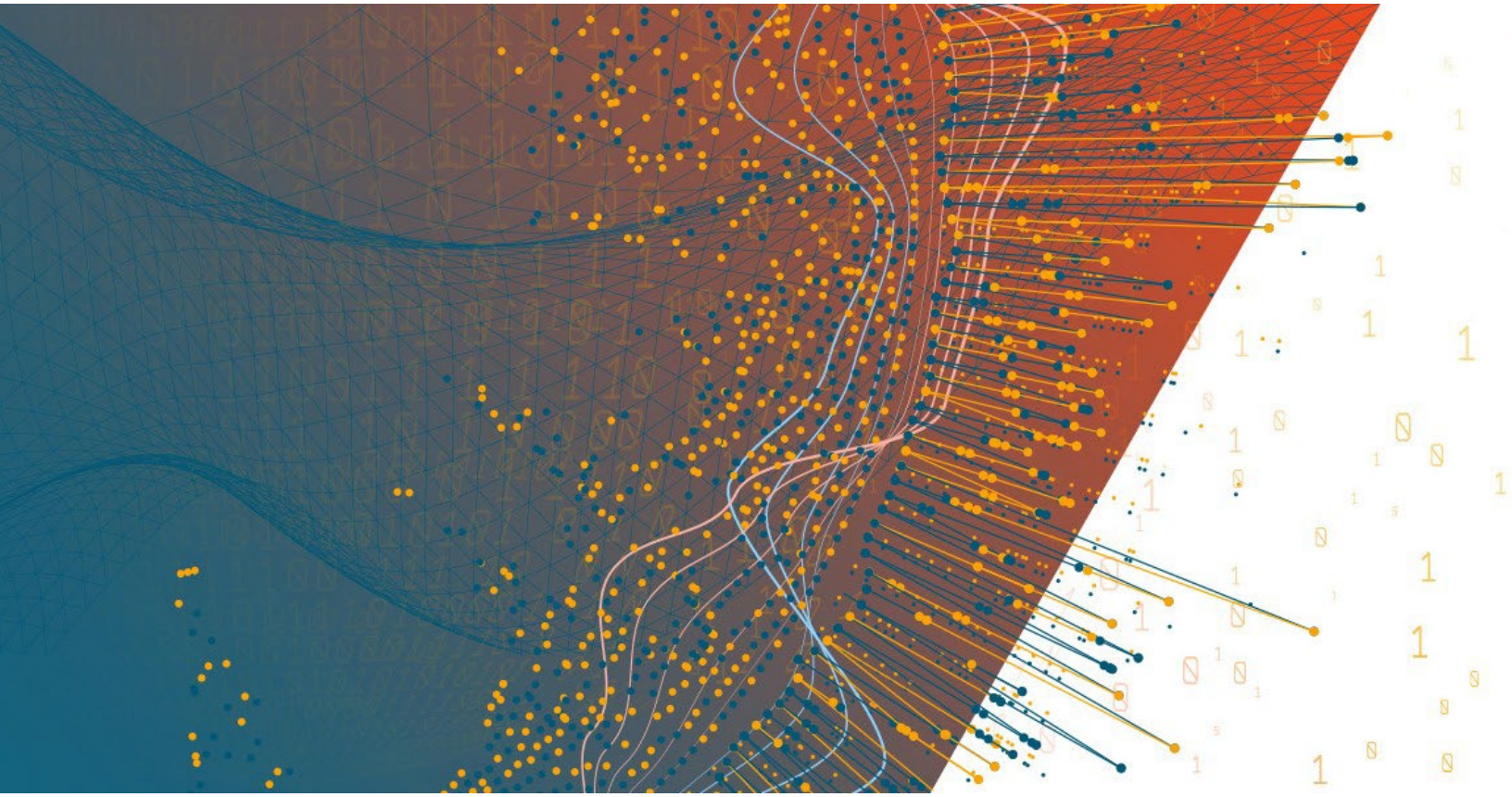


ALTAIR

Altair® Monarch Server™ v2025.0
API Guide

TABLE OF CONTENTS

INTRODUCTION

Monarch Server provides the API for external applications based on the Hypertext Transfer Protocol (HTTP).

PRINCIPLE AND PURPOSE

The API allows external applications to use various functions of Monarch Server and access the Monarch Server database data. Using the API, a new front-end or automation application can be built easily on top of Monarch Server.

MONARCH SERVER API

This document contains the following sections:

VISUAL PROCESS DESIGNER API

API TESTING

You can test how the API works in the following ways:

- On the interactive API help page at

  `http://localhost/MSAdmin/api/help`

  On this page, you can view detailed information about each method and test all the methods by clicking the **Test API** button.

> **NOTE** In Monarch Server v2025.0 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as “DSAdmin” (or "DSClient"). In this case, use “DSAdmin” (or "DSClient”) instead of “MSAdmin” (or “MSClient”) to open the Admin (or Client) page.

- Via Swagger at

  `http://localhost/MSAdmin/swagger`

WORKING WITH THE MONARCH SERVER API

To work with the Monarch Server API successfully, review the information below.

Request authentication

For authentication, use the Login method.

API entry point

The Monarch Server HTTP API is accessed through an entry point.

The format of the request URL is

`http://<host>:<port>/MSAdmin/api`

Here, `<host>` is the name of the computer where Monarch Server is running, and `<port>` is the port number on which the Monarch Server is listening.

The request can be sent using either an HTTP GET or an HTTP POST method (unless specified otherwise). The target area can be either the top window or a new browser window.

Parameter values that contain URL-prohibited symbols must be encoded as required by the URL specification.

Some parameters depend on a particular operation type.

Response type and error handling

A response to a request can be either one of the following depending on the success condition:

- Request succeeded (the response has a corresponding MIME-type or an HTTP redirect)
- Request failed

Depending on the operation nature and status, the response can be one of the following:

- Standard Monarch Server error page describing the nature of the error (for HTML-based views)
- HTTP error code with an optional message
- XML error description (non-HTML views)

In case of an XML error description, the response type is text/xml. The format of an XML error message is

```
<Error>
 <Message>[Message]</Message>
 <ExceptionMessage>[ExceptionMessage]</ExceptionMessage>
 <ExceptionType>[ExceptionType]</ExceptionType>
 <StackTrace>[StackTrace]</StackTrace>
</Error>
```

The admin application can check the root node's name if text/xml content is returned to determine whether an error occurred.

Monarch Server API Usage Sample

Use the Login method for user authentication.

```
http://localhost/MSAdmin/api/login
```

Upon logging in, the user can either stay in the system or leave via the Logout method.

```
http://localhost/MSAdmin/api/logout?redirectUrl={redirectUrl}
```

The user can use different methods to get information about a visual process. See example below.

1. Use the Start Process method to start a visual process and get a tracking ID.

   http://localhost/MSAdmin/api/v1/visualprocesses/start

2. Then, use the Get Status method to get the status of the visual process.

   http://localhost/MSAdmin/api/v1/visualprocesses/{trackingId}/status

3. Use the Get Log method to get a log of the visual process.

   http://localhost/MSAdmin/api/v1/visualprocesses/{trackingId}/log

The user can utilize the Visual Process Designer API Samples installed using the Single Installer.

API REFERENCE

The following is a complete reference of the methods supported by the Monarch Server API.

Login

Use the following methods to begin and end a user session to work with Monarch Server.

Logging In

Used to establish a “session” — a secure environment for further operation.

POST Method

The entry point for this method is

```
api/login
```

Request Information

URI Parameters: None

The body parameters are as follows:

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
Username	Username	String	Required
Password	Password	String	None
Domain	User domain	String	None

Request format sample (application/json, text/json):

```
{
  "Username": "sample string 1",
  "Password": "sample string 2",
  "Domain": "sample string 3"
}
```

Request format sample (application/xml, text/xml):

```
<LoginRequest xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2004/07/Datawatch.Web.EnterpriseServer.Common.Models.Api">
  <Domain>sample string 3</Domain>
  <Password>sample string 2</Password>
  <Username>sample string 1</Username>
</LoginRequest>
```

Autologin

Used to establish a “session” when the Active directory authentication is enabled with the selected SSO option.

POST method

The entry point for this method is

```
api/autologin
```

Request Information

URI Parameters: None

Body Parameters: None

Logging out

This operation terminates the current API session. It can be useful when the maximum number of user sessions is limited.

GET method

The entry point for this method is

```
api/logout?redirectUrl={redirectUrl}
```

Request Information

The URI parameter is as follows:

Name	Description	Type	Additional information
redirectUrl	URL for redirection after session termination. If this parameter is absent, then no redirect is sent.	String	None

Body Parameters: None

ProcessDesigner

This allows the use of Visual Process Designer as a web service.

Start Process

This initiates a visual process.

POST method

The entry point for this method is

```
api/v1/visualprocesses/start
```

Request Information

URI parameters: None

Body parameters: None

Request format sample (application/json, text/json)

```
"sample string 1"
```

Here, “sample string 1” is a name of a visual process.

Request format sample (application/xml, text/xml)

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">sample string 1</string>
```

Response Information

Resource Description: None

Start Process Using Manifest

This initiates a visual process as described by a given manifest.

POST method

The entry point for this method is

```
api/v1/visualprocesses/startwithmanifest
```

Request Information

URI Parameters: None

Body parameters: None

Request format sample (application/json, text/json)

```
"sample string 1"
```

For Model, Input, RunTime fields, Export, Credentials, DataPrep items, and Burster, "sample string 1" may be replaced as follows.

Model:

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
{
      "ItemType": "ModelFile",
      "Model": {
        "Location": "sample string 3",
        "Name": "sample string 2",
         "ExternalLookupOption": [
{
                  "Name": "sample string 4",
                  "Password": "sample string 5"
}
]
      }
    }
]
}
```

Input:

File input

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
{
      "ItemType": "InputFile",
      "FileInput": {
        "FileLocation": "sample string 3",
        "PdfPassword": "sample string 4",
        "Name": "sample string 2"
      }
    }
]
}
```

Web Input

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
{
      "ItemType": "InputWeb",
      "WebInput": {
        "Location": "sample string 6",
        "PdfPassword": "sample string 7",
        "Name": "sample string 5"
      }
    }
]
}
```

Database Input

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "InputDatabase",
      "DatabaseInput": {
        "DataSource": "sample string 9",
        "Password": "sample string 10",
        "TableName": "sample string 11",
        "Filter": "sample string 12",
        "ReplaceNulls": true/false,
        "FirstRowContainsNames": true/false,
         "TextParameters": {
          "LinesToIgnoreAtStart": int,
         "AllowEmbeddedLineBreaks": true/false,
          "Delimiter": "sample string 13",
          "TextQualifier": "sample string 14",
          "TextEncoding": "sample string 15"
        },
        "Name": "sample string 8"
      }
    }
]
}
```

Report Warehouse Input

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "InputContentSystem",
      "ContentSystemInput": {
        "Since": "sample string 14",
        "Till": "sample string 15",
        "DocType": {
          "Id": 1
        },
        "SelectCriteria": "ByDate",
        "DateType": "FilingDate",
        "Name": "sample string 13"
      }
    }
]
}
```

Runtime field:

```
{
"ProcessName": "sample string 1",

         "runtime_fields": [
                 {
                 "name": "sample string 2",
                 "value": "sample string 3"
                 }
         ]
}
```

Credentials:

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "Credentials",
      "Credentials": {
        "HttpCredential": {
          "UserName": "sample string 3",
          "Password": "sample string 4",
          "Domain": "sample string 5",
          "UseCurrent": false/true
        },
        "FtpCredential": {
          "UserName": "sample string 6",
          "Password": "sample string 7",
          "Domain": "sample string 8",
          "UseCurrent": false/true
        },
        "S3Credential": {
          "UserName": "sample string 9",
          "Password": "sample string 10",
          "Domain": "sample string 11",
          "UseCurrent": false/true
        },
        "Name": "sample string 2"
      }
```

```
    }
]
}
```

Exports

Report Export

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "ExportReport",
      "ExportReport": {
        "Destination": "sample string 3",
        "ExportFileExistingOptions": "Overwrite",
        "Name": "sample string 2"
      }
    }
]
}
```

Table Export

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "ExportTable",
      "ExportTable": {
        "OutputType": "FileSystem/Database",
        "Destination": "sample string 3",
        "FileName": "sample string 4",
        "ExportFileType": "Unspecified",
        "WhenFileExist": "Overwrite",
         "ConnectionString": "sample string 5",
        "ConnectionStringPassword": "sample string 6",
        "TableName": "sample string 7",
        "WhenTableExist": "Overwrite",
        "ByFilesNaming": "Filter",
         "ByTablesNaming": "Unspecified",
        "CurrentSort": {
          "Name": "sample string 8",
          "NoSort": false/true
        },
        "CurrentFilter": {
          "Name": "sample string 9",
          "NoFilter": false/true,
          "AllFilters": true/false
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Summary Export

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
```

```
 {
      "ItemType": "ExportSummary",
      "ExportSummary": {
        "OutputType": "FileSystem/Database",
        "Destination": "sample string 3",
        "FileName": "sample string 4",
        "ExportFileType": "Unspecified",
        "WhenFileExist": "Overwrite",
         "ConnectionString": "sample string 5",
        "ConnectionStringPassword": "sample string 6",
        "TableName": "sample string 7",
        "WhenTableExist": "Overwrite",
        "ByFilesNaming": "Filter",
         "ByTablesNaming": "Unspecified",
        "CurrentSort": {
          "Name": "sample string 8",
          "NoSort": false/true
        },
        "CurrentFilter": {
          "Name": "sample string 9",
          "NoFilter": false/true,
          "AllFilters": true/false
        },
         "CurrentSummary": {
          "Name": "sample string 10",
          "AllSummaries": false/true
        },
        "Name": "sample string 2"
      }
    }
]
}
```

DataPrep Items

Workspace

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
    {
      "ItemType": "DataPrepWorkspace",
      "DataPrepWorkspaceItem": {
        "WorkspacePath": "sample string 3",
        "LoadPlans": [
          {
            "TableName": "sample string 4",
            "LoadPlanName": "sample string 5"
          },
          {
            "TableName": "sample string 6",
            "LoadPlanName": "sample string 7"
          }
        ],
        "Name": "sample string 2"
       }
    }
  ]
}
```

DataPrep Export

CSV

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": true/false,
        "ExportPlanNames": [ "sample string 3", "sample string 4"],
        "ExportType": "Delimited",
        "DelimitedExportOptions": {
          "IncludeHeader": true/false,
          "Delimiter": "sample string 6",
          "Qualifier": "sample string 7",
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Tableau

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
{
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "Tableau",
        "TableauExportOptions": {
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Qlik

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
```

```
          "sample string 4"
        ],
        "ExportType": "Qlik",
        "QlikExportOptions": {
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Excel

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "MicrosoftExcel",
        "ExcelExportOptions": {
          "TableName": "sample string 6",
          "WhenTableExist": "Overwrite",
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Designer

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "DatawatchDesigner",
        "DesignerExportOptions": {
          "TableName": "sample string 6",
          "WhenTableExist": "Overwrite",
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
```

```
    }
]
}
```

Access

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "MicrosoftAccess",
        "AccessExportOptions": {
          "TableName": "sample string 6",
          "WhenTableExist": "Overwrite",
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

OleDB

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "OleDb",
        "OleDbExportOptions": {
          "IncludePassword": true/false,
          "Password": "sample string 5",
          "TableName": "sample string 6",
          "WhenTableExist": "Overwrite",
          "Destination": "sample string 7"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Cognos

```
{
"ProcessName": "sample string 1",
```

```
  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "Cognos",
        "CognosExportOptions": {
          "ServerUrl": "sample string 6",
         "Namespace": "sample string 7",
         "UserName": "sample string 8",
          "Password": "sample string 9",
          "IsAnonymous": false/true,
          "TableName": "sample string 5",
          "WhenTableExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Power BI

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
         "ExportType": "PowerBI",
        "PowerBIExportOptions": {
          "AccessToken": "sample string 3",
         "RefreshToken": "sample string 4",
          "TableName": "sample string 5",
          "WhenTableExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Knowledge Studio

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
```

```
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "Angoss",
        "AngossExportOptions": {
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Fixed

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
 {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "Fixed",
        "FixedExportOptions": {
          "Destination": "sample string 5",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
}
```

SAS

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "DataPrepExport",
      "DataPrepExportItem": {
        "AllLoadPlans": false/true,
        "ExportPlanNames": [
          "sample string 3",
          "sample string 4"
        ],
        "ExportType": "Sas",
        "SasExportOptions": {
          "TableName": "sample string 5",
          "Destination": "sample string 6",
          "WhenFileExist": "Overwrite"
        },
        "Name": "sample string 2"
      }
    }
]
```

```
}
```

File Burster

```
{
"ProcessName": "sample string 1",

  "ProcessItems": [
  {
      "ItemType": "FileBurster",
      "FileBurster": {
        "DestinationPath": "sample string 3",
        "OutputFileNameMask": "sample string 4",
        "UseHeaderOutputMask": true/false,
        "HeaderOutputMask": "sample string 5",
        "ErrorDestination": "sample string 6",
        "UseRecognizerLineKeyMask": true/false,
        "RecognizerLineKeyMask": "sample string 7",
        "RecognizerLineKeyMaskOffset": int,
        "IsTrimLeadingCharacters": true/false,
        "TrimLeadingCharacters": int,
        "BurstType": "Defined/Pages",
        "DefinitionBurstOptions": {
          "IndexTableStartMarker": "sample string 8",
          "LineValidator": "sample string 9",
          "TableEndMarker": "sample string 10",
          "ContentStartMarker": "sample string 11",
          "IndexTableStartMarkerOffset": int,
          "LineValidatorOffset": int,
          "LineFieldOffset": int,
          "LineFieldLength": int,
          "IdFieldOffset": int,
          "IdFieldLength": int,
          "TableEndMarkerOffset": int,
          "ContentStartMarkerOffset": int
        },
         "PageBurstOptions": {
          "ModelPath": "sample string 12",
          "PagesPerOutputFile": int,
          "ProcessSuccessiveFilesAsOne": true/false
        },
        "Name": "sample string 2"
      }
    }
]
}
```

Request format sample (application/xml, text/xml)

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">sample string
1</string>
```

For Model, Input, RunTime fields, Export, Credentials, DataPrep items, and Burster, “sample string 1” may be replaced as follows.

Model:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
    <ProcessItem>
```

```
      <ItemType>ModelFile</ItemType>
         <Model>
          <Name>sample string 2</Name>
          <Location>sample string 3</Location>
          <ExternalLookupOptions>
           <ExternalLookupOption>
             <Name>sample string 4</Name>
             <Password>sample string 5</Password>
           </ExternalLookupOption>
          </ExternalLookupOptions>
        </Model>
</ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>
```

NOTE Sample string 3 (Location) is the file location, definition ID, or web address. Its value depends on the model item type.

Input:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
<!—File Input-->
    <ProcessItem>
      <ItemType>InputFile</ItemType>
      <FileInput>
        <Name>sample string 2</Name>
        <FileLocation>sample string 3</FileLocation>
        <PdfPassword>sample string 4</PdfPassword>
      </FileInput>
</ProcessItem>

<!—Web Input-->
   <ProcessItem>
    <ItemType>InputFile</ItemType>
      <WebInput>
        <Name>sample string 5</Name>
        <FileLocation>sample string 6</FileLocation>
        <PdfPassword>sample string 7</PdfPassword>
      </WebInput>
  </ProcessItem>

<!—Database Input-->
  <ProcessItem>
    <ItemType>InputFile</ItemType>
      <DatabaseInput>
        <Name>sample string 8</Name>
        <DataSource>sample string 9</DataSource>
        <Password>sample string 10</Password>
        <TableName>sample string 11</TableName>
        <Filter>sample string 12</Filter>
        <ReplaceNulls>true/false</ReplaceNulls>
        <FirstRowContainsNames>true/false</FirstRowContainsNames>
        <TextParameters>
           <LinesToIgnoreAtStart>int</LinesToIgnoreAtStart>
           <AllowEmbeddedLineBreaks>true/false</AllowEmbeddedLineBreaks>
           <Delimiter>sample string 13</Delimiter>
           <TextQualifier>sample string 14</TextQualifier>
           <TextEncoding>sample string 15</TextEncoding>
         </TextParameters>
```

```
      </DatabaseInput>
 </ProcessItem>

<!—ContentSystem Input-->
  <ProcessItem>
   <ItemType>InputFile</ItemType>
    <ContentSystemInput>
        <Name>sample string 13</Name>
        <Since>sample string 14</Since>
        <Till>sample string 15</Till>
        <DocType>
          <Id>1</Id>
        </DocType>
        <SelectCriteria>ByDate </SelectCriteria>
        <DateType>FilingDate</DateType>
    </ContentSystemInput>
  </ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>
```

Runtime field:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
<RuntimeFields>
   <RuntimeField>
       <Name>sample string 2</Name>
       <Value>sample string 3</Value>
    </RuntimeField>
  </RuntimeFields>
</ProcessStartParams>]]>
```

Credentials:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
<ProcessItems>
<!—FTP Credentials-->
  <ProcessItem>
   <ItemType>Credentials</ItemType>
   <Credentials>
     <Name>sample string 2</Name>
          <FtpCredential>
            <UserName>sample string 3</UserName>
            <Password>sample string 4</Password>
            <Domain>sample string 5</Domain>
            <UseCurrent>true</UseCurrent>
          </FtpCredential>
     </Credentials>
  </ProcessItem>

<!—HTTP Credentials-->
  <ProcessItem>
   <ItemType>Credentials</ItemType>
      <Credentials>
        <Name>sample string 6</Name>
          <HttpCredential>
            <UserName>sample string 7</UserName>
            <Password>sample string 8</Password>
            <Domain>sample string 9</Domain>
            <UseCurrent>true</UseCurrent>
          </HttpCredential>
        </Credentials>
   </ProcessItem>
```

```
<!-S3 Credentials-->
    <ProcessItem>
      <ItemType>Credentials</ItemType>
      <Credentials>
        <Name>sample string 10</Name>
        <S3Credential>
          <UserName>sample string 11</UserName>
          <Password>sample string 12</Password>
          <Domain>sample string 13</Domain>
          <UseCurrent>true or false</UseCurrent>
        </S3Credential>
      </Credentials>
    </ProcessItem>
  </ProcessItems>
</ProcessStartParams>]]>
```

Exports:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
<!—Report export-->
   <ProcessItem>
     <ItemType>ExportReport</ItemType>
        <ExportReport>
          <Name>sample string 2 </Name>
          <Destination>sample string 3</Destination>
          <ExportFileExistingOptions>Overwrite</ExportFileExistingOptions>
        </ExportReport>
   </ProcessItem>

<!—Table export-->
   <ProcessItem>
    <ItemType>ExportTable</ItemType>
      <ExportTable>
        <Name>sample string 4</Name>
        <OutputType>FileSystem/Database</OutputType>
        <Destination>sample string 5</Destination>
       <FileName>sample string 6</FileName>
        <ExportFileType>Unspecified</ExportFileType>
        <WhenFileExist>Overwrite</WhenFileExist>
        <ConnectionString>sample string 7</ConnectionString>
        <ConnectionStringPassword>sample string 8</ConnectionStringPassword>
        <TableName>sample string 9</TableName>
        <WhenTableExist>Overwrite</WhenTableExist>
        <ByFilesNaming>Unspecified</ByFilesNaming>
        <ByTablesNaming>Filter</ByTablesNaming>
        <CurrentSort>
          <Name>sample string 10</Name>
          <NoSort>false</NoSort>
        </CurrentSort>
        <CurrentFilter>
          <Name>sample string 11</Name>
          <NoFilter>false</NoFilter>
          <AllFilters>false</AllFilters>
        </CurrentFilter>
      </ExportTable>
  </ProcessItem>

<!—Summary export-->
<ProcessItem>
  <ItemType>ExportSummary</ItemType>
      <ExportSummary>
        <Name>sample string 12</Name>
```

```
        <OutputType>FileSystem/Database</OutputType>
        <Destination>sample string 13</Destination>
        <FileName>sample string 14</FileName>
        <ExportFileType>Unspecified</ExportFileType>
        <WhenFileExist>Overwrite</WhenFileExist>
        <ConnectionString>sample string 15</ConnectionString>
        <ConnectionStringPassword>sample string 16</ConnectionStringPassword>
        <TableName>sample string 17</TableName>
        <WhenTableExist>Overwrite</WhenTableExist>
        <ByFilesNaming>Unspecified</ByFilesNaming>
        <ByTablesNaming>Filter</ByTablesNaming>
        <CurrentSort>
           <Name>sample string 18</Name>
           <NoSort>true</NoSort>
        </CurrentSort>
        <CurrentFilter>
           <Name>sample string 19</Name>
           <NoFilter>true</NoFilter>
           <AllFilters>true</AllFilters>
        </CurrentFilter>
        <CurrentSummary>
           <Name>sample string 20</Name>
           <AllSummaries>false</AllSummaries>
        </CurrentSummary>
      </ExportSummary>
  </ProcessItem>
</ProcessItems>
</ProcessStartParams>]]>
```

DataPrep items:
Workspace:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
    <ProcessItem>
      <ItemType>DataPrepWorkspace</ItemType>
      <DataPrepWorkspaceItem>
        <Name>sample string 2</Name>
        <WorkspacePath>sample string 3</WorkspacePath>
        <LoadPlans>
          <LoadPlan>
            <TableName>sample string 4</TableName>
            <LoadPlanName>sample string 5</LoadPlanName>
          </LoadPlan>
          <LoadPlan>
            <TableName>sample string 6</TableName>
            <LoadPlanName>sample string 7</LoadPlanName>
          </LoadPlan>
        </LoadPlans>
      </DataPrepWorkspaceItem>
    </ProcessItem>
  </ProcessItems>
</ProcessStartParams>]]>
```

DataPrep export:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
<!—CSV export-->
    <ProcessItem>
     <ItemType>DataPrepExport</ItemType>
```

```
      <DataPrepExportItem>
        <Name>sample string 2</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Delimited</ExportType>
        <DelimitedExportOptions>
           <Destination>sample string 4</Destination>
           <!--For MS v13.2 and 13.5-->
            <OverwriteFile>true</OverwriteFile>
            <!--For MS v14.0+-->
            <WhenFileExist>Overwrite</WhenFileExist>
           <IncludeHeader>true</IncludeHeader>
           <Delimiter>sample string 5</Delimiter>
           <Qualifier>sample string 6</Qualifier>
        </DelimitedExportOptions>
      </DataPrepExportItem>
   </ProcessItem>

<!—Tableau export-->
 <ProcessItem>
    <ItemType>DataPrepExport</ItemType>
       <DataPrepExportItem>
         <Name>sample string 7</Name>
         <AllLoadPlans>false/true</AllLoadPlans>
          <ExportPlanNames>sample string1</ExportPlanNames>
          <ExportPlanNames>sample string2</ExportPlanNames>
         <ExportType>Tableau</ExportType>
         <TableauExportOptions>
            <Destination>sample string 9</Destination>
             <!--For MS v13.2 and 13.5-->
             <OverwriteFile>true</OverwriteFile>
             <!--For MS v14.0+-->
             <WhenFileExist>Overwrite</WhenFileExist>
         </TableauExportOptions>
       </DataPrepExportItem>
   </ProcessItem>

<!—Qlik export-->
  <ProcessItem>
   <ItemType>DataPrepExport</ItemType>
       <DataPrepExportItem>
        <Name>sample string 10</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Qlik</ExportType>
        <QlikExportOptions>
           <Destination>sample string 12</Destination>
            <!--For MS v13.2 and 13.5-->
            <OverwriteFile>true</OverwriteFile>
            <!--For MS v14.0+-->
            <WhenFileExist>Overwrite</WhenFileExist>
        </QlikExportOptions>
      </DataPrepExportItem>
   </ProcessItem>

<!—Excel export-->
  <ProcessItem>
    <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 13</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
```

```
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>MicrosoftExcel</ExportType>
        <ExcelExportOptions>
           <Destination>sample string 15</Destination>
             <!--For MS v13.2 and 13.5-->
             <OverwriteFile>true</OverwriteFile>
             <!--For MS v14.0+-->
             <WhenFileExist>Overwrite</WhenFileExist>
           <TableName>sample string 16</TableName>
           <WhenTableExist>Overwrite</WhenTableExist>
        </ExcelExportOptions>
      </DataPrepExportItem>
   </ProcessItem>

<!—Designer export-->
   <ProcessItem>
    <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 17</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>DatawatchDesigner</ExportType>
        <DesignerExportOptions>
           <Destination>sample string 19</Destination>
             <!--For MS v13.2 and 13.5-->
             <OverwriteFile>true</OverwriteFile>
             <!--For MS v14.0+-->
             <WhenFileExist>Overwrite</WhenFileExist>
           <TableName>sample string 20</TableName>
           <WhenTableExist>Overwrite</WhenTableExist>
        </DesignerExportOptions>
      </DataPrepExportItem>
   </ProcessItem>

<!—Access export-->
  <ProcessItem>
    <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 21</Name>
         <AllLoadPlans>false/true</AllLoadPlans>
          <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>MicrosoftAccess</ExportType>
        <AccessExportOptions>
           <Destination>sample string 23</Destination>
             <!--For MS v13.2 and 13.5-->
             <OverwriteFile>true</OverwriteFile>
             <!--For MS v14.0+-->
             <WhenFileExist>Overwrite</WhenFileExist>
           <TableName>sample string 24</TableName>
           <WhenTableExist>Overwrite</WhenTableExist>
        </AccessExportOptions>
      </DataPrepExportItem>
   </ProcessItem>

<!—OleDb export-->
    <ProcessItem>
     <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 25</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
```

```
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>OleDb</ExportType>
        <OleDbExportOptions>
           <IncludePassword>true</IncludePassword>
           <Password>sample string 27</Password>
           <TableName>sample string 28</TableName>
           <WhenTableExist>Overwrite</WhenTableExist>
           <Destination>sample string 29</Destination>
        </OleDbExportOptions>
      </DataPrepExportItem>
    </ProcessItem>

<!—Cognos export-->
<ProcessItem>
<ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 1</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>Cognos</ExportType>
         <CognosExportOptions>
                 <Destination>sample string3</Destination>
                <!--For MS v13.2 and 13.5-->
                <OverwriteFile>true</OverwriteFile>
                <!--For MS v14.0+-->
                <WhenFileExist>Overwrite</WhenFileExist>
              <TableName>sample string 4</TableName>
              <WhenTableExist>Overwrite/Skip</WhenTableExist>
              <ServerUrl>sample string 5</ServerUrl>
              <Namespace>sample string 6</Namespace>
              <UserName>sample string 7</UserName>
              <Password>sample string 8</Password>
              <IsAnonymous>true/false</IsAnonymous>
         </CognosExportOptions>
      </DataPrepExportItem>
</ProcessItem>

<!—Power BI export-->
<ProcessItem>
<ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 1</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
         <ExportPlanNames>sample string1</ExportPlanNames>
         <ExportPlanNames>sample string2</ExportPlanNames>
        <ExportType>PowerBI</ExportType>
         <PowerBIExportOptions>
                  <Destination>sample string 3</Destination>
                    <!--For MS v13.2 and 13.5-->
                    <OverwriteFile>true</OverwriteFile>
                    <!--For MS v14.0+-->
                  <WhenFileExist>Overwrite</WhenFileExist>
                  <TableName>sample string 4</TableName>
                  <WhenTableExist>Overwrite/Skip</WhenTableExist>
                  <AccessToken>sample string 5</AccessToken>
                  <RefreshToken>sample string 6</RefreshToken>
         </PowerBIExportOptions>
      </DataPrepExportItem>
</ProcessItem>
```

```
<!-Knowledge Studio export-->
    <ProcessItem>
      <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 2</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string 3</ExportPlanNames>
        <ExportPlanNames>sample string 4</ExportPlanNames>
        <ExportType>Angoss</ExportType>
        <AngossExportOptions>
          <Destination>sample string 5</Destination>
          <WhenFileExist>Overwrite</WhenFileExist>
        </AngossExportOptions>
      </DataPrepExportItem>
    </ProcessItem>

<!-Fixed export-->
    <ProcessItem>
      <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 2</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string 3</ExportPlanNames>
        <ExportPlanNames>sample string 4</ExportPlanNames>
        <ExportType>Fixed</ExportType>
        <FixedExportOptions>
          <Destination>sample string 5</Destination>
          <WhenFileExist>Overwrite</WhenFileExist>
        </FixedExportOptions>
      </DataPrepExportItem>
    </ProcessItem>

<!-SAS export-->
    <ProcessItem>
      <ItemType>DataPrepExport</ItemType>
      <DataPrepExportItem>
        <Name>sample string 2</Name>
        <AllLoadPlans>false/true</AllLoadPlans>
        <ExportPlanNames>sample string 3</ExportPlanNames>
        <ExportPlanNames>sample string 4</ExportPlanNames>
        <ExportType>Sas</ExportType>
        <SasExportOptions>
          <Destination>sample string 5</Destination>
          <WhenFileExist>Overwrite</WhenFileExist>
          <TableName>sample string 6</TableName>
        </SasExportOptions>
      </DataPrepExportItem>
    </ProcessItem>
```

File Burster:

```
<![CDATA[<ProcessStartParams ProcessName="sample string 1">
  <ProcessItems>
    <ProcessItem>
      <ItemType>FileBurster</ItemType>
      <FileBurster>
        <Name>sample string 2</Name>
        <DestinationPath>sample string 3</DestinationPath>
        <OutputFileNameMask>sample string 4</OutputFileNameMask>
        <UseHeaderOutputMask>true/false</UseHeaderOutputMask>
        <HeaderOutputMask>sample string 5</HeaderOutputMask>
        <ErrorDestination>sample string 6</ErrorDestination>
```

```
        <UseRecognizerLineKeyMask>true/false</UseRecognizerLineKeyMask>
        <RecognizerLineKeyMask>sample string 7</RecognizerLineKeyMask>
        <RecognizerLineKeyMaskOffset>int</RecognizerLineKeyMaskOffset>
        <IsTrimLeadingCharacters>true/false</IsTrimLeadingCharacters>
        <TrimLeadingCharacters>int</TrimLeadingCharacters>
        <BurstType>Defined/Pages</BurstType>
        <DefinitionBurstOptions>
          <IndexTableStartMarker>sample string 8</IndexTableStartMarker>
          <LineValidator>sample string 9</LineValidator>
          <TableEndMarker>sample string 10</TableEndMarker>
          <ContentStartMarker>sample string 11</ContentStartMarker>
          <IndexTableStartMarkerOffset>int</IndexTableStartMarkerOffset>
          <LineValidatorOffset>int</LineValidatorOffset>
          <LineFieldOffset>int</LineFieldOffset>
          <LineFieldLength>int</LineFieldLength>
          <IdFieldOffset>int</IdFieldOffset>
          <IdFieldLength>int</IdFieldLength>
          <TableEndMarkerOffset>int</TableEndMarkerOffset>
          <ContentStartMarkerOffset>int</ContentStartMarkerOffset>
        </DefinitionBurstOptions>
        <PageBurstOptions>
          <ModelPath>sample string 12</ModelPath>
          <PagesPerOutputFile>int</PagesPerOutputFile>
          <ProcessSuccessiveFilesAsOne>true/false</ProcessSuccessiveFilesAsOne>
        </PageBurstOptions>
      </FileBurster>
    </ProcessItem>
  </ProcessItems>
</ProcessStartParams>]]>
```

Response Information

Resource Description: None

Get Process Status

This returns the status of a visual process provided by a specific tracking ID.

GET method

The entry point for this method is

`api/v1/visualprocesses/{trackingId}/status`

Request Information

The URI parameter is as follows:

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
trackingId	Tracking ID returned from the initial call to the Start Process or Start Process with Manifest methods	Globally unique identifier	None

Body Parameters: None

Response Information

Definition of returned process log status:

“0”	NotStarted
“100”	PlanBuilding
“200”	Ready
“300”	Running
“400”	Completed
“500”	PartiallyFailed
“600”	PrerequisitesFailed
“700”	ExportFailed
“800”	Cancelled
“900”	Cancelling

Get Process Log

This returns the log of a visual process.

GET method

The entry point for this method is

`api/v1/visualprocesses/{trackingId}/log`

Request Information

The URI parameter is as follows:

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
trackingId	Tracking ID returned from the initial call to the Start Process or Start Process with Manifest methods	Globally unique identifier	Required

Body Parameters: None

Response Information

Resource Description: None

NOTE The Job Log is returned in the following format:

```
<StartTime>{time of job starting}</StartTime>
  <EndTime> {time of job ending} </EndTime>
```

```
  <Status>{jobStatus}</Status>
  <Items>
    <ProcessLogItemInfo>
      <Id>{item id}</Id>
      <ExecutionState>{state of item
execution}</ExecutionState>
      <ItemLayoutSettings>{item settings}</ItemLayoutSettings>
      <Name>{Item name}</Name>
      <ItemType>{type of item}</ItemType>
      <ItemInfo>{info of item, for example:  model name, input
file name, export file name as well as number of records
exported to the output, location}</ItemInfo>
</ProcessLogItemInfo>
    </ProcessLogItemInfo>
   </Items>
  <Connections>
      <SourceLogItemId>{SourceLogItem id}</SourceLogItemId>
      <TargetLogItemId>{TargetLogItem id}</TargetLogItemId>
  </Connections>
  <ProcessId>{Id of the process}</ProcessId>
  <LogId>{Id of the current log}</LogId>
  <ServerLocation>{the name of server
location}</ServerLocation>
</ProcessLogInfo>
```

Get Item Log

This returns the log of a visual process item.

GET method

The entry point for this method is

`api/v1/visualprocesses/{id}/logitem`

Request Information

The URI parameter is as follows:

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
Id	ID of a process log's item	Integer	Required

Body Parameters: None

Response Information

Resource Description: None

NOTE The Item Log is returned in the following format:

```
<ProcessLogItemInfo>
  <Id>{item id}</Id>
  <ExecutionState>{state of item execution}</ExecutionState>
```

```
  <ItemLayoutSettings>{settings of item}</ItemLayoutSettings>
  <Name>{item name}</Name>
  <ItemType>{type of item}</ItemType>
  <ItemInfo>{info of item, for example: the model name, input
file name, export file name as well as number of records
exported to the output, location}</ItemInfo>
</ProcessLogItemInfo>
```

REPORT WAREHOUSE API

AVAILABLE OPERATIONS

The API's functionality consists of the following major parts:

- **Access to the MS Database data** includes retrieving various database data such as saved search lists and document type information. Results are represented in a defined XML or JSON format.
- **Document search** includes performing document searches over the MS database. Both pre-defined saved searches and arbitrary criteria-based searches can be performed. The result is a list of found documents in the XML or JSON format.
- **Report data retrieval**. Report data can be taken from the MS storage in a plain text format.
- **Access to data-derived views**. Most of the data-derived views available in the MS system can be accessed this way. This includes report and table/summary data exported into various presentational formats, such as XML, JSON, HTML, Excel, PRF, PDF, and ES Style. The format of output data depends on the particular view type.

API TESTING

You can test how the API works in the following ways:

- On the interactive API help page at

  `http://localhost/MSAdmin/api/help`

  On this page, you can view detailed information about each method and test all the methods by clicking the **Test API** button.

  > **NOTE** In Monarch Server v2025.0 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as "DSClient" (or "DSAdmin"). In this case, use "DSClient" (or "DSAdmin") instead of "MSClient" (or "MSAdmin") to open the Client (or Admin) page.

- Via Swagger at

  `http://localhost/MSAdmin/swagger`

WORKING WITH THE MONARCH SERVER API

To work with the Monarch Server API successfully, review the information below.

Request authentication

For authentication, use the Login method.

API entry point

Access to the MS HTTP API is performed through an entry point.

The format of the request URL is

```
http://<host>:<port>/MSClient/api
```

Here, <host> is the name of the computer where MS is running, and <port> is the port number on which MS is listening.

The request can be sent using either an HTTP GET or an HTTP POST method (unless specified otherwise). The target area can be either the top window or a new browser window.

Parameter values that contain URL-prohibited symbols must be encoded as required by the URL specification.

Some parameters depend on the particular operation type.

For example, the list of all saved searches can be retrieved with the following request (assuming that the built-in administrator account is used):

```
http://localhost/MSClient/api/searches
```

Response type and error handling

The response to a request can be one of the following depending on the success condition:

- Request succeeded. The response has a corresponding MIME-type or an HTTP redirect.
- Request failed.

Depending on the operation nature and status, the response can be one of the following:

- The standard MS error page describing the nature of the error (for HTML-based views)
- HTTP error code with an optional message
- XML error description (non-HTML views)

In case of an XML error description, the response type is text/xml. The format of an XML error message is

```
<Error>
 <Message>[Message]</Message>
 <ExceptionMessage>[ExceptionMessage]</ExceptionMessage>
 <ExceptionType>[ExceptionType]</ExceptionType>
 <StackTrace>[StackTrace]</StackTrace>
</Error>
```

The client application can check the root node's name if text/xml content is returned to determine whether an error occurred.

Monarch Server API Usage Sample

Use the Login method for user authentication.

```
http://localhost/MSClient/api/login
```

Upon logging in, the user can either stay in the system or leave via the Logout method.

```
http://localhost/MSClient/api/logout?redirectUrl={redirectUrl}
```

The user can use different methods for data viewing; for example, the user can retrieve a table export in PDF format by using the following API method.

1. Use the Saved search list method to get the Search ID.

```
http://localhost/MSClient/api/searches/
```

2. Then, use the Saved search method with the Search ID value to get the Document IDs.

```
http://localhost/MSClient /api/search/{searchid}
```

3. Use the Model method with the Document ID value to get the Model ID used with the current document type and to get the data on the current model (e.g., filters, sorting, and summary).

```
http://localhost/MSClient/api/model/reportModel?documentId={documentId}
```

4. Finally, use the TableExport — Export to PDF method and supply the model values to get a filtered and sorted PDF export table.

```
http://localhost/MSClient/api/export/{documentIds}/table/pdf?sortName={sort
Name}&filterName={filterName}&modelName={modelName}&joinPassword={joinPassw
ord}
```

API REFERENCE

The following is a complete reference of methods supported by the MS API.

Login

Use the following methods to begin and end a user session to work with MS.

Logging in

Used to establish a “session” — a secure environment for further operation.

POST method

The entry point for this method is

```
api/login
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
Username	Username	String	Required
Password	Password	String	None
Domain	User domain	String	None

Request format sample (application/xml, text/xml)

```
<LoginRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Username>admin</Username>
```

```
  <Password>password</Password>
</LoginRequest>
```

Autologin

Used to establish a “session” when the Active directory authentication is enabled with the selected SSO option.

POST method

The entry point for this method is

`api/autologin`

Request Information

URI Parameters: none.

Body Parameters: none.

Logging out

This operation terminates the current API session. It can be useful when the maximum number of user sessions is limited.

GET method

The entry point for this method is

`api/logout?redirectUrl={redirectUrl}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
redirectUrl	URL for the redirect after session termination. If this parameter is absent, then no redirect is sent.	String	None

Body Parameters: none.

Summary Export

Summary information can be exported into a variety of formats listed below.

Export to PDF

Adobe Portable Document Format can represent report data, tables, and summaries. You can export summary data using both GET and POST methods.

GET method

The entry point for this method is

```
api/export/{documentIds}/summary/{summaryName}/pdf?filterName={filterName}&modelName={modelName}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
summaryName	The name of a summary that is used for summary generation.	String	Required
filterName	Name of the filter used for table or summary generation. The filter specified must exist in a report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

```
application/pdf
```

POST method

The entry point for this method is

```
api/export/summary/pdf
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
FilterName	Name of the filter to be used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None

SummaryName	The name of a summary that is used for summary generation.	String	Required
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
  "SummaryName": "sample string 2",
  "ModelName": "sample string 3",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<SummaryExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SummaryName>sample string 2</SummaryName>
  <ModelName>sample string 3</ModelName>
</SummaryExportRequest>
```

Response Information

The response MIME-type is

```
application/pdf
```

Export to XLS

The Excel format is suitable for displaying tables and summaries.

GET method

The entry point for this method is

```
api/export/{documentIds}/summary/{summaryName}/xls?filterName={filterName}&mo
delName={modelName}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
summaryName	The name of the summary that is used for summary generation.	String	Required
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

`application/vnd.ms-excel`

POST method

The entry point for this method is

`api/export/summary/xls`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SummaryName	The name of the summary that is used for summary generation.	String	Required
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

Name	Description	Type	Additional information
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
  "SummaryName": "sample string 2",
  "ModelName": "sample string 3",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<SummaryExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SummaryName>sample string 2</SummaryName>
  <ModelName>sample string 3</ModelName>
</SummaryExportRequest>
```

Response Information

The response MIME-type is

```
application/vnd.ms-excel
```

Export raw data (XML)

This method is used to export raw summary data (in XML format).

GET method

The entry point for this method is

```
api/export/{documentIds}/summary/{summaryName}?filterName={filterName}&modelN
ame={modelName}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the	String	Required

	documents requested are of more than one type, an error is returned.		
summaryName	The name of the summary that is used for summary generation.	String	Required
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

Body Parameters: none.

Response Information

Resource Description:

`XmlExportResponse`

Name	Description	Type	Additional information
Headers	Includes all column headers in this table.	Collection of Header	None
Rows	Includes all table data.	Collection of RowData	None

Response formats (application/xml, text/xml)

```
<XmlExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Headers>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
  </Headers>
  <Rows>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
    <RowData>
      <Values>
        <string>sample string 1</string>
```

```
        <string>sample string 2</string>
      </Values>
    </RowData>
  </Rows>
</XmlExportResponse>
```

POST method

The entry point for this method is

`api/export/summary`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SummaryName	The name of the summary that is used for summary generation.	String	Required
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
  "SummaryName": "sample string 2",
  "ModelName": "sample string 3",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<SummaryExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SummaryName>sample string 2</SummaryName>
  <ModelName>sample string 3</ModelName>
```

```
</SummaryExportRequest>
```

Response Information

Resource Description:

`XmlExportResponse`

Name	Description	Type	Additional information
Headers	Includes all column headers in this table.	Collection of Header	None
Rows	Includes all table data.	Collection of RowData	None

Response format sample (application/xml, text/xml)

```
<XmlExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Headers>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
  </Headers>
  <Rows>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
  </Rows>
</XmlExportResponse>
```

Export — HTML View

The HTML format can represent report data, tables, and summaries. You can export summary data using both GET and POST methods.

GET method

The entry point for this method is

`api/export/{documentIds}/summary/{summaryName}/html/{pageNum}?filterName={filterName}&modelName={modelName}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
summaryName	The name of the summary that is used for summary generation.	String	Required
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
pageNum	Number of pages to return. Valid range is from 1 to the number of pages.	Integer	None

Body Parameters: none.

Response Information

The response MIME-type is

`text/html`

POST method

The entry point for this method is

`api/export/summary/html`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
PageNum	Number of pages to return. Valid range is from 1 to the number of pages.	Integer	Range: inclusive between 1 and 2147483647
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None

Name	Description	Type	Additional information
SummaryName	The name of the summary that is used for summary generation.	String	Required
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "PageNum": 1,
  "FilterName": "sample string 1",
  "SummaryName": "sample string 2",
  "ModelName": "sample string 3",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<HtmlSummaryExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SummaryName>sample string 2</SummaryName>
  <ModelName>sample string 3</ModelName>
  <PageNum>1</PageNum>
</HtmlSummaryExportRequest>
```

Response Information

The response MIME-type is

```
text/html
```

Report Export

Report information can be exported into a variety of formats listed below.

Export to PDF

Adobe Portable Document Format can represent report data, tables, and summaries.

GET method

The entry point for this method is

`api/export/{documentIds}/report/pdf`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required

Body Parameters: none.

Response Information

The response MIME-type is

`application/pdf`

POST method

The entry point for this method is

`api/export/report/pdf`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "DocumentIds": [
    1,
    2
```

```
  ]
}
```

Request format sample (application/xml, text/xml)

```
<ReportExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
</ReportExportRequest>
```

Response Information

The response MIME-type is

`application/pdf`

Export raw data (XML)

This method is used to export raw report data (in the XML format).

GET method

The entry point for this method is

`api/export/{documentIds}/report`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required

Body Parameters: none.

Response Information

Resource Description:

`XmlReportExportResponse`

Name	Description	Type	Additional information
PagesResponse	Report pages	Collection of PageResponse	None

Response format sample (application/xml, text/xml)

```
<XmlReportExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PagesResponse>
    <PageResponse>
```

```
      <Lines>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Lines>
    </PageResponse>
    <PageResponse>
      <Lines>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Lines>
    </PageResponse>
  </PagesResponse>
</XmlReportExportResponse>
```

POST method

The entry point for this method is

```
api/export/report
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<ReportExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
</ReportExportRequest>
```

Response Information

Resource Description:

`XmlReportExportResponse`

Name	Description	Type	Additional information
PagesResponse	Report's pages	Collection of PageResponse	None

Response format sample (application/xml, text/xml)

```
<XmlReportExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PagesResponse>
    <PageResponse>
      <Lines>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Lines>
    </PageResponse>
    <PageResponse>
      <Lines>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Lines>
    </PageResponse>
  </PagesResponse>
</XmlReportExportResponse>
```

Export to PRF

A portable report format can present report data in conjunction with data model/navigational information.

GET method

The entry point for this method is

`api/export/{documentIds}/report/prf/modelName?modelName={modelName}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

`application/prf`

POST method

The entry point for this method is

`api/export/report/prf`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "ModelName": "sample string 1",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<PrfExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <ModelName>sample string 1</ModelName>
</PrfExportRequest>
```

Response Information

The response MIME-type is:

`application/prf`

Export — HTML View

The HTML format can represent report data, tables, and summaries. You can export report data using both GET and POST methods.

GET method

The entry point for this method is

`api/export/{documentIds}/report/html/{pageNum}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
pageNum	Number of pages to return. Valid range is from 1 to the number of pages.	Integer	None

Body Parameters: none.

Response Information

The response MIME-type is

`text/html`

POST method

The entry point for this method is

`api/export/report/html`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
PageNum	Number of pages to return. Valid range is from 1 to the number of pages.	Integer	Range: inclusive between 1 and 2147483647
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
```

```
  "PageNum": 1,
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<HtmlReportExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <PageNum>1</PageNum>
</HtmlReportExportRequest>
```

Response Information

The response MIME-type is

`text/html`

Export to ES Style

The ES Style view is based on the XML output from a table/report. It is combined with pre-designed XSL files for form-based viewing.

GET method

The entry point for this method is

`api/export/{documentIds}/report/esstyle/{templateName}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
templateName	Name of the template to be used for the export.	String	Required

Body Parameters: none.

Response Information

The response MIME-type is

`text/xml`

POST method

The entry point for this method is

```
api/export/report/esstyle
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
TemplateName	Name of the template to be used for the export.	String	Required
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "TemplateName": "sample string 1",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<EsStyleReportExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <TemplateName>sample string 1</TemplateName>
</EsStyleReportExportRequest>
```

Response Information

The response MIME-type is

```
text/xml
```

Download Original Report

This operations provides the ability to download original reports from the server by providing file IDs.

GET method

The entry point for this method is

`api/export/{fileIds}/originalreports`

Request Information

URI Parameters:

Name	Description	Type	Additional information
fileIds	Comma-separated list of IDs of reports to retrieve. If no reports are found, an error will be returned.	String	Required

Body Parameters: none.

Response Information

The response MIME-type is

`application/json`

Model

This operation returns information about the given model.

GET method

The entry point for this method is

`api/model/reportModel?documentId={documentId}&modelName={modelName}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentId	ID of a document. Information about the model associated with this document is returned.	Integer	None
modelName	Model name. If the docId parameter is absent, information about the model with this name is returned.	String	None

Body Parameters: none.

Response Information

Resource Description:

Name	Description	Type	Additional information
Id	Database ID of the model.	Integer	None
Summaries	All summaries in this model.	Collection of strings	None
Sorts	All sorts in this model.	Collection of strings	None
Filters	All filters in this model.	Collection of strings	None

Response format sample (application/json, text/json)

```
{
  "Id": 1,
  "Summaries": [
    "sample string 1",
    "sample string 2"
  ],
  "Sorts": [
    "sample string 1",
    "sample string 2"
  ],
  "Filters": [
    "sample string 1",
    "sample string 2"
  ]
}
```

Response format sample (application/xml, text/xml)

```
<ModelInfoResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Id>1</Id>
  <Summaries>
    <string>sample string 1</string>
    <string>sample string 2</string>
  </Summaries>
  <Sorts>
    <string>sample string 1</string>
    <string>sample string 2</string>
  </Sorts>
  <Filters>
    <string>sample string 1</string>
    <string>sample string 2</string>
  </Filters>
</ModelInfoResponse>
```

Report

This operation returns report information — document boundary information and annotations.

GET method

The entry point for this method is

```
api/report/{documentIds}
```

Request Information
URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required

Body Parameters: none.

Response Information

Resource Description:

Collection of `ReportDataResponse`.

Name	Description	Type	Additional information
FileId	Database ID of the file.	Integer	None
Documents	Includes all documents in this report.	Collection of ReportDocumentResponse (see below)	None
Annotations	Includes all annotations for this report.	Collection of AnnotationResponse (see below)	None

ReportDocumentResponse

Name	Description	Type	Additional information
Id	Database ID of the document	Integer	None
StartPage	Number of start pages of the document	Integer	None
StartLine	Number of start lines of the document	Integer	None
EndPage	Number of end pages of the document	Integer	None
EndLine	Number of end lines of the document	Integer	None
Title	Document name	String	None

AnnotationResponse

Name	Description	Type	Additional information
Id	Database ID of the annotation	Integer	None
Page	Number of annotated pages	Integer	None
BeginLine	Number of first lines of annotation	Integer	None
EndLine	Number of last lines of annotation	Integer	None
BeginColumn	Number of first columns of annotation	Integer	None

EndColumn	Number of last columns of annotation	Integer	None
AuthorId	Database ID of annotation author (user)	Integer	None
LastModified	Date when the user modified the annotation	Date	None
AuthorName	Database ID of annotation author (user)	String	None
Text	Annotation text	String	None

Response format sample (application/json, text/json)


```
[
 {
   "FileId": 1,
   "Documents": [
     {
       "Id": 1,
       "StartPage": 2,
       "StartLine": 3,
       "EndPage": 4,
       "EndLine": 5,
       "Title": "sample string 6"
     },
     {
       "Id": 1,
       "StartPage": 2,
       "StartLine": 3,
       "EndPage": 4,
       "EndLine": 5,
       "Title": "sample string 6"
     }
   ],
   "Annotations": [
     {
       "Id": 1,
       "Page": 2,
       "BeginLine": 3,
       "EndLine": 4,
       "BeginColumn": 5,
       "EndColumn": 6,
       "AuthorId": 7,
       "LastModified": "2014-07-09T16:59:05.4642865Z",
       "AuthorName": "sample string 9",
       "Text": "sample string 10"
     },
     {
       "Id": 1,
       "Page": 2,
       "BeginLine": 3,
       "EndLine": 4,
       "BeginColumn": 5,
       "EndColumn": 6,
       "AuthorId": 7,
       "LastModified": "2014-07-09T16:59:05.4642865Z",
       "AuthorName": "sample string 9",
       "Text": "sample string 10"
     }
   ]
```

```
  },
  {
    "FileId": 1,
    "Documents": [
      {
        "Id": 1,
        "StartPage": 2,
        "StartLine": 3,
        "EndPage": 4,
        "EndLine": 5,
        "Title": "sample string 6"
      },
      {
        "Id": 1,
        "StartPage": 2,
        "StartLine": 3,
        "EndPage": 4,
        "EndLine": 5,
        "Title": "sample string 6"
      }
    ],
    "Annotations": [
      {
        "Id": 1,
        "Page": 2,
        "BeginLine": 3,
        "EndLine": 4,
        "BeginColumn": 5,
        "EndColumn": 6,
        "AuthorId": 7,
        "LastModified": "2014-07-09T16:59:05.4642865Z",
        "AuthorName": "sample string 9",
        "Text": "sample string 10"
      },
      {
        "Id": 1,
        "Page": 2,
        "BeginLine": 3,
        "EndLine": 4,
        "BeginColumn": 5,
        "EndColumn": 6,
        "AuthorId": 7,
        "LastModified": "2014-07-09T16:59:05.4642865Z",
        "AuthorName": "sample string 9",
        "Text": "sample string 10"
      }
    ]
  }
]
```

Response format sample (application/xml, text/xml)

```
<ArrayOfReportDataResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReportDataResponse>
    <FileId>1</FileId>
```

```
    <Documents>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
    </Documents>
    <Annotations>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T19:59:05.4642865+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T19:59:05.4642865+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
    </Annotations>
  </ReportDataResponse>
  <ReportDataResponse>
    <FileId>1</FileId>
    <Documents>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
      <ReportDocumentResponse>
```

```
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
    </Documents>
    <Annotations>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T19:59:05.4642865+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T19:59:05.4642865+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
    </Annotations>
  </ReportDataResponse>
</ArrayOfReportDataResponse>
```

POST method

The entry point for this method is

```
api/report
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<ReportDataRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
</ReportDataRequest>
```

Response Information

Resource Description:

Collection of `ReportDataResponse`

Name	Description	Type	Additional information
FileId	Database ID of the file.	Integer	None
Documents	Includes all documents in this report.	Collection of ReportDocumentResponse (see below)	None
Annotations	Includes all annotations for this report.	Collection of AnnotationResponse (see below)	None

ReportDocumentResponse

Name	Description	Type	Additional information
Id	Database ID of the document	Integer	None
StartPage	Number of start pages of the document	Integer	None
StartLine	Number of start lines of the document	Integer	None
EndPage	Number of end pages of the document	Integer	None
EndLine	Number of end lines of the document	Integer	None
Title	Document name	String	None

AnnotationResponse

Name	Description	Type	Additional information
Id	Database ID of the annotation	Integer	None
Page	Number of annotated pages	Integer	None
BeginLine	Number of first lines of annotation	Integer	None
EndLine	Number of last lines of annotation	Integer	None
BeginColumn	Number of first columns of annotation	Integer	None
EndColumn	Number of last columns of annotation	Integer	None
AuthorId	Database ID of annotation author (user)	Integer	None
LastModified	Date when the user modified the annotation	Date	None
AuthorName	Database ID of annotation author (user)	String	None
Text	Annotation text	String	None

Response format sample (application/json, text/json)


```
[
 {
   "FileId": 1,
   "Documents": [
     {
       "Id": 1,
       "StartPage": 2,
       "StartLine": 3,
       "EndPage": 4,
       "EndLine": 5,
       "Title": "sample string 6"
     },
     {
       "Id": 1,
       "StartPage": 2,
       "StartLine": 3,
       "EndPage": 4,
       "EndLine": 5,
       "Title": "sample string 6"
     }
   ],
   "Annotations": [
     {
       "Id": 1,
       "Page": 2,
       "BeginLine": 3,
       "EndLine": 4,
       "BeginColumn": 5,
       "EndColumn": 6,
       "AuthorId": 7,
       "LastModified": "2014-07-09T17:03:43.1248092Z",
       "AuthorName": "sample string 9",
       "Text": "sample string 10"
```

```
    },
    {
      "Id": 1,
      "Page": 2,
      "BeginLine": 3,
      "EndLine": 4,
      "BeginColumn": 5,
      "EndColumn": 6,
      "AuthorId": 7,
      "LastModified": "2014-07-09T17:03:43.1248092Z",
      "AuthorName": "sample string 9",
      "Text": "sample string 10"
    }
  ]
},
{
  "FileId": 1,
  "Documents": [
    {
      "Id": 1,
      "StartPage": 2,
      "StartLine": 3,
      "EndPage": 4,
      "EndLine": 5,
      "Title": "sample string 6"
    },
    {
      "Id": 1,
      "StartPage": 2,
      "StartLine": 3,
      "EndPage": 4,
      "EndLine": 5,
      "Title": "sample string 6"
    }
  ],
  "Annotations": [
    {
      "Id": 1,
      "Page": 2,
      "BeginLine": 3,
      "EndLine": 4,
      "BeginColumn": 5,
      "EndColumn": 6,
      "AuthorId": 7,
      "LastModified": "2014-07-09T17:03:43.1248092Z",
      "AuthorName": "sample string 9",
      "Text": "sample string 10"
    },
    {
      "Id": 1,
      "Page": 2,
      "BeginLine": 3,
      "EndLine": 4,
      "BeginColumn": 5,
      "EndColumn": 6,
      "AuthorId": 7,
      "LastModified": "2014-07-09T17:03:43.1248092Z",
```

```
        "AuthorName": "sample string 9",
        "Text": "sample string 10"
      }
    ]
  }
]
```

Response format sample (application/xml, text/xml)

```
<ArrayOfReportDataResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReportDataResponse>
    <FileId>1</FileId>
    <Documents>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
    </Documents>
    <Annotations>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T20:03:43.1248092+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T20:03:43.1248092+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
```

```
    </Annotations>
  </ReportDataResponse>
  <ReportDataResponse>
    <FileId>1</FileId>
    <Documents>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
      <ReportDocumentResponse>
        <Id>1</Id>
        <StartPage>2</StartPage>
        <StartLine>3</StartLine>
        <EndPage>4</EndPage>
        <EndLine>5</EndLine>
        <Title>sample string 6</Title>
      </ReportDocumentResponse>
    </Documents>
    <Annotations>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T20:03:43.1248092+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
      <AnnotationResponse>
        <Id>1</Id>
        <Page>2</Page>
        <BeginLine>3</BeginLine>
        <EndLine>4</EndLine>
        <BeginColumn>5</BeginColumn>
        <EndColumn>6</EndColumn>
        <AuthorId>7</AuthorId>
        <LastModified>2014-07-09T20:03:43.1248092+03:00</LastModified>
        <AuthorName>sample string 9</AuthorName>
        <Text>sample string 10</Text>
      </AnnotationResponse>
    </Annotations>
  </ReportDataResponse>
</ArrayOfReportDataResponse>
```

Search

MS API supports several search options.

Search by criteria

This operation executes a search with explicitly defined criteria. The search returns a list of found documents.

POST method

The entry point for this method is

`api/search`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
DocumentTypes	Contains the list of IDs of all document types for a search.	Collection of integers	Required min length: 1
DateFrom	Finds documents after this date Date format: m/d/yyyy You can use next patterns: <TODAY>, <YESTERDAY>, <[N] DAYS AGO>, <LAST WEEKDAY>, <START OF LAST MONTH>, <END OF LAST MONTH>, <START OF THIS MONTH>, <END OF THIS MONTH>, <START OF LAST WEEK>, <END OF LAST WEEK>, <START OF THIS WEEK>, <END OF THIS WEEK>, <START OF THIS QUARTER>, <END OF THIS QUARTER>, <START OF LAST QUARTER>, <END OF LAST QUARTER>.	String	None
DateTo	Finds all documents before this date Date format: m/d/yyyy You can use next patterns: <TODAY>, <YESTERDAY>, <[N] DAYS AGO>, <LAST WEEKDAY>, <START OF LAST MONTH>, <END OF LAST MONTH>, <START OF THIS MONTH>, <END OF THIS MONTH>, <START OF LAST WEEK>, <END OF LAST WEEK>, <START OF THIS WEEK>, <END OF THIS WEEK>, <START OF THIS QUARTER>, <END OF THIS QUARTER>, <START OF LAST QUARTER>, <END OF LAST QUARTER>.	String	None
SearchCriteria	Finds all documents with specified keyword types and keyword values.	Collection of SearchCriterionRequest (see below)	None

SearchCriterionRequest

Name	Description	Type	Additional information

IndexFieldId	Database ID of an index that must match these criteria	Integer	Required range: inclusive between 100 and 2147483647
Value	The correct value in a search expression	String	Required
CompareType	One of the relational operators: >, <, >=, <=, <>, =	String	Required matching of regular expression patterns: (>|<|>=|<=|<>|=)
JoinType	One of the following operations: AND, OR. It sets the relation of the criteria with the previous one.	String	Required matching of regular expression patterns: (AND|OR)

Request format sample (application/json, text/json)

```
{
  "DocumentTypes": [
    1,
    2
  ],
  "DateFrom": "sample string 1",
  "DateTo": "sample string 2",
  "SearchCriteria": [
    {
      "IndexFieldId": 1,
      "Value": "sample string 2",
      "CompareType": "sample string 3",
      "JoinType": "sample string 4"
    },
    {
      "IndexFieldId": 1,
      "Value": "sample string 2",
      "CompareType": "sample string 3",
      "JoinType": "sample string 4"
    }
  ]
}
```

Request format sample (application/xml, text/xml)

```
<DocumentSearchRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentTypes>
    <int>1</int>
    <int>2</int>
  </DocumentTypes>
  <DateFrom>sample string 1</DateFrom>
  <DateTo>sample string 2</DateTo>
  <SearchCriteria>
    <SearchCriterionRequest>
      <IndexFieldId>1</IndexFieldId>
      <Value>sample string 2</Value>
      <CompareType>sample string 3</CompareType>
      <JoinType>sample string 4</JoinType>
    </SearchCriterionRequest>
    <SearchCriterionRequest>
```

```
      <IndexFieldId>1</IndexFieldId>
      <Value>sample string 2</Value>
      <CompareType>sample string 3</CompareType>
      <JoinType>sample string 4</JoinType>
    </SearchCriterionRequest>
  </SearchCriteria>
</DocumentSearchRequest>
```

Response Information

Resource Description:

Collection of `SearchDocumentTypeGroupResponse`

Name	Description	Type	Additional information
Id	Database ID.	Integer	None
Name	Document group name	String	None
Description	Document group description	String	None
DocumentTypeResponses		Collection of SearchDocumentTypeResponse (see below)	None

SearchDocumentResponse

Name	Description	Type	Additional information
Name	Document name	String	None
Id	Database ID of the document	Integer	None
FileId	Database file ID	Integer	None
DocumentDate	Document date	Date	None
IsBinary	True if document is binary, False if not binary	Boolean	None
AnnotationCount	Quantity of annotations for the current document	Integer	None

Response format sample (application/json, text/json)

```
[
 {
   "Id": 1,
   "Name": "sample string 2",
   "Description": "sample string 3",
   "DocumentTypeResponses": [
     {
       "Id": 1,
       "Name": "sample string 2",
       "Description": "sample string 3",
       "Documents": [
         {
           "Name": "sample string 1",
           "Id": 2,
```

```
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      },
      {
        "Id": 1,
        "Name": "sample string 2",
        "Description": "sample string 3",
        "Documents": [
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      }
    ]
  },
  {
    "Id": 1,
    "Name": "sample string 2",
    "Description": "sample string 3",
    "DocumentTypeResponses": [
      {
        "Id": 1,
        "Name": "sample string 2",
        "Description": "sample string 3",
        "Documents": [
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
```

```
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      },
      {
        "Id": 1,
        "Name": "sample string 2",
        "Description": "sample string 3",
        "Documents": [
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:09:41.1644355Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      }
    ]
  }
]
```


Response format sample (application/xml, text/xml)


```
<ArrayOfSearchDocumentTypeGroupResponse
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SearchDocumentTypeGroupResponse>
    <Id>1</Id>
    <Name>sample string 2</Name>
    <Description>sample string 3</Description>
    <DocumentTypeResponses>
      <SearchDocumentTypeResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Description>sample string 3</Description>
        <Documents>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
```

```
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
      </Documents>
    </SearchDocumentTypeResponse>
    <SearchDocumentTypeResponse>
      <Id>1</Id>
      <Name>sample string 2</Name>
      <Description>sample string 3</Description>
      <Documents>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
      </Documents>
    </SearchDocumentTypeResponse>
  </DocumentTypeResponses>
</SearchDocumentTypeGroupResponse>
<SearchDocumentTypeGroupResponse>
  <Id>1</Id>
  <Name>sample string 2</Name>
  <Description>sample string 3</Description>
  <DocumentTypeResponses>
    <SearchDocumentTypeResponse>
      <Id>1</Id>
      <Name>sample string 2</Name>
      <Description>sample string 3</Description>
      <Documents>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
```

```
          </SearchDocumentResponse>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
        </Documents>
      </SearchDocumentTypeResponse>
      <SearchDocumentTypeResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Description>sample string 3</Description>
        <Documents>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:09:41.1644355+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
        </Documents>
      </SearchDocumentTypeResponse>
    </DocumentTypeResponses>
  </SearchDocumentTypeGroupResponse>
</ArrayOfSearchDocumentTypeGroupResponse>
```

Saved search

This operation executes a saved search and returns a list of documents.

GET method

The entry point for this method is

```
api/search/{searchid}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
searchid	ID of the saved search. Search ID can be obtained using the Search List operation.	Integer	Required

Body Parameters: none.

Response Information

Resource Description:

Collection of `SearchDocumentTypeGroupResponse`

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Document group name	String	None
Description	Document group description	String	None
DocumentTypeResponses		Collection of SearchDocumentTypeResponse (see below)	None

SearchDocumentTypeResponse

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Document type name	String	None
Description	Document type description	String	None
Documents		Collection of SearchDocumentResponse (see below)	None

SearchDocumentResponse

Name	Description	Type	Additional information
Name	Document name	String	None
Id	Database ID of the document	Integer	None
FileId	Database file ID	Integer	None
DocumentDate	Document date	Date	None
IsBinary	True if document is binary, False if not binary	Boolean	None
AnnotationCount	Quantity of annotations for the current document	Integer	None

Response format sample (application/json, text/json)

```
[
 {
   "Id": 1,
   "Name": "sample string 2",
```

```
  "Description": "sample string 3",
  "DocumentTypeResponses": [
    {
      "Id": 1,
      "Name": "sample string 2",
      "Description": "sample string 3",
      "Documents": [
        {
          "Name": "sample string 1",
          "Id": 2,
          "FileId": 3,
          "DocumentDate": "2014-07-09T17:26:12.6072406Z",
          "IsBinary": true,
          "AnnotationCount": 6
        },
        {
          "Name": "sample string 1",
          "Id": 2,
          "FileId": 3,
          "DocumentDate": "2014-07-09T17:26:12.6072406Z",
          "IsBinary": true,
          "AnnotationCount": 6
        }
      ]
    },
    {
      "Id": 1,
      "Name": "sample string 2",
      "Description": "sample string 3",
      "Documents": [
        {
          "Name": "sample string 1",
          "Id": 2,
          "FileId": 3,
          "DocumentDate": "2014-07-09T17:26:12.6072406Z",
          "IsBinary": true,
          "AnnotationCount": 6
        },
        {
          "Name": "sample string 1",
          "Id": 2,
          "FileId": 3,
          "DocumentDate": "2014-07-09T17:26:12.6072406Z",
          "IsBinary": true,
          "AnnotationCount": 6
        }
      ]
    }
  ]
},
{
  "Id": 1,
  "Name": "sample string 2",
  "Description": "sample string 3",
  "DocumentTypeResponses": [
    {
      "Id": 1,
```

```
        "Name": "sample string 2",
        "Description": "sample string 3",
        "Documents": [
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:26:12.6072406Z",
            "IsBinary": true,
            "AnnotationCount": 6
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:26:12.6072406Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      },
      {
        "Id": 1,
        "Name": "sample string 2",
        "Description": "sample string 3",
        "Documents": [
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:26:12.6072406Z",
            "IsBinary": true,
            "AnnotationCount": 6
          },
          {
            "Name": "sample string 1",
            "Id": 2,
            "FileId": 3,
            "DocumentDate": "2014-07-09T17:26:12.6072406Z",
            "IsBinary": true,
            "AnnotationCount": 6
          }
        ]
      }
    ]
  }
]
```


Response format sample (application/xml, text/xml)


```
<ArrayOfSearchDocumentTypeGroupResponse
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SearchDocumentTypeGroupResponse>
    <Id>1</Id>
    <Name>sample string 2</Name>
```

```
  <Description>sample string 3</Description>
  <DocumentTypeResponses>
    <SearchDocumentTypeResponse>
      <Id>1</Id>
      <Name>sample string 2</Name>
      <Description>sample string 3</Description>
      <Documents>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
      </Documents>
    </SearchDocumentTypeResponse>
    <SearchDocumentTypeResponse>
      <Id>1</Id>
      <Name>sample string 2</Name>
      <Description>sample string 3</Description>
      <Documents>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
        <SearchDocumentResponse>
          <Name>sample string 1</Name>
          <Id>2</Id>
          <FileId>3</FileId>
          <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
          <IsBinary>true</IsBinary>
          <AnnotationCount>6</AnnotationCount>
        </SearchDocumentResponse>
      </Documents>
    </SearchDocumentTypeResponse>
  </DocumentTypeResponses>
</SearchDocumentTypeGroupResponse>
<SearchDocumentTypeGroupResponse>
  <Id>1</Id>
  <Name>sample string 2</Name>
  <Description>sample string 3</Description>
  <DocumentTypeResponses>
    <SearchDocumentTypeResponse>
      <Id>1</Id>
```

```
        <Name>sample string 2</Name>
        <Description>sample string 3</Description>
        <Documents>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
        </Documents>
      </SearchDocumentTypeResponse>
      <SearchDocumentTypeResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Description>sample string 3</Description>
        <Documents>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
          <SearchDocumentResponse>
            <Name>sample string 1</Name>
            <Id>2</Id>
            <FileId>3</FileId>
            <DocumentDate>2014-07-09T20:26:12.6072406+03:00</DocumentDate>
            <IsBinary>true</IsBinary>
            <AnnotationCount>6</AnnotationCount>
          </SearchDocumentResponse>
        </Documents>
      </SearchDocumentTypeResponse>
    </DocumentTypeResponses>
  </SearchDocumentTypeGroupResponse>
</ArrayOfSearchDocumentTypeGroupResponse>
```

Saved search list

This operation returns the list of available saved searches.

GET method

The entry point for this method is

`api/searches/{searchGroupId}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
searchGroupId	Id of the requested search group. If this parameter is absent, searches from all search groups are returned. Otherwise, only searches of the specified search group are returned.	Integer	None

Body Parameters: none.

Response Information

Resource Description:

Collection of `SearchGroupResponse`

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Search group name	String	None
SearchResponses	Contains available searches	Collection of SearchResponse	None

SearchResponse

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Search name	String	None
Rights	Rights on this search	SearchGroupRights (see below)	None
DocumentTypes	Document types assigned for this search	Collection of DocumentTypeResponse (see below)	None
IsReport	True if it is a report search	Boolean	None

SearchGroupRights

Possible enumeration values:

Name	Value	Description
Personal	0	The current user is the owner of this search, and only the owner has rights to this search.
Group	1	The current user is the owner of this search, and other users have rights to this search.
Restrict	2	The current user is not the owner of this search but has rights to use it.

DocumentTypeResponse

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Document type name	String	None
Description	Document type description	String	None

Response format sample (application/json, text/json)

```
[
 {
   "Id": 1,
   "Name": "sample string 2",
   "SearchResponses": [
     {
       "Id": 1,
       "Name": "sample string 2",
       "Rights": 0,
       "DocumentTypes": [
         {
           "Id": 1,
           "Name": "sample string 2",
           "Description": "sample string 3"
         },
         {
           "Id": 1,
           "Name": "sample string 2",
           "Description": "sample string 3"
         }
       ],
       "IsReport": true
     },
     {
       "Id": 1,
       "Name": "sample string 2",
       "Rights": 0,
       "DocumentTypes": [
         {
           "Id": 1,
           "Name": "sample string 2",
           "Description": "sample string 3"
         },
         {
           "Id": 1,
           "Name": "sample string 2",
           "Description": "sample string 3"
         }
       ],
       "IsReport": true
     }
   ]
 },
 {
   "Id": 1,
   "Name": "sample string 2",
   "SearchResponses": [
```

```
      {
        "Id": 1,
        "Name": "sample string 2",
        "Rights": 0,
        "DocumentTypes": [
          {
            "Id": 1,
            "Name": "sample string 2",
            "Description": "sample string 3"
          },
          {
            "Id": 1,
            "Name": "sample string 2",
            "Description": "sample string 3"
          }
        ],
        "IsReport": true
      },
      {
        "Id": 1,
        "Name": "sample string 2",
        "Rights": 0,
        "DocumentTypes": [
          {
            "Id": 1,
            "Name": "sample string 2",
            "Description": "sample string 3"
          },
          {
            "Id": 1,
            "Name": "sample string 2",
            "Description": "sample string 3"
          }
        ],
        "IsReport": true
      }
    ]
  }
]
```

Response format sample (application/xml, text/xml)

```
<ArrayOfSearchGroupResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SearchGroupResponse>
    <Id>1</Id>
    <Name>sample string 2</Name>
    <SearchResponses>
      <SearchResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Rights>Personal</Rights>
        <DocumentTypes>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
```

```
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
        </DocumentTypes>
        <IsReport>true</IsReport>
      </SearchResponse>
      <SearchResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Rights>Personal</Rights>
        <DocumentTypes>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
        </DocumentTypes>
        <IsReport>true</IsReport>
      </SearchResponse>
    </SearchResponses>
  </SearchGroupResponse>
  <SearchGroupResponse>
    <Id>1</Id>
    <Name>sample string 2</Name>
    <SearchResponses>
      <SearchResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Rights>Personal</Rights>
        <DocumentTypes>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
        </DocumentTypes>
        <IsReport>true</IsReport>
      </SearchResponse>
      <SearchResponse>
        <Id>1</Id>
        <Name>sample string 2</Name>
        <Rights>Personal</Rights>
        <DocumentTypes>
```

```
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
          <DocumentTypeResponse>
            <Id>1</Id>
            <Name>sample string 2</Name>
            <Description>sample string 3</Description>
          </DocumentTypeResponse>
        </DocumentTypes>
        <IsReport>true</IsReport>
      </SearchResponse>
    </SearchResponses>
  </SearchGroupResponse>
</ArrayOfSearchGroupResponse>
```

Table Export

Table information can be exported into a variety of formats listed below.

Export to PDF

Adobe Portable Document Format can represent report data, tables, and summaries.

GET method

The entry point for this method is

```
api/export/{documentIds}/table/pdf?sortName={sortName}&filterName={filterName
}&modelName={modelName}&joinPassword={joinPassword}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
sortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

joinPassword	The password for joining with an external database (the table data exported from a report can contain additional columns from an external database).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

`application/pdf`

POST method

The entry point for this method is

`api/export/table/pdf`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type, otherwise an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported from a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of Integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
  "SortName": "sample string 2",
  "JoinPassword": "sample string 3",
  "ModelName": "sample string 4",
```

```
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<TableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SortName>sample string 2</SortName>
  <JoinPassword>sample string 3</JoinPassword>
  <ModelName>sample string 4</ModelName>
</TableExportRequest>
```

Response Information

The entry point for this method is

```
api/export/table/pdf
```

Export to XLS

The Excel format is suitable to display tables and summaries.

GET method

The entry point for this method is

```
api/export/{documentIds}/table/xls?sortName={sortName}&filterName={filterName
}&modelName={modelName}&joinPassword={joinPassword}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
sortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report	String	None

	model set for this document type; otherwise, an error is returned.		
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
joinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

`application/vnd.ms-excel`

POST method

The entry point for this method is

`api/export/table/xls`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
```

```
  "SortName": "sample string 2",
  "JoinPassword": "sample string 3",
  "ModelName": "sample string 4",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<TableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SortName>sample string 2</SortName>
  <JoinPassword>sample string 3</JoinPassword>
  <ModelName>sample string 4</ModelName>
</TableExportRequest>
```

Response Information

The response MIME-type is

```
application/vnd.ms-excel
```

Export raw data (XML)

This method is used to export raw table data (in XML format).

GET method

The entry point for this method is

```
api/export/{documentIds}/table?sortName={sortName}&filterName={filterName}&mo
delName={modelName}&joinPassword={joinPassword}
```

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
sortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None

modelName	Name of the model to be use for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
joinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None

Body Parameters: none.

Response Information

Resource Description:

`XmlExportResponse`

Name	Description	Type	Additional information
Headers	Includes all column headers in this table.	Collection of Header	None
Rows	Includes all table data.	Collection of RowData	None

Response format sample (application/json, text/json)

```
{
  "Headers": [
    {
      "Name": "sample string 1",
      "FieldType": 0
    },
    {
      "Name": "sample string 1",
      "FieldType": 0
    }
  ],
  "Rows": [
    {
      "Values": [
        "sample string 1",
        "sample string 2"
      ]
    },
    {
      "Values": [
        "sample string 1",
        "sample string 2"
      ]
    }
  ]
}
```

Response format sample (application/xml, text/xml)

```
<XmlExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Headers>
```

```
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
  </Headers>
  <Rows>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
  </Rows>
</XmlExportResponse>
```

POST method

The entry point for this method is:

```
api/export/table
```

Request Information

URI Parameters: none.

Body Parameters are the following:

Name	Description	Type	Additional information
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "FilterName": "sample string 1",
  "SortName": "sample string 2",
  "JoinPassword": "sample string 3",
  "ModelName": "sample string 4",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<TableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SortName>sample string 2</SortName>
  <JoinPassword>sample string 3</JoinPassword>
  <ModelName>sample string 4</ModelName>
</TableExportRequest>
```

Response Information

Resource Description:

`XmlExportResponse`

Name	Description	Type	Additional information
Headers	Includes all column headers in this table.	Collection of Header	None
Rows	Includes all table data.	Collection of RowData	None

Response format sample (application/xml, text/xml)

```
<XmlExportResponse xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Headers>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
    <Header>
      <Name>sample string 1</Name>
      <FieldType>String</FieldType>
    </Header>
  </Headers>
  <Rows>
    <RowData>
      <Values>
        <string>sample string 1</string>
```

```
        <string>sample string 2</string>
      </Values>
    </RowData>
    <RowData>
      <Values>
        <string>sample string 1</string>
        <string>sample string 2</string>
      </Values>
    </RowData>
  </Rows>
</XmlExportResponse>
```

Export — HTML View

The HTML format can represent report data, tables, and summaries. You can export table data using both GET and POST methods.

GET method

The entry point for this method is

`api/export/table/esstyle`

Request Information

URI Parameters: none

Body Parameters:

Name	Description	Type	Additional information
TemplateName	Name of the template to be used for the export.	String	Required
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None

DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "TemplateName": "sample string 1",
  "FilterName": "sample string 2",
  "SortName": "sample string 3",
  "JoinPassword": "sample string 4",
  "ModelName": "sample string 5",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<EsStyleTableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 2</FilterName>
  <SortName>sample string 3</SortName>
  <JoinPassword>sample string 4</JoinPassword>
  <ModelName>sample string 5</ModelName>
  <TemplateName>sample string 1</TemplateName>
</EsStyleTableExportRequest>
```

Response Information

The response MIME-type is

`text/html`

POST method

The entry point for this method is

`api/export/table/html`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
PageNum	Number of pages to return. Valid range is from 1 to the number of pages.	Integer	Range: inclusive between 1 and 2147483647
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "PageNum": 1,
  "FilterName": "sample string 1",
  "SortName": "sample string 2",
  "JoinPassword": "sample string 3",
  "ModelName": "sample string 4",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<HtmlTableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 1</FilterName>
  <SortName>sample string 2</SortName>
  <JoinPassword>sample string 3</JoinPassword>
```

```
  <ModelName>sample string 4</ModelName>
  <PageNum>1</PageNum>
</HtmlTableExportRequest>
```

Response Information

The response MIME-type is

`text/html`

Export to ES Style

The ES Style view is based on the XML output from a table/report and is combined with pre-designed XSL files for form-based viewing.

GET method

The entry point for this method is

`api/export/{documentIds}/table/esstyle/{templateName}?filterName={filterName}&sortName={sortName}&modelName={modelName}&joinPassword={joinPassword}`

Request Information

URI Parameters:

Name	Description	Type	Additional information
documentIds	Comma-separated list of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	String	Required
templateName	Name of the template to be used for the export.	String	Required
filterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type; otherwise, an error is returned.	String	None
sortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type; otherwise, an error is returned.	String	None
modelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
joinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None

Body Parameters: none.

Response Information

The response MIME-type is

```
text/xml
```

POST method

The entry point for this method is

```
api/export/table/esstyle
```

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
TemplateName	Name of the template to be used for the export.	String	Required
FilterName	Name of the filter used for table or summary generation. The specified filter must exist in the report model set for this document type, otherwise an error is returned.	String	None
SortName	Name of the sorting to apply. The specified sorting must exist in the report model set for this document type, otherwise an error is returned.	String	None
JoinPassword	The password for joining with an external database (the table data exported form a report can contain additional columns from an external database).	String	None
ModelName	Name of the model to be used for export. If this parameter is absent, then the appropriate model is detected out of the document IDs passed (however, this method is slow).	String	None
DocumentIds	List of IDs of the documents to retrieve. All of these documents must be of the same type. If the documents requested are of more than one type, an error is returned.	Collection of integers	Required min length: 1

Request format sample (application/json, text/json)

```
{
  "TemplateName": "sample string 1",
  "FilterName": "sample string 2",
  "SortName": "sample string 3",
  "JoinPassword": "sample string 4",
  "ModelName": "sample string 5",
  "DocumentIds": [
    1,
    2
  ]
}
```

Request format sample (application/xml, text/xml)

```
<EsStyleTableExportRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DocumentIds>
```

```
    <int>1</int>
    <int>2</int>
  </DocumentIds>
  <FilterName>sample string 2</FilterName>
  <SortName>sample string 3</SortName>
  <JoinPassword>sample string 4</JoinPassword>
  <ModelName>sample string 5</ModelName>
  <TemplateName>sample string 1</TemplateName>
</EsStyleTableExportRequest>
```

Response Information

The response MIME-type is:

`text/xml`

Dynamic Login

Use the following method to login to MS Client with Dynamic User.

POST method

The entry point for this method is

`api/dynamicLogin`

Request Information

URI Parameters: none.

Body Parameters:

Name	Description	Type	Additional information
UserName	Name of Dynamic User.	String	Required max length: 60
UserFullName	Full Name of Dynamic User.	String	Optional max length: 60

GroupName	The names of the Groups of which the User will be a member. For multiple Groups, use a comma as a delimiter (,).	Collection of string	Required
RedirectPage	Page that redirects the User after login. Main Page, Document Search, or Report Search.	RedirectType (see below)	Required
DocumenTypeId	Document Type ID, which is set on the Search Form by default.	Integer	Optional
DateFrom	The lower boundary limit by date that is set on the Search Form by default.	Date	Optional data type: DateTime
DateTo	The upper boundary limit by date that is set on the Search Form by default.	Date	Optional Data type: DateTime
IsLatest	Default value of Is Latest option on the Search form.	Boolean	Optional
SecurityKeywords	Security keywords	Collection of SecurityKeywordItem(see below)	Optional

RedirectPage

Name	Value	Description	
Main	0	Main Page	
DocumentSearch	1	Document Search Page	
ReportSearch	2	Report Search Page	

SecurityKeywordItem

Name	Description	Type	Additional information
Name	Security keyword name	String	None
Value	Security keyword value	String	None

Request format sample (application/json, text/json)

```
{
  "UserName": "sample string 1",
  "UserFullName": "sample string 2",
  "GroupNames": [
    "sample string 1",
    "sample string 2"
  ],
  "RedirectPage": 0,
  "DocumenTypeId": 1,
  "DateFrom": "2015-10-09T12:26:32.1385658Z",
  "DateTo": "2015-10-09T12:26:32.1385658Z",
  "IsLatest": true,
  "SecurityKeywords": [
    {
      "Name": "sample string 1",
      "Value": "sample string 2"
    },
    {
      "Name": "sample string 1",
      "Value": "sample string 2"
    }
  ]
}
```

Response Information

The response is: Redirect to Monarch Server Client.

REPORT WAREHOUSE DYNAMIC LOGIN API

API ENTRY POINT

Access to the MS HTTP API is performed through an entry point.

The format of the request URL is

```
http://<host>:<port>/MSClient/api
```

Here, <host> is the name of the computer where MS is running, and <port> is the port number on which MS is listening.

The request can be sent using either an HTTP GET or an HTTP POST method (unless specified otherwise). The target area can be either the top window or a new browser window.

Parameter values that contain URL-prohibited symbols must be encoded as required by the URL specification.

> **NOTE** In Monarch Server v2025.0 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as “DSClient” (or "DSAdmin"). In this case, use “DSClient” (or "DSAdmin”) instead of “MSClient” (or “MSAdmin”) to open the Client (or Admin) page.

RESPONSE TYPE AND ERROR HANDLING

The response to a request can be one of the following depending on the success condition:

- Request succeeded. The response has the corresponding HTTP redirect.
- Request failed.

Depending on the operation nature and status, the response can be one of the following:

- The standard MS error page describing the nature of the error (for HTML-based views)
- HTTP error code with an optional message
- XML error description (non-HTML views)

In case of an XML error description, the response type is text/xml. The format of an XML error message is

```
<Error>
 <Message>[Message]</Message>
 <ExceptionMessage>[ExceptionMessage]</ExceptionMessage>
 <ExceptionType>[ExceptionType]</ExceptionType>
 <StackTrace>[StackTrace]</StackTrace>
</Error>
```

The client application can check the root node’s name if text/xml content is returned to determine whether an error occurred.

DYNAMIC LOGIN METHOD

Use the following method to login to MS Client with Dynamic User.

POST method

The POST method entry point is

`api/dynamicLogin`

Request Information

URI Parameters: none.

Body parameters:

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
UserName	Name of Dynamic User.	String	Required max length: 60
UserFullName	Full Name of Dynamic User.	String	Optional max length: 60
GroupNames	The names of the Groups of which the User will be a member. Refer to Specifying User Group for Dynamic Login section to get information about specifying user groups. For multiple groups, use a comma as a delimiter (,).	Collection of string	Required
RedirectPage	Page that redirects the User after login. Main Page, Document Search, or Report Search.	RedirectType (see below)	Required
DocumenTypeId	Document Type ID, which is set on the Search Form by default.	Integer	Optional
DateFrom	The lower boundary limit by date, which is set on the Search Form by default.	Date	Optional data type: DateTime
DateTo	The upper boundary limit by date, which is set on the Search Form by default.	Date	Optional data type: DateTime
IsLatest	Default value of Is Latest option on the Search form.	Boolean	Optional
SecurityKeywords	Security keywords	Collection of SecurityKeywordItem (see below)	Optional

RedirectType

NAME	VALUE	DESCRIPTION
Main	0	Main Page
DocumentSearch	1	Document Search Page
ReportSearch	2	Report Search Page

SecurityKeywordItem

NAME	DESCRIPTION	TYPE	ADDITIONAL INFORMATION
Name	Security Keyword Name	string	None
Value	Security Keyword Value	string	None

Request format sample (application/json, text/json)

```
{
  "UserName": "sample string 1",
  "UserFullName": "sample string 2",
  "GroupNames": [
    "sample string 1",
    "sample string 2"
  ],
  "RedirectPage": 0,
  "DocumenTypeId": 1,
  "DateFrom": "2015-10-09T12:26:32.1385658Z",
  "DateTo": "2015-10-09T12:26:32.1385658Z",
  "IsLatest": true,
  "SecurityKeywords": [
    {
      "Name": "sample string 1",
      "Value": "sample string 2"
    },
    {
      "Name": "sample string 1",
      "Value": "sample string 2"
    }
  ]
}
```

Response Information

The response is: Redirect to Monarch Server Client.

SPECIFYING USER GROUPS FOR DYNAMIC LOGIN

When logging in to MS Client via Dynamic Login API, the user should enter the name of the user group (or multiple user groups) in the MS that he/she is entitled to log in to. Once the user is logged in, all rights of this user group are assigned to this user.

Server Administrator should create a user group in the MS Admin application and add the newly created group to the list of user groups that the dynamic users are entitled to log in to.

To specify a user group for dynamic login

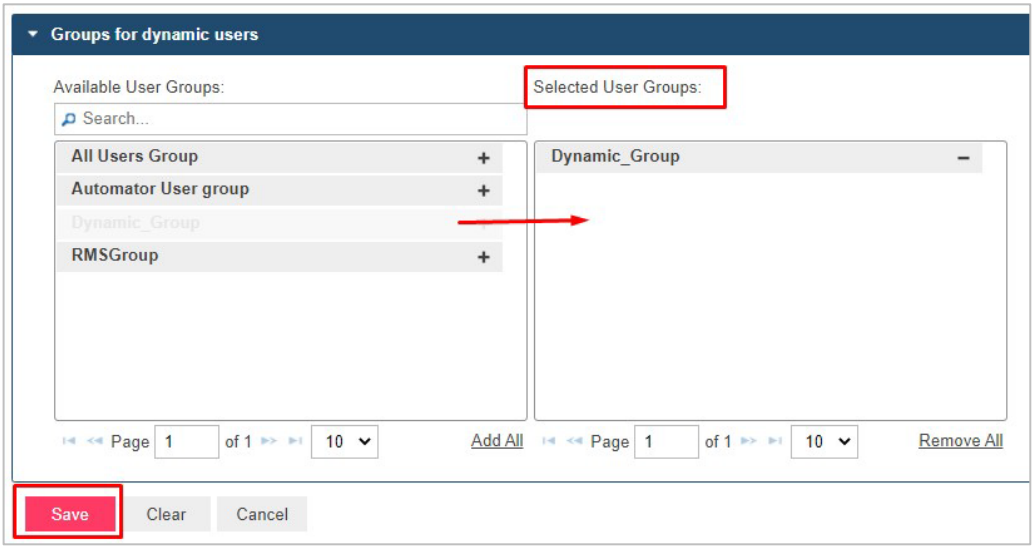
1. In the **Web.config** file of Admin application, change the value of the **ShowDynamicUserSettings** parameter from “false” to “true” to enable the Dynamic Login Setting page.

   The path to the **Web.config** file is C:\inetpub\wwwroot\MSAdmin\Web.config.

   The **ShowDynamicUserSettings** parameter is located in the <appSettings> section:

   ```
   <appSettings>
      <add key="ShowDynamicUserSettings" value="true"/>
       …
   </appSettings>
   ```

2. Login to the Admin application, create a user group (e.g., Dynamic_Group), add some rights, and assign any document types to them.
3. Open **System Settings > System Preferences > Dynamic Login Settings**, expand the **Groups** for dynamic users section, and assign the newly created user group to the **Selected User Groups** list.



4. Click the **Save** button.

REPORT WAREHOUSE DYNAMIC LOGIN INTERFACE SETTINGS

INTRODUCTION

When logging in to MS Client via Dynamic Login Test Page, the user should enter the name of the user group in the MS that is entitled to log in. Once the user is logged in, all rights of this user group are assigned to this user.

> **NOTE** In Monarch Server v2025.0 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as “DSClient” (or "DSAdmin"). In this case, use “DSClient” (or "DSAdmin”) instead of “MSClient” (or “MSAdmin”) to open the Client (or Admin) page.

Server Administrator should create a user group in the MS Admin application and add the newly created group to the list of user groups that are entitled to log in.

To specify user group for dynamic login

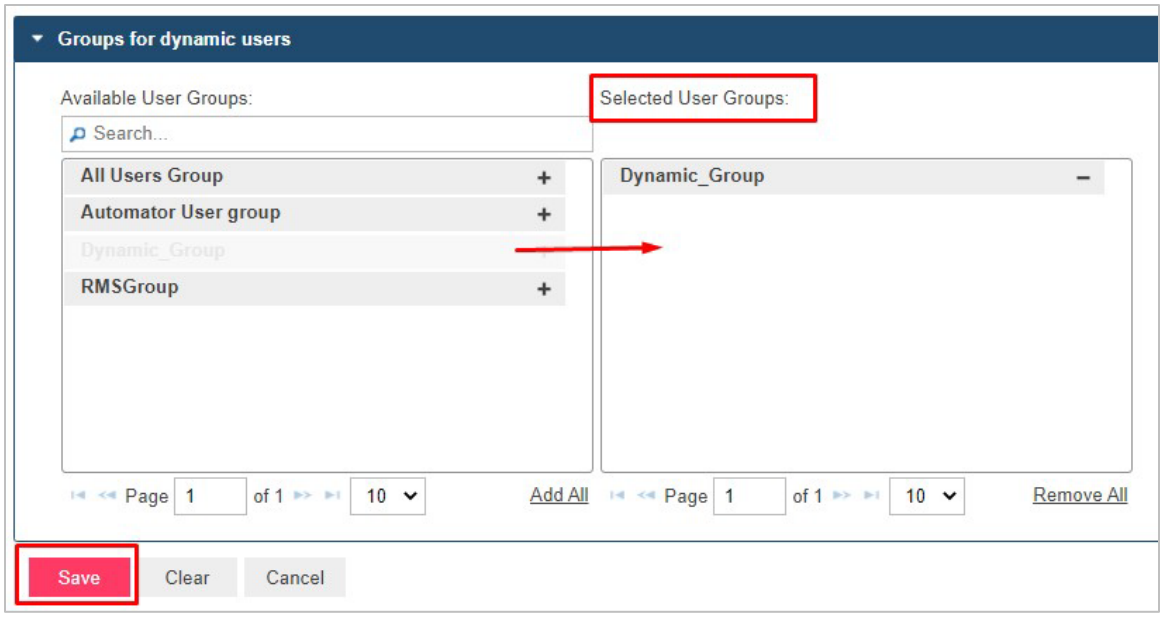
1. In the **Web.config** file of Admin application, change the value of the **ShowDynamicUserSettings** parameter from “false” to “true” to enable the Dynamic Login Setting page.

   The path to the **Web.config** file is C:\inetpub\wwwroot\MSAdmin\Web.config.

   The **ShowDynamicUserSettings** parameter is located in the <appSettings> section:

   ```
   <appSettings>
      <add key="ShowDynamicUserSettings" value="true"/>
       …
   </appSettings>
   ```

2. Login to the Admin application, create a user group (e.g., Dynamic_Group), add some rights, and assign any document types to them.
3. Open **System Settings > System Preferences > Dynamic Login Settings**, expand the **Groups** for dynamic users section, and assign the newly created user group to the **Selected User Groups list**.



4. Click the **Save** button.

DYNAMIC LOGIN INTERFACE SETTINGS

To specify dynamic login interface settings

1. Copy the folder with the Dynamic Login web application to the IIS webapps directory (`..\inetpub\wwwroot`). Run IIS Management Studio and convert the DynamicLogin folder to the web application. Restarting the IIS service after converting is recommended.
2. In the **Web.config** file of the Client application, replace the **DYNAMICLOGIN_ENDSESSION_URL** parameter's local host and port settings with values that match your local machine settings.

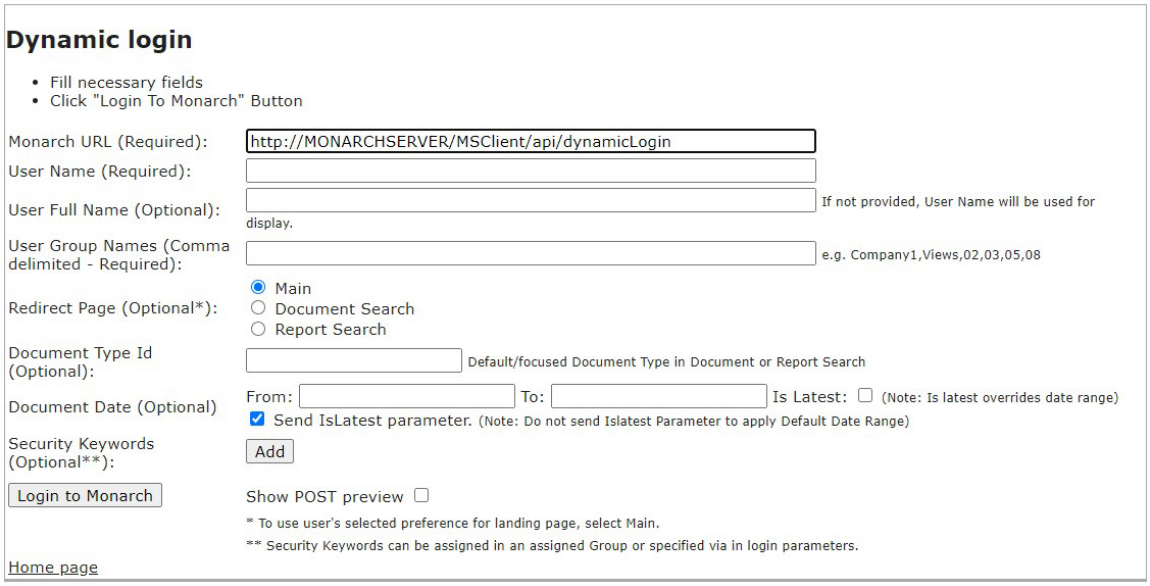
   The path to the **Web.config** file is C:\inetpub\wwwroot\MSClient\Web.config.

   The **DYNAMICLOGIN_ENDSESSION_URL** parameter is located in the <appSettings> section:

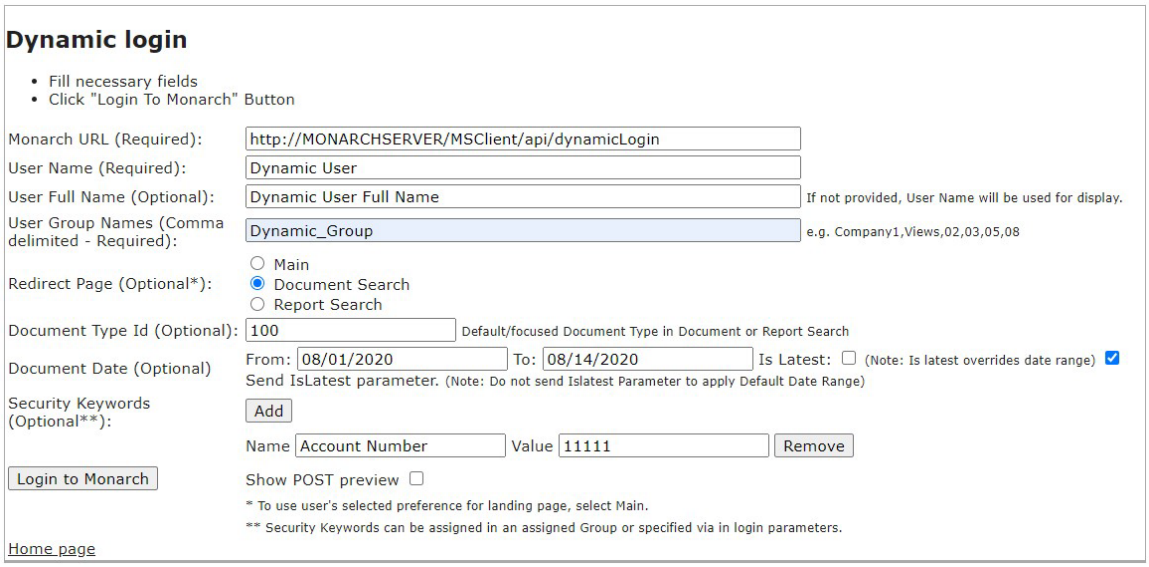
   ```
   <appSettings>
      <add key="DYNAMICLOGIN_ENDSESSION_URL"
   value="http://localhost/DynamicLogin/DynamicUserApi/Login"/>
      …
   </appSettings>
   ```

3. After editing the web.config file, restart the IIS service.
4. Open the DynamicLogin application in the browser window (e.g., http://localhost/DynamicLogin/DynamicUserApi/Login).

   The Dynamic Login to Monarch form should be opened:



5. The following fields should be filled manually:
   - Required:
     - **Monarch URL** – Enter the Monarch Server Report Warehouse Client Dynamic Login URL.
     - **User Name** – Enter any value to be used for dynamic user creation in the dynamic login user group (Dynamic_Group).
     - **User Group Name** – Enter the names of the Groups of which the User will be a member. Separate user group names with a comma (**,**).
     - **Redirect Page** – Select the page on which the User will be redirected after successful login if the User Group has the corresponding rights; otherwise, the My Home page opens.
   - Optional:
     - **User Full Name** – Enter some value for the full name of the dynamic user.
     - **Document Type Id** – Enter the ID value of some existing document types (for which the User Group has rights).
     - **Document Date** – Enter some correct date range, or select “Is Latest” checkbox.
     - **Security Keywords** – Click the “Add” button to add the new Security Keyword -> enter the name of some index field and the value of this field.
6. **Show POST preview** – Select the checkbox to display the POST request with specified login settings.



7. Click Login to Monarch button.

After performing these steps, you should be logged in to the MS Report Warehouse Client application automatically with a Dynamic Login User Group, and the corresponding Security Keywords should be applied.

CONTACT US

GET IN TOUCH

We'd love to hear from you. Here's how you can reach us.

SALES CONTACT INFORMATION

Portal: Contact Altair

US: + 1.800 445 3311

International: + 1 978 441 2200

TECHNICAL SUPPORT

Portal: Customer Support | Altair