|  |  |
| --- | --- |
| Release Date: | 25 August 2021 |
| Revision: | External API v3 |

# **Iridium GMDSS**

# External API Documentation

# **Table of Contents**

Contents

[**Iridium GMDSS** 0](file:///C:\Users\Lawrence_M\Desktop\GMDSSExternalAPIDocumentationV4.docx#_Toc81490746)

[External API Documentation 0](file:///C:\Users\Lawrence_M\Desktop\GMDSSExternalAPIDocumentationV4.docx#_Toc81490747)

[**Table of Contents** 1](#_Toc81490748)

[**API Overview** 2](#_Toc81490749)

[**What is an API** 2](#_Toc81490750)

[**Authentication** 2](#_Toc81490751)

[**API Details** 3](#_Toc81490752)

[**Errors and Error Handling** 3](#_Toc81490753)

[**Authentication** 3](#_Toc81490754)

[**Pagination** 4](#_Toc81490755)

[**API Resources** 4](#_Toc81490756)

[**Authenticate** 4](#_Toc81490757)

[**Re-Authenticate** 6](#_Toc81490758)

[**MSIs** 7](#_Toc81490759)

[**List MSIs** 7](#_Toc81490760)

[**Fetch Referenced MSI** 9](#_Toc81490761)

[**Create New MSI** 10](#_Toc81490762)

[**Cancel Referenced MSI** 12](#_Toc81490763)

[**Acknowledgements** 13](#_Toc81490764)

[**Version Number** 15](#_Toc81490765)

[**External Documents** 15](#_Toc81490766)

# **API Overview**

Iridium’s API provide an alternative method of communicating with our web server. The API allows RCCs (Rescue Coordination Centers) and MSIPs (Maritime Safety Information Providers) to send data to our infrastructure without utilizing our web portal GUI (gmdss.iridium.com), SFTP, or Email.

To access this API, you will need to be pre-registered with the GMDSS Web Portal, as we use the same credentials. To reset your password, please follow the steps in the Iridium GMDSS Web Portal User Guide. Users will have the same access and roles as they do on the Web Portal.

## **What is an API**

An API is an Application Programming Interface, meaning that it is a set of standards that let programs communicate with each other. This allows software to expose data for each other and increase their functionality around the internet in a consistent format.

REST, which stands for Representation State Transfer, is the architectural standard that Iridium’s GMDSS API utilizes to provide our services and expose the data requested by the end user. This means that our API is accessed by utilizing the HTTP protocol at a set of URLs we have defined for you. These URLs provide several different resources to the end user, and the data we provide is accessed through these URLs, given to the requester as JSON data.

Our calls utilize 3 different HTTP methods, GET, POST, and PUT. These methods categorize the types of requests APIs handle and determine how the API will manage the request. A GET request works as you would expect, by grabbing the data stored at the URL requested and providing it to the end user. POST and PUT requests have specific, not as obvious meanings. The common rule of thumb is to use POST to create resources, and use PUT to update resources.

Iridium exposes these RESTFUL API endpoints for users to be able to send and receive data from our GMDSS System. Iridium does not want just anyone to be able to access these endpoints and read through our data, so we require an authentication token to be passed along with the API URL that has the data the end user requests. To receive an authentication token, you need to send a HTTP GET Request to our first RESTFUL API Endpoint, <https://gmdss.iridium.com/api/authenticate>.

## **Authentication**

In order to access the data at the API endpoints Iridium exposes, we need to be able to verify your identity. This ensures that our data remains secure and private, accessible only if you have been granted access. To verify your identity, we require you to send an authentication token with your requests to receive and create data in our system.

By using our authentication GET request, you can receive your authentication token. All clients with username and password credentials for the gmdss.iridium.com web portal will utilize the same credentials to receive the authentication token for our API. By default, the token received will expire after 10 minutes, although this can be changed with an optional parameter. The token will also expire if the user logs in the Web Portal or if the user calls for another token from the API. Only one log-in session per-user at a time is allowed. Any attempt to log in a second time will cause the older session to return a **401 Unauthorized** error.

API calls generally are more complex than a standard web query, and a tool such as cURL can be used to make these calls. All API calls listed will include an example shown in cURL.

# **API Details**

All the exposed paths in the Iridium GMDSS External API are case-sensitive, and an all-lowercase resource name convention is used. Any API requests should be appended to the base URL:

**Base URL:** <https://gmdss.iridium.com/api>

For example, the path /api/msi refers to the msi resource and should be used to perform operations on that resource. For MSIs, you can reference an individual MSI by adding the MSI id to the resource path. For instance, the path /api/msi/60ed9\_302 refers to the MSI with the reference number 60ed9\_302 and will provide the MSI and its data. Some of the API calls allow for the use of Query Parameters to help filter results. Please note that our query parameters use a camel case convention.

An example of our query parameters to only show Meteorological MSIs, would be <https://gmdss.iridium.com/api/msi?msiType=met>.

## **Errors and Error Handling**

The API utilizes conventional HTTPs response codes to indicate pass or fail of an API request.

|  |  |  |
| --- | --- | --- |
| HTTP Response Code | Description | In More Detail |
| 200-299 | Success | Anything in this range means that your operation was received and completed by the server |
| 400-499 | Client Error | This error is likely due to a missing request parameter, or perhaps an invalid request |
| 500-599 | Server Error | There was an error on the server side |

Iridium uses a few specific codes for these common cases:

* **400 Bad Request** – occurs when the client provides a request with invalid syntax
* **401 Unauthorized** – occurs when the server receives requests without correct authentication
* **403 Forbidden** – for requests that the authentication provided does not have grant access
* **404 Not Found** – for requests where associated referenced entities are not found in the system

In addition to the HTTP Response Codes, this API will return a response payload with an ‘errorMessage’ and an ‘errorCode’ that may provide clarity on what went wrong. For example,

{

“errorMessage”: “Invalid Repetition Code”,

“errorCode”: 400

}

## **Authentication**

Iridium uses a security token to authenticate requests. Without the correct token, you will be unable to access any of the data at the API endpoints. To get a security token, you can use the authenticate resource to supply the server with your username and password credentials. All client access to the GMDSS API resources requires username and password GMDSS Web Portal credentials. Using the same credentials as you would on the GMDSS Web Portal, you can request a security token that will be required to request data from the API.

## **Pagination**

Two of our API calls support Pagination. This allows the client to flip through the list of responses by the server as though they were in a book. We supply two query parameters to implement this capability, limit and offset. Think of limit as how many records you would like to see. If you chose a limit of 5, the server would only send the first 5 responses in the system. Offset represents how many records the server should ignore before sending you the list of responses. If you set an offset of 5, the server would only show you the 6th response and onward. Combining these two query parameters will allow the client to fetch records at any interval. If you wanted to receive the records in intervals of 5, you would first make a call with a limit of 5 and an offset of 0, then you would repeat the call but with an offset of 5, so that it returns the next 5 records, and an offset of 10 to show the third set of 5 and so on.

# **API Resources**

To access any of the following API resources, you need to send your request to the proper path. The proper path for each API is the value in the Path column of the below table appended to the Base URL listed below.

For example, the proper path for the Authenticate resource is <https://gmdss.iridium.com/api/authenticate>.

**Base URL**: <https://gmdss.iridium.com/api/>

|  |  |  |  |
| --- | --- | --- | --- |
| API Resource | HTTP Method | Path | Description |
| [Authenticate](#_Authenticate) | POST | /authenticate | Provides security token required to access the other API Resources |
| [Re-Authenticate](#_Re-Authenticate) | GET | /authenticate | Refreshes security token if called with an active token |
| [List MSIs](#_List_MSIs) | GET | /msi | Returns a list of all MSIs created by your provider, can be filtered |
| [Fetch Referenced MSI](#_Fetch_Referenced_MSI) | GET | /msi/{reference} | Returns the MSI that has the id provided |
| [Create New MSI](#_Create_New_MSI) | POST | /msi | Allows the creation of a new MSI to be sent to our server |
| [Cancel Referenced MSI](#_Cancel_Referenced_MSI) | PUT | /msi/{reference} | Cancels the MSI that has the id provided |
| [List Acknowledgements](#_Acknowledgements) | GET | /msi/{reference}/ack | Returns list of all ships that have been sent the MSI, and the time the ship received the acknowledgement |
| [Version Number](#_Version_Number) | GET | /version/external | Returns the current version number of the External API |

## **Authenticate**

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| POST | /authenticate | Provides Security Token needed for full API access |

The authenticate request is a POST request that requires data to be sent in JSON format. Username and password credentials are required. There is a third, optional, parameter which allows you to set how long the security token will be valid. If you do not include this parameter, the token will be valid for the default time of 10 minutes. If you include the parameter, the token will be valid for the length of time provided (in minutes). If you set “time” to be “0”, the token will be valid for 1 year, or until that user account is logged in somewhere else (as discussed in [Authentication](#_Authentication)). To use this call, provide, at minimum, your username and password credentials as payload:

{

“username”: “yourusername”,

“password”: “yourpassword”

}

The call will also accept a payload in this format:

{

“username”: “yourusername”,

“password”: “yourpassword”,

“time”: “10”

}

If used, time needs to be an integer passed as a string like shown above. The units of time are minutes, so if “time” is set to “100”, the token will remain valid for 100 minutes. If “time” is set to “0”, the security token will be set to be valid for 1 year. If “time” is not included in the payload, the token will default to be valid for 10 minutes.

**cURL Example**:

curl -X POST "https://gmdss.iridium.com/api/authenticate" -H "accept: application/json" -H "Content-Type: application/json" -d "{ "username": "yourusername", "password": "yourpassword"}"

**Example Response:**

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0NzQ3NzExNjAiLCJpYXQiOjE2Mjk0NzQ3NzEsImV4cCI6MTYyOTQ3NTM3MX0.bX3Z431qdF9DSkcTfD4P6gmXi84nPc0dB5CTdU3OiQ"

}

**cURL Example with Time:**

curl -X POST "https://gmdss.iridium.com/api/authenticate" -H "accept: application/json" -H "Content-Type: application/json" -d "{ "username": "yourusername", "yourpassword": "123456", "time": "100"}"

**Example Response:**

{

"token": "eyJhbGciOiJIUzI1NiIsInD4eWI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0NzYxMDMwOTIiLCJpYXQiOjE2Mjk0NzYxMDMsImV4cCI6MTYyOTQ4MjEwM30.Lv1vdG3LMQPGD8rmiabi4yPRBE4S-rV8OSpBtHnnZ\_E"

}

As shown, the server will respond with a JSON response that includes the token in the payload:

{

"token": "eyJhbGciOiJIUzI1NiJ9.cmNiJFEtMTYyOTIxNzc2SFU3Mw.FdRZk\_Jj3FC2wGhWzHmABjlstizRfZ6vjxgxGqI6vR4"

}

If the authentication fails, the server will respond with a **401 Unauthorized** error with an error message indicating an invalid username / password combination:

{

"errorMessage": "Invalid User Credentials",

"errorCode": 401

}

After receiving the token, the client will need to use it by including it as an Authorization header of any other HTTP request made to the server. This Authorization header will be required for every other API call, and the server will return a **401 Unauthorized** error if it is not included.

{

“error”: “Access Required”

“status”: 401

}

As mentioned, the token will expire, and the server will return a **401 Unauthorized** error when you try to use the same token after its expiration. To renew your token before it expires, call the GET /authenticate method discussed below. If your token has already expired, you will need to redo the ‘Post’ authenticate call.

## **Re-Authenticate**

|  |  |  |
| --- | --- | --- |
| Method | Path | Description |
| GET | /authenticate | Provides a new security token, if your current security token is valid |

The GET authenticate request allows users to renew their security token. When called, this call will check your current security token, and if it is still valid, it will cancel it and issue you a new token. This allows you to avoid re-entering your credentials. It also allows the use of a single query parameter, time.

|  |  |
| --- | --- |
| Query Parameter | Description |
| time | How long you want the new security token to remain valid, in minutes. (Default is 10 minutes) |

If the time query parameter is not included, the security token generated by this call will be valid for 10 minutes. If the time query parameter is set to equal 0, the newly provided security token will expire in 1 year.

**cURL Example:**

curl -X GET "https://gmdss.iridium.com/api/authenticate?time=100" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0NzQ3NzExNjAiLCJoDXSiOjE2Mjk0NzQ3NzEsImV4cCI6MTYyOTQ3NTM3MX0.bX3Z431qiD2DQfkcTfD4P6gmXi84nPc0dB5CTdU3OiQ"

**Example Response:**

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0NzUyNTIFSESiLCJpYXQiOjE2Mjk0NzUyNTIsImV4cCI6MTYyOTQ3NTg1Mn0.Ze-iLNJGzSe-BQJe6r32QLZOYa9moOCwDejfTITfe2k"

}

If your current token has already expired, or if you do not pass an authenticated security token as an Authorization header, the server will respond with a **401 Unauthorized** error.

## **MSIs**

This API allows RCCs and MSIPs to navigate through their created MSIs, create new MSIs and cancel current MSIs. These calls vary in HTTP Method, but they all include /msi as the starting path. Be sure to confirm that your HTTP requests are using the correct combination of the path and the HTTP method, shown in the table below.

|  |  |  |
| --- | --- | --- |
| HTTP Method | Path | Operation |
| GET | /msi | [List MSIs](#_List_MSIs) |
| GET | /msi/{reference} | [Fetch referenced MSI](#_Fetch_Referenced_MSI) |
| POST | /msi | [Create New MSI](#_Create_New_MSI) |
| PUT | /msi/{reference} | [Cancel referenced MSI](#_Cancel_Referenced_MSI) |

## **List MSIs**

This API endpoint is used to request a list of MSIs from the server. This call will return all MSIs created by anyone with the same ProviderId as the client, including the MSIs they have created. This call has a list of Query Parameters that can be used to filter the response from the server.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| GET | /msi | List MSIs |

**cURL Example:**

curl -X GET "https://gmdss.iridium.com/api/msi" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0Nzg2MjYwMDAiLCJpYXQiOjE2Mjk0Nzg2MjYsImV4cCI6MTYyOTQ4NDYyNn0.ZSacbGjEUds\_maxjgVrZC4umABf60SDxJWH0L7rM7eY"

**Example Response:**

[

{

"referenceNumber": "6112ed4c\_c5d3",

"msiType": "met",

"areaType": "NAVMET",

"address": "04",

"activeStartDate": "2021-08-10T21:19:08Z",

"activeEndDate": "2021-08-14T08:00:00Z",

"repetitionCode": "01",

"priority": "safety",

"stationId": "81",

"payload": "test",

"cancelled": "true",

"status": "cancelled",

"serviceCode": "31",

"egcId": "N/A",

"readAck": "false",

"receiveAck": "true",

"presentation": "ia5"

},

{

"referenceNumber": "6112ed61\_e8b4",

"msiType": "met",

"areaType": "NAVMET",

"address": "04",

"activeStartDate": "2021-08-10T21:19:29Z",

"activeEndDate": "2021-08-15T08:00:00Z",

"repetitionCode": "01",

"priority": "safety",

"stationId": "81",

"payload": "test again",

"cancelled": "false",

"status": "finished",

"serviceCode": "31",

"egcId": "N/A",

"readAck": "false",

"receiveAck": "true",

"presentation": "ia5"

}

]

To filter the server response, the following query parameters can be appended to the URL.

|  |  |
| --- | --- |
| Query Parameter | Description |
| limit | Maximum number of MSIs to return on each request |
| offset | How many records to skip, from the beginning of the list. See [Pagination](#_Pagination) section for clarification |
| startDate | The date and time the MSI(s) were transmitted. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| startDateBefore | List MSIs that were transmitted before the provided date and time. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| startDateAfter | List MSIs that were transmitted after the provided date and time. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| endDate | The date and time the MSI(s) are set to expire. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| endDateBefore | List MSIs that expired before the provided date and time. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| endDateAfter | List MSIs that expired after the provided date and time. (UTC date format: YYYY-MM- DDThh:mm:ssZ) |
| msiType | List certain types of MSIs (sar, nav, met) - supports multiple values, separated by commas |
| priority | List only MSIs with the provided priority (urgency, safety) |
| address | List only the MSI(s) that have this address code |
| repetitionCode | List only the MSI(s) that have the provided repetition code |
| status | List only the MSI(s) that have the provided status (scheduled, active, finished, cancelled) - supports multiple values, separated by commas |
| serviceCode | List only the MSI(s) that have the provided service code |

For example, if you wanted to search for MSIs that have the priority “urgency” and the msiType “nav”, you would send a GET request, with an Authorization header that includes your authentication token, to the following URL:

<https://gmdss.iridium.com/api/msi?msiType=nav&priority=urgency>

This would return a list of every MSI that has been created by any user who belongs to the same organization, that is the MSI type of ‘nav’ and has a priority of ‘urgency’.

An additional note: this call will provide up to 25 MSIs with default parameters. Using the limit parameter, you can extend this up to 100 MSIs per API call. If there are not 100 MSIs the system will return as many as possible.

## **Fetch Referenced MSI**

If you know the referenceNumber/ID of a specific MSI, you can use the Fetch MSI endpoint to access the data for that unique MSI, provided you have the proper authorization and permissions.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| GET | /msi/{reference} | Fetch Referenced MSI |

The {reference} part of the path is a placeholder for the reference number of the MSI you are requesting. For example, if you wanted to view the MSI with a reference number of “6112ed61\_e8b4”, you would send:

**cURL Example:**

curl -X GET "https://gmdss-test.iridium.com/api/msi/6112ed61\_e8b4" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0Nzg2MjYwMDAiLCJpYXQiOjE2Mjk0Nzg2MjYsImV4cCI6MTYyOTQ4NDYyNn0.ZSacbGjEUds\_maxjgVrZC4umABf60WEzJWG8k4M7eY"

**Example Response:**

[

{

"referenceNumber": "6112ed61\_e8b4",

"msiType": "met",

"areaType": "NAVMET",

"address": "04",

"activeStartDate": "2021-08-10T21:19:29Z",

"activeEndDate": "2021-08-15T08:00:00Z",

"repetitionCode": "01",

"priority": "safety",

"stationId": "81",

"payload": "test",

"cancelled": "false",

"status": "finished",

"serviceCode": "31",

"egcId": "N/A",

"readAck": "false",

"receiveAck": "true",

"presentation": "ia5"

}

]

If the user does not have the proper privileges or access to the requested MSI, the server will respond with **a 403 Forbidden** error.

If the MSI is not found, the server will respond with a **404 Not Found** error.

This API endpoint does not support Query Parameters. Any query parameters provided will be ignored.

## **Create New MSI**

This API endpoint sends our system a new MSI to be broadcasted by providing valid MSI message parameters in the payload of the request.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| POST | /msi | Create New MSI |

There are 13 parameters that can be passed in the payload of your HTTP request. Of these 13, only 7 of them are required, and the other 6 may or may not be included at your discretion. Also note that these are sent in the JSON payload, and NOT as query parameters. Any query parameters sent to this call will be ignored.

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Required |
| msiType | The type of MSI (sar, nav, met) | Yes |
| areaType | The MSI area type (NAVMET, COASTAL, CIRCULAR, RECTANGULAR) | Yes |
| address | The message destination per Annex 4 (C3) of the [Iridium EGC Manual](#_External_Documents) | Yes |
| startDate | The date to start broadcasting the MSI. (UTC date format: YYYY-MM-DDThh:mm:ssZ) | No, defaults to start now |
| endDate | The date to stop broadcasting the MSI. (UTC date format: YYYY-MM-DDThh:mm:ssZ) | No, defaults to expire 7 days after it starts |
| repetitionCode | The repetition code per Part E of the [Iridium EGC Manual](#_External_Documents) | Yes |
| priority | The MSI priority (urgency, safety) | Yes |
| serviceCode | The MSI service code per Annex 4 of the [Iridium EGC Manual](#_External_Documents) | Yes |
| receiveAck | Receive Acknowledgements (true or false) | No, defaults to true |
| readAck | Read Acknowledgements (true or false) | No, defaults to false |
| presentation | The MSI payload encoding (ia5 is only acceptable value) | No, defaults to ia5 |
| nationalSasId | The national SAS ID | No, defaults to providerID |
| payload | The message payload. (Only valid IA5 Characters are allowed) | Yes |

In the payload of the HTTP request, create the MSI and send the parameters as properties of a JSON object. For example, if you wanted to create a Meteorological Warning for the Meteorological Area ‘04’, you could send:

{

"msiType": "nav",

"areaType": "NAVMET",

"address": "04",

"activeStartDate": "2021-08-20T10:15:00Z",

"activeEndDate": "2021-08-30T10:00:00Z",

"repetitionCode": "01",

"priority": "safety",

"serviceCode": "31",

"receiveAck": "true",

"readAck": "false",

"presentation": "ia5",

"nationalSasId": "101",

"payload": "test"

}

**cURL Example:**

curl -X POST "https://gmdss.iridium.com/api/msi" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Im1zaXBOQVY0X3N1bi0xNjI5NDgzMTQ0NTY2IiwiaWF0IjoxNjI5NDgzMTQ0LCJleHAiOjE2Mjk0ODM3NDR9.6FBqsdxu-a38oHx76pQI7-M2w40D9x4MloaPa3pAqDM" -H "Content-Type: application/json" -d "{ "msiType": "nav", "areaType": "NAVMET", "address": "04", "activeStartDate": "2021-08-20T10:15:00Z", "activeEndDate": "2021-08-30T10:00:00Z", "repetitionCode": "01", "priority": "safety", "serviceCode": "31", "receiveAck": "true", "readAck": "false", "presentation": "ia5", "nationalSasId": "101", "payload": "test"}"

If the MSI is created successfully, a **201 Created** response will be returned to the client, along with a JSON Object displaying the new MSI.

{

"referenceNumber": "611ff096\_bcc9",

"activeStartDate": "2021-08-20T10:15:00Z",

"activeEndDate": "2021-08-30T10:00:00Z",

"serviceCode": "31",

"repetitionCode": "01",

"msiType": "nav",

"areaType": "NAVMET",

"address": "04",

"payload": "test",

"stationId": "101",

"cancelled": false,

"egcId": "N/A",

"priority": "safety",

"readAck": false,

"receiveAck": true,

"presentation": "ia5"

}

If the MSI message parameters are invalid, the server will respond with a **400 Bad Request** alongside an appropriate error message:

{

“errorMessage”: “Address parameter is required”,

“errorCode”: 400

}

## **Cancel Referenced MSI**

This API endpoint is used to cancel an MSI. You can only cancel MSIs that are created by someone with the same provider ID. Once cancelled, the MSI cannot be restored.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| PUT | /msi/{reference} | Cancel MSI |

{reference} is a placeholder for the MSI ID that you would like to cancel. For example, if you needed to cancel an MSI with the id “6101\_1929”, then your path would become (Remember you need to append this to the base URL linked above):

/msi/6101\_1929

Also included with the request is a JSON object with the “cancelled” parameter. If cancelled is set to “true”, then the MSI will be cancelled, and the newly updated MSI will be returned, notably with “cancelled” set to “true”.

{

“cancelled”: “true”

}

This will change the status of the referenced MSI to cancelled. It is the only parameter that can be updated.

**cURL Example:**

curl -X PUT "https://gmdss.iridium.com/api/msi/6112ed4c\_c5d3" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0ODQwMTkxMjAiLCJpYXLjOjE2Mjk0ODQwMTksImV4cCI6MTYyOTQ4NDYxOX0.g-Qg-uVmnhclgooJvcFvS2vER-a3ltZmlJY26S0lvR4" -H "Content-Type: application/json" -d "{ "cancelled": "true"}"

**Example Response:**

[

{

"referenceNumber": "6112ed4c\_c5d3",

"msiType": "met",

"areaType": "NAVMET",

"address": "04",

"activeStartDate": "2021-08-10T21:19:08Z",

"activeEndDate": "2021-08-14T08:00:00Z",

"repetitionCode": "01",

"priority": "safety",

"stationId": "101",

"payload": "test",

"cancelled": "true",

"status": "cancelled",

"serviceCode": "31",

"egcId": "N/A",

"readAck": "false",

"receiveAck": "true",

"presentation": "ia5"

}

]

In the above example, notice that the “cancelled” parameter was passed as “true”, and the “cancelled” parameter in the returned value is now true. This indicates that the MSI is now cancelled. If the “cancelled” parameter you pass is set to “false”, the MSI will not be cancelled, but the MSI will still be returned. Please note that if the MSI was already cancelled, the MSI will still return “cancelled” as true.

If the user does not have proper privileges, the server will respond with a **403 Forbidden** error.

If the MSI cannot be found, the server will respond with a **404 Not found** error.

If the request cannot be authenticated, the server will respond with a **401 Unauthorized** error.

## **Acknowledgements**

This endpoint allows clients to view the recipients of an MSI. Using the MSI’s reference number, clients can request a list of ships that will be sent the MSI. This call will return both ships that have not yet received the MSI, and ships that have received it.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| GET | /msi/{reference}/ack | List Acknowledgements |

This call will return a list of JSON Objects representing each ship that will be sent the MSI. Ships that have received the MSI will have the time they received it stored in the “storeTime” property. In the below example, the first object represents a ship that has not yet received the MSI, while the second object represents a ship that has received the MSI. The second ship has recorded what time it has received the MSI, while the first one has not.

**cURL Example:**

curl -X GET "https://gmdss-test.iridium.com/api/msi/6112ed4c\_c5d3/ack" -H "accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InJjYzgxLTE2Mjk0ODQwMTkxDsAiLCJpYXQiOjE2Mjk0ODQwMTksImV4cCI6MTYyOTQ4NDYxOX0.g-Qg-uVmnhclgooJvcFvS2vER-a3ltZmlJY26S0lvR4"

**Example Response:**

[

{

"id": 1,

"acknowledgeTime": "",

"type": "receive",

"source": {

"id": 1,

"storeTime": "",

"type": "ut",

"parameters": [

{

"addressId": 0,

"storeTime": "",

"type": "callsign",

"value": "Thomas The Tank Engine"

},

{

"addressId": 0,

"storeTime": "",

"type": "imei",

"value": "300125343210230"

}

]

}

},

{

"id": 2,

"acknowledgeTime": "2021-08-11T16:12:11Z",

"type": "receive",

"source": {

"id": 1,

"storeTime": "2021-08-11T16:12:15Z",

"type": "ut",

"parameters": [

{

"addressId": 0,

"storeTime": "2021-08-11T16:12:15Z",

"type": "callsign",

"value": "Dora the Explorer"

},

{

"addressId": 0,

"storeTime": "2021-08-11T16:12:15Z",

"type": "imei",

"value": "300123261238470"

}

]

}

}

]

If the user does not have proper privileges to access the MSI, the server will respond with a **403 Forbidden** error.

If the server is unable to locate the MSI, the server will respond with a **404 Not found** error.

This API call does support several query parameters, listed below.

|  |  |
| --- | --- |
| Parameter | Description |
| limit | Maximum number of acknowledgements to return on each request |
| start or offset | Record number of the first record to be returned by the call out of the result set when multiple, identical are sent repeatedly. See [Pagination](#_Pagination) for clarification. |
| date | The time that the MSI was received by the vessel (UTC date format: YYYY-MM-DDThh:mm:ssZ). |
| dateBefore | Vessels that received the MSI before the time (UTC date format: YYYY-MM-DDThh:mm:ssZ) |
| dateAfter | Vessels that received the MSI after the time (UTC date format: YYYY-MM-DDThh:mm:ssZ) |
| type | The acknowledgement type (read, receive) |

Note: date requires an exact match and will most likely return no results. We recommend using dateBefore and dateAfter to establish a range of time that the MSI is between.

## **Version Number**

This API Call will return the Version Number of the API you are using.

|  |  |  |
| --- | --- | --- |
| Method | Path | Operation |
| GET | /external/version | Version Number |

## **External Documents**

Iridium EGC Manual: <https://iho.int/mtg_docs/com_wg/CPRNW/CPRNW_Doc-review/2017.06.02-Iridium_Global_Satellite_EGC_Manual(Without%20Video)-Draft.pdf>