

BIMCO Data Exchange Platform

Technical description

Version	Comments
1.1	Initial version by Kim K. Steensgaard

Contents

- Description
- Security
- JWT Token
- Common HTTP Headers
- Base API responses
- Structure
- Request to obtain the structure
- Request to fetch holiday information
- Example requests
- Understanding the data

Description

The BIMCO Data Exchange Platform is a REST API, that will expose otherwise protected information to BIMCO members. The API is currently found on the following URL: <https://membersapi.bimco.org>
The current version of the API only exposes information about local holidays, but intended to be extended in the future to also include other data found in the BIMCO databases.

Security

The REST API uses Azure Active Directory (AAD) as the identity-provider. All calls to the API must provide a JWT bearer token that identifies the caller. Each caller must be represented by a valid serviceprincipal present in AAD, created by BIMCO for this purpose and each caller must have its own secret. The following article describes the setup:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/v1-oauth2-client-creds-grant-flow>

To get a secret please forward a request to marinfo@bimco.org / holidays@bimco.org

JWT Token

In order for the caller to obtain a JWT access token the caller must have the following information:

- Client Secret
- Client ID

Create a POST against <https://login.microsoftonline.com/1b2dc093-9347-4783-9cec-e0a6b791acdf/oauth2/v2.0/token>

with the following x-www-form-urlencoded values

KEY	VALUE
client_id	https://bimcodataservices.onmicrosoft.com/684726db-8f1a-48fd-a3a8-3af3c608c499

scope	https://bimcodataservices.onmicrosoft.com/684726db-8f1a-48fd-a3a8-3af3c608c499/.default
client_secret	[value provided by BIMCO]
Grant_type	client_credentials

Posting will result in a Bearer token returned, that must be stored and kept for calls the API. Access tokens are short-lived, so this must be repeated when calls are made during a longer period of time.

Common HTTP Headers

All calls to the API require the same HTTP headers to be present:

- Method: POST
- Headers
 - Content-type: application/json
 - Authorization: Bearer [JWT-token obtained in previous step]

Base API responses

All requests to the API, where a valid JWT token is not present, will receive HTTP status: 401 Unauthorized

When a valid JWT token is present, all responses from the API will always contain 2 properties:

- Success (boolean)
- Message (String)

Success will indicate the overall success of the request. Message will contain any error messages to be inspected in case of no success, or simply a message informing that the request went as expected.

Structure

The structure internally kept with BIMCO for holding holiday information is constructed in the following way:

- Country
 - Region [Optional - only for certain countries]
 - Port
 - Port

Under each country, region or port up to 2 identifiers are available:

- Locode
- ExternalReference

Both of these identifiers are valid when querying for information, and in some cases only one identifier will be present pr. port/region/country

Request to obtain the structure

The structure can be downloaded using the following request:

- Method: POST
- URL: <https://membersapi.bimco.org/api/structure>
- Payload:


```
{
    "RegistrationNumber" : "[Your BIMCO registration number]",
    "Secret" : "[Same secret as provided when obtaining bearer token]"
  }
```

Return values

Success	Message	Information	Description
False	You are not authorized to access this information	None	You have not provided a valid JSON payload including RegistrationNumber and Secret
True	Information fetched OK	Escaped JSON list of countries, regions and ports	Request executed OK

Request to fetch holiday information

After inspecting the structure and pinpointing the required identifier for a specific port/region/country, the information can now be fetched using the following request:

- Method: POST
- URL: <https://membersapi.bimco.org/api/holiday/information>
- Payload:


```
{
    "RegistrationNumber" : "[Your BIMCO registration number]",
    "Secret" : "[Same secret as provided when obtaining bearer token]",
    "LocodeOrExternalReference" : "[identifier]"
  }
```

This will result in a JSON object containing the holiday information matching the supplied Identifier

Return values

Success	Message	Information	Description
False	You are not authorized to access this information	None	The request did not contain a valid combination of registration-number and secret
False	No locode or identifier provided in request	None	No locode/identifier was provided in the request
False	No country, region or port could be found matching locode/identifier	None	The locode/identifier provided in the request does not match the values present in the structure
False	No country holiday information found	None	No holiday information found for country

False	No region holiday information found	None	No holiday information found for region
False	No port holiday information found	None	No Port Working Hours information found for port
True	Information fetched OK	Escaped JSON holiday/PWH information	Request executed OK

Example requests

3 example queries are available for testing the output of the API

- Output generated for country holidays
- Output generated for region holidays
- Output generated for port working hours

When using the example queries, use the "ExampleType" parameter to specify the desired output type. Possible values are: "Country", "Region" or "Port"

- Method: POST
- URL: <https://membersapi.bimco.org/api/holiday/example>
- Payload:


```
{
  "RegistrationNumber" : "[Your BIMCO registration number]",
  "Secret" : "[Same secret as provided when obtaining bearer token]",
  "ExampleType" : "Country"
}
```

Return values

Success	Message	Information	Description
False	You are not authorized to access this information	None	The request did not contain a valid combination of registration-number and secret
True	Information fetched OK	Escaped JSON holiday/PWH information	Request executed OK

JSON schemes and enums

Above example methods can also be used to build a JSON scheme using tools such as <https://jsonschema.net/>

If a tool is needed for unescaping JSON, this website can be used: <https://www.freeformatter.com/json-escape.html>

Please find below the list of enums used in the different output

Output	Enum	Values
Port Working Hours	ActivityType	Normal Break OvertimePlanned, OvertimeByRequest, NoWork
Port Working Hours	EntityType	Port Terminal Operator

Understanding the data

Country and region holidays

Root array "Holidays" contains 1 or more elements with the following properties

- StartDate (DateTime)
- EndDate (DateTime)
- Description (String)
- Comments (String)

Root array "Notes" contains 0 or more elements with the following properties

- Name (String)
- Content (String)

(Notes are not specific to a single holiday)

Example:

```
{
  "Holidays": [
    {
      "StartDate": "17-12-2019",
      "EndDate": "18-12-2019",
      "Description": "Country Holiday example",
      "Comment": "No comments"
    }
  ],
  "Notes": [
    {
      "Name": "Test Note",
      "Content": "Test Note Content"
    }
  ]
}
```

Port Working Hours

Port Working Hours is a complex structure, where the top level object is a list of "Entities". Each entity represents the port itself, a terminal or an operator under port or terminal. The type of the entity is specified in the "Type" property, and the hierarchy is specified using the "Parent" property on "Entity".

Under each entity you will find:

1. A list of weekdays.
2. A list of holidays
3. A list of notes

Weekdays and activities

Array: "Weekdays" will normally contain the 7 weekdays, each identified by the actual name of the weekday. Each weekday will have an "ActivityList" array, that will have 0 or more "Activity" elements, which sums up the activities for the given port/terminal/operator for the given weekday. Each activity has the following properties:

- StartTime (TimeSpan)
- EndTime (TimeSpan)
- ActivityType (Enum as described previously)
- Rate
 - Used to indicate any overtime rate that may be applied if activity is of type: OverTimePlanned or OvertimeByRequest
- Comments
 - Any comments relevant for the activity
- Group
 - Generic property that can be used to describe contexts such as working in different shifts

Please note that overlapping activities may occur, especially if "shifts" or other "Group" definitions are used

Please note that activities may span multiple weekdays, fx when starting 23:00:00 and finishing at 07:00:00. An activity is only visible on the weekday that defines its start-time

Holidays

Array "Holidays" contains a list of 0 or more "Holiday" entries. These holidays may be relevant only for the port, or inherited from the region/country found in the BIMCO database. Each "Holiday" will have the following properties:

- Name (String)
- StartDate (Date)
- EndDate (Date)
- ActivityList (Array of 0 or more activities during this specific holiday)

Activities found for a specific holiday should be considered more "relevant" than activities defined on a regular weekday

Activities for a specific holiday contains the same properties as activities found under weekdays, with the following 2 exceptions:

- StartTime is now StartDateTime (DateTime instead of Timespan)

- EndTime is now EndDateTime (DateTime instead of Timespan)

Please note that the StartDateTime and EndDateTime properties of the individual activity may extend further than the actual start/end date of the holiday.

Notes

Array "Notes" contains a list of 0 or more "Note" entries. Each note contains the following properties:

- Name
- Content

Notes are used for information that does not otherwise fit into the complex structure.

Example:

```
{
  "Entities": [
    {
      "Name": "Test Port",
      "Type": "Port",
      "Parent": "",
      "Weekdays": [
        {
          "Day": "Monday",
          "ActivityList": [
            {
              "StartTime": "07:00:00",
              "EndTime": "16:00:00",
              "ActivityType": "Normal",
              "Rate": "n/a",
              "Comments": "Normal work",
              "Group": ""
            }
          ]
        },
        {
          "Day": "Tuesday",
          "ActivityList": [
            {
              "StartTime": "07:00:00",
              "EndTime": "16:00:00",
              "ActivityType": "Normal",
              "Rate": "n/a",
              "Comments": "Normal work",
              "Group": ""
            }
          ]
        }
      ]
    }
  ],
}
```

```
{
  "Day": "Wednesday",
  "ActivityList": [
    {
      "StartTime": "07:00:00",
      "EndTime": "16:00:00",
      "ActivityType": "Normal",
      "Rate": "n/a",
      "Comments": "Normal work",
      "Group": ""
    }
  ]
},
{
  "Day": "Thursday",
  "ActivityList": [
    {
      "StartTime": "07:00:00",
      "EndTime": "16:00:00",
      "ActivityType": "Normal",
      "Rate": "n/a",
      "Comments": "Normal work",
      "Group": ""
    }
  ]
},
{
  "Day": "Friday",
  "ActivityList": [
    {
      "StartTime": "07:00:00",
      "EndTime": "16:00:00",
      "ActivityType": "Normal",
      "Rate": "n/a",
      "Comments": "Normal work",
      "Group": ""
    }
  ]
},
{
  "Day": "Saturday",
  "ActivityList": [
    {
      "StartTime": "00:00:00",
      "EndTime": "00:00:00",
      "ActivityType": "OvertimePlanned",
      "Rate": "25",
      "Comments": "Planned overtime at an additional 25 percent",
    }
  ]
}
```

```
    "Group": ""
  }
]
},
{
  "Day": "Sunday",
  "ActivityList": [
    {
      "StartTime": "00:00:00",
      "EndTime": "00:00:00",
      "ActivityType": "OvertimePlanned",
      "Rate": "25",
      "Comments": "Planned overtime at an additional 25 percent",
      "Group": ""
    }
  ]
}
],
"Holidays": [
  {
    "Name": "Local port holiday",
    "StartDate": "16-12-2019",
    "EndDate": "16-12-2019",
    "ActivityList": [
      {
        "StartDateTime": "16-12-2019 00:00:00",
        "EndDateTime": "16-12-2019 07:00:00",
        "ActivityType": "NoWork",
        "Rate": "n/a",
        "Comments": "",
        "Group": ""
      },
      {
        "StartDateTime": "16-12-2019 07:00:00",
        "EndDateTime": "16-12-2019 00:00:00",
        "ActivityType": "OvertimeByRequest",
        "Rate": "25",
        "Comments": "",
        "Group": ""
      }
    ]
  }
]
},
"Notes": []
},
{
  "Name": "Test Terminal",
  "Type": "Terminal",
```

```
"Parent": "Test Port",
"Weekdays": [
  {
    "Day": "Monday",
    "ActivityList": [
      {
        "StartTime": "07:00:00",
        "EndTime": "16:00:00",
        "ActivityType": "OvertimeByRequest",
        "Rate": "20",
        "Comments": "Overtime by request",
        "Group": ""
      }
    ]
  },
  {
    "Day": "Tuesday",
    "ActivityList": [
      {
        "StartTime": "07:00:00",
        "EndTime": "16:00:00",
        "ActivityType": "OvertimeByRequest",
        "Rate": "20",
        "Comments": "Overtime by request",
        "Group": ""
      }
    ]
  },
  {
    "Day": "Wednesday",
    "ActivityList": [
      {
        "StartTime": "07:00:00",
        "EndTime": "16:00:00",
        "ActivityType": "OvertimeByRequest",
        "Rate": "20",
        "Comments": "Overtime by request",
        "Group": ""
      }
    ]
  },
  {
    "Day": "Thursday",
    "ActivityList": [
      {
        "StartTime": "07:00:00",
        "EndTime": "16:00:00",
        "ActivityType": "OvertimeByRequest",
```

```
    "Rate": "20",
    "Comments": "Overtime by request",
    "Group": ""
  }
]
},
{
  "Day": "Friday",
  "ActivityList": [
    {
      "StartTime": "07:00:00",
      "EndTime": "16:00:00",
      "ActivityType": "OvertimeByRequest",
      "Rate": "20",
      "Comments": "Overtime by request",
      "Group": ""
    }
  ]
},
{
  "Day": "Saturday",
  "ActivityList": [
    {
      "StartTime": "00:00:00",
      "EndTime": "00:00:00",
      "ActivityType": "NoWork",
      "Rate": "n/a",
      "Comments": "",
      "Group": ""
    }
  ]
},
{
  "Day": "Sunday",
  "ActivityList": [
    {
      "StartTime": "00:00:00",
      "EndTime": "00:00:00",
      "ActivityType": "NoWork",
      "Rate": "n/a",
      "Comments": "",
      "Group": ""
    }
  ]
}
],
"Holidays": [],
"Notes": [
```

```
{
  "Name": "Test note",
  "Content": "This is a test entity"
}
]
```