserLogins

GET /userlogins

### Parameters

| Name | Description |
|------|-------------|
|------|-------------|

|                                            |            |
|--------------------------------------------|------------|
| FromDate<br>string(\$date-time)<br>(query) | 2023-01-01 |
|--------------------------------------------|------------|

|                                          |            |
|------------------------------------------|------------|
| ToDate<br>string(\$date-time)<br>(query) | 2024-01-01 |
|------------------------------------------|------------|

|                                         |    |
|-----------------------------------------|----|
| PageSize<br>integer(\$int32)<br>(query) | 10 |
|-----------------------------------------|----|

|                                           |   |
|-------------------------------------------|---|
| PageNumber<br>integer(\$int32)<br>(query) | 1 |
|-------------------------------------------|---|

2. Enter the dates you'd like to audit User Logins between using the *FromDate* and *ToDate* fields.

**Note: Dates follow UTC format and will take the hours and minutes as 00:00 unless specified. To fetch values for a specific date, either specify the time or use the following date. For example, to fetch values for 11/1/2023, use either 2023-11-01T23:59Z or 2023-11-02.**

3. The *pageSize* and *pageNumber* fields are required and configure the formatting for the returned results. *pageSize* indicates the number of items returned per page, and *pageNumber* dictates the current page number.
4. Execute the method to view matching results, which display in the *Response body* field. There are options to **(A) Copy** this to your clipboard, or **(B) Download** the response body.



## Audit Methods

Server response

Code Details

200

Response body

```
{
  "sessionId": 1,
  "userName": "Admin, Admin",
  "actionOffice": "HQ",
  "workStation": "::1",
  "loginTime": "2023-10-16T13:31:54.64",
  "logoutTime": "2023-10-16T14:16:22.55",
  "duration": 2668,
  "groupID": 1,
  "groupName": "Admin",
  "loginSuccess": "Success"
},
{
  "sessionId": 2,
  "userName": "Admin, Admin",
  "actionOffice": "HQ",
  "workStation": "::1",
  "loginTime": "2023-10-17T11:51:28.41",
  "logoutTime": "2023-10-17T11:56:40.91",
  "duration": 312,
  "groupID": 1,
  "groupName": "Admin",
  "loginSuccess": "Success"
}
}
```

A B

Download



## 6 EmailService API

### 6.1 Mail/api/getmail

Search the email inbox and return the information requested in the query.

#### 6.1.1 Parameters

```
{
  "hasWords": [
    "string"
  ],
  "from": [
    "string"
  ],
  "to": [
    "string"
  ],
  "dateFrom": "2024-04-09T16:07:35.291Z",
  "dateTo": "2024-04-09T16:07:35.291Z",
  "hasAttachment": true
}
```

#### 6.1.2 Output

```
[
  {
    "id": "Random Text",
    "from": "email",
    "to": "email",
    "cc": null,
```



```

"bcc": null,
"subject": "test",
"shortBody": "html code",
"body": "html code",
"receivedDateTime": "2023-12-04T22:27:28",
"hasAttachments": false,
"attachments": "plus.png"
},

```

**Mail**

**POST** /mail/api/getmail

**Parameters**

No parameters

**Request body**

Example Value | Schema

```

{
  "hashWords": [
    "string"
  ],
  "from": [
    "string"
  ],
  "to": [
    "string"
  ],
  "dateFrom": "2024-04-08T20:16:06.683Z",
  "dateTo": "2024-04-08T20:16:06.683Z",
  "hasAttachment": true
}

```

## 6.2 Mail/api/getmailcontent

Search the email inbox and return the requested information in the query in MIME format.

### 6.2.1 Parameters

```

{
  "mailId": [

```

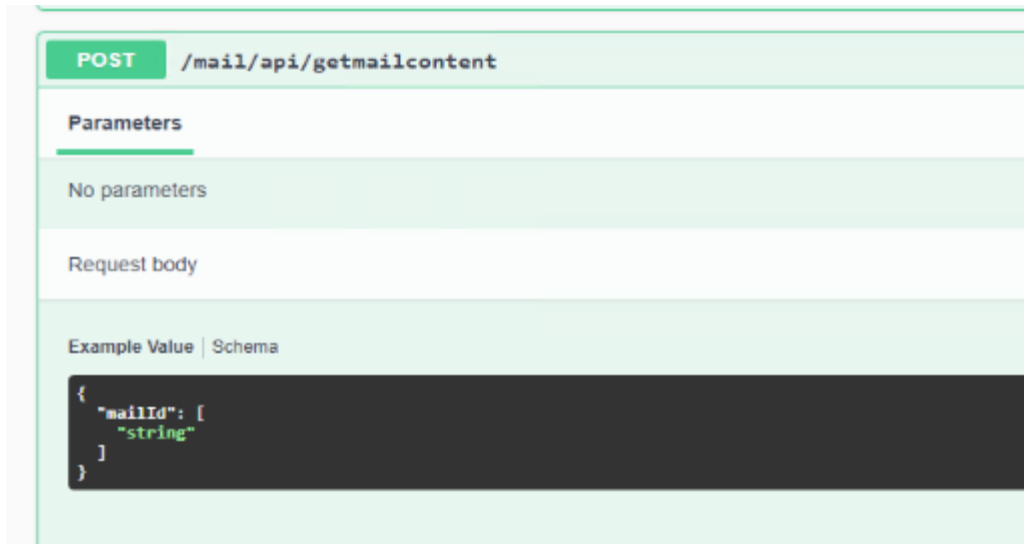


"string"

]

## 6.2.2 Output

["HTML MIME format"]



The image shows a snippet of API documentation for a POST endpoint. The endpoint is `POST /mail/api/getmailcontent`. Below the endpoint, there are sections for "Parameters", "Request body", and "Example Value | Schema". The "Parameters" section indicates "No parameters". The "Request body" section is empty. The "Example Value | Schema" section shows a JSON object with a key `"mailId"` and a value `["string"]`.

```
POST /mail/api/getmailcontent
```

Parameters

No parameters

Request body

Example Value | Schema

```
{
  "mailId": [
    "string"
  ]
}
```



# 7 EntityRecognition API

## 7.1 Comprehend/api/detectentities

Perform a Named Entity Recognition (NER) process using AWS Comprehend according to the specified language and entity type for the requested text

### 7.1.1 Parameters

lang: Text language

EntityType: Name entities, PII or both

### 7.1.2 Output

```
{"Entities":[
  {"BeginOffset":0, initial position
  "EndOffset":13 final position
  ,"Score":0.9925614 score from aws
  ,"Text":"test", text reviewed
  "Type":{"Value":"OTHER"}}] name entity
}
```

### Comprehend

POST

/comprehend/api/detectentities

Parameters

| Name                            | Description                             |
|---------------------------------|-----------------------------------------|
| lang<br>string<br>(query)       | <input type="text" value="lang"/>       |
| entityType<br>string<br>(query) | <input type="text" value="entityType"/> |



## 7.2 Comprehend/API/detectentities

Perform a Named Entity Recognition (NER) process using Core NLP according to the specified language and entity type for the requested text.

### 7.2.1 Parameters

lang: Text language

### 7.2.2 Output

```
{
  "sentences": [
    {
      "index": 0,
      "entitymentions": [
        {
          "docTokenBegin": 0, token start position
          "docTokenEnd": 1, token end position
          "tokenBegin": 0, token start position
          "tokenEnd": 1, token end position
          "text": "Yesterday", text reviewed
          "characterOffsetBegin": 0, token start position
          "characterOffsetEnd": 9, token end position
          "ner": "DATE", name entity

          "normalizedNER": "OFFSET P-1D",
          "nerConfidences": {
            "DATE": -1 name entity confidence
          }
        }
      ]
    }
  ]
}
```



## CoreNLP

**POST** /corenlp/api/getentities

### Parameters

| Name | Description |
|------|-------------|
|------|-------------|

|                           |  |
|---------------------------|--|
| lang<br>string<br>(query) |  |
|---------------------------|--|



## 8 MediaRedactions API

### 8.1 Mediaredactions/apiex/mediafile/getsignedurl

Obtains a signed URL which will be used to download a media file.

#### 8.1.1 Parameters

```
{  
  "uniqueId": "string",  
  "folderId": 0,  
  "reviewLayerId": 0,  
  "pageId": 0,  
  "createdBy": 0,  
  "createdDate": "2024-04-09T16:11:28.623Z",  
  "declassInfo": "string",  
  "reviewLayerName": "string",  
  "tokenHash": "string",  
  "downloadedFilePath": "string"  
}
```

Output:

string text



The image shows a snippet of API documentation for a POST endpoint. The endpoint is `/mediaredactions/apix/mediafile/getsignedurl`. It indicates that there are no parameters and no request body. An example JSON value is provided, showing fields like `uniqueId`, `folderId`, `reviewLayerId`, `pageId`, `createdBy`, `createdDate`, `declassInfo`, `reviewLayerName`, `tokenHash`, and `downloadedFilePath`.

```
POST /mediaredactions/apix/mediafile/getsignedurl
```

**Parameters**

No parameters

**Request body**

Example Value | Schema

```
{
  "uniqueId": "string",
  "folderId": 0,
  "reviewLayerId": 0,
  "pageId": 0,
  "createdBy": 0,
  "createdDate": "2024-04-08T21:02:59.626Z",
  "declassInfo": "string",
  "reviewLayerName": "string",
  "tokenHash": "string",
  "downloadedFilePath": "string"
}
```

## 8.2 Mediaredactions/apix/mediafile/getsignedurl/sig nedToken

Retrieves the file with the signed URL obtained from the `getSignedUrl` method

### 8.2.1 Parameters

`signedToken` :URL obtained from the `getSignedUrl` method to get a file



## 8.2.2 Output

|                                                                                                  |                                                                |
|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>POST</b>                                                                                      | <b>/mediaredactions/apix/mediafile/submitredactedmediainfo</b> |
| <b>Parameters</b>                                                                                |                                                                |
| No parameters                                                                                    |                                                                |
| Request body                                                                                     |                                                                |
| Example Value   Schema                                                                           |                                                                |
| <pre>{   "requestID": "string",   "redactedFileURL": "string",   "auditLogURL": "string" }</pre> |                                                                |
| <b>Responses</b>                                                                                 |                                                                |



## 9 Reports API

### 9.1 Report/api/isaccuratexml

Checks if the user's XML has a valid format for the annual report submission.

#### 9.1.1 Parameters

XML file

#### 9.1.2 Output

```
{  
  "accurate": true, if is accurate  
  "differencesCount": 0, differences count  
  "differences": null list of differences  
}
```

| Report                      |                           |
|-----------------------------|---------------------------|
| POST                        | /report/api/isaccuratexml |
| Parameters                  |                           |
| No parameters               |                           |
| Request body                |                           |
| XmlFile<br>string(\$binary) |                           |

