



Altair® Monarch® v2020.0
MONARCH SERVER
CONTENT API GUIDE

TABLE OF CONTENTS

Introduction	1
Principle and purpose	1
Available operations.....	1
API testing.....	1
Working with the Monarch Server API	2
Request authentication	2
API entry point.....	2
Response type and error handling	2
Monarch Server API Usage Sample	3
API reference.....	4
Login	4
Logging in	4
Autologin.....	4
Logging out.....	5
SummaryExport	6
Export to PDF	6
Export to XLS.....	8
Export raw data (XML).....	9
Export — HTML View	13
ReportExport.....	16
Export to PDF	16
Export raw data (XML).....	17
Export to PRF	19
Export — HTML View	21
Export to ES Style.....	22
Model	24
GET method	24
Report	25
GET method	25
POST method.....	31
Search.....	37
Search by criteria.....	37
Saved search.....	45
Saved search list	50
TableExport.....	56
Export to PDF	56
Export to XLS.....	58
Export raw data (XML).....	61

Export — HTML View	65
Export to ES Style.....	68
Dynamic Login	71

INTRODUCTION

Monarch Server (MS) provides the API for external applications based on the HTTP protocol.

PRINCIPLE AND PURPOSE

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

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 on the interactive API help page located here:

`http://localhost/MSCClient/api/help`

You can view detailed information about each method on the method's page. You can then test it by clicking the **Test API** button.



NOTES

In Monarch Server v2020.0 that is upgraded from previous versions (e.g., v13.2), the virtual folders could still be labeled as "DSCClient" (or "DSAdmin"). In this case, use "DSCClient" (or "DSAdmin") instead of "MSCClient" (or "MSAdmin") to open the Client (or Admin) page.



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={sortName}&filterName={filterName}&modelName={modelName}&joinPassword={joinPassword}
```

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}&modelName={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
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}&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

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	Collection of integers	Required min length: 1

Name	Description	Type	Additional information
	the documents requested are of more than one type, an error is returned.		

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>
```

```

</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	String	None

Name	Description	Type	Additional information
	slow).		
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
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	String	None

Name	Description	Type	Additional information
	passed (however, this method is slow).		

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	Collection of integers	Required min length: 1

Name	Description	Type	Additional information
	more than one type, an error is returned.		

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

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 <code>ReportDocumentResponse</code> (see below)	None
Annotations	Includes all annotations for this report.	Collection of <code>AnnotationResponse</code> (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

Name	Description	Type	Additional information
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>
      </AnnotationResponse>
    </Annotations>
  </ReportDataResponse>
</ArrayOfReportDataResponse>

```



```

    <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

Name	Description	Type	Additional information
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

Name	Description	Type	Additional information
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	Collection of SearchCriterionRequest	None

Name	Description	Type	Additional information
	types and keyword values.	(see below)	
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

Name	Description	Type	Additional information
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>
  </SearchDocumentTypeGroupResponse>

```

```

        <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>
</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 <code>SearchDocumentTypeResponse</code> (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>
            </Documents>
        </SearchDocumentTypeResponse>
    </DocumentTypeResponses>
</SearchDocumentTypeGroupResponse>

```

```

    <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 <code>SearchResponse</code>	None

SearchResponse

Name	Description	Type	Additional information
Id	Database ID	Integer	None
Name	Search name	String	None
Rights	Rights on this search	<code>SearchGroupRights</code> (see below)	None
DocumentTypes	Document types assigned for this search	Collection of <code>DocumentTypeResponse</code> (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 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/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

Name	Description	Type	Additional information
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 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}&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 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 from a report can contain additional columns from an external database).	String	None

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)

```
{
  "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

Name	Description	Type	Additional information
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)

```
{
  "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 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)

```
{
  "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	String	None

Name	Description	Type	Additional information
	exist in the report 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

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

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)

```
{
  "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
DocumentTypeId	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	Optional

Name	Description	Type	Additional information
		below)	

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,
  "DocumentTypeId": 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.

CONTACT US

GET IN TOUCH

We'd love to hear from you. Here's how you can [reach us](#).

SALES CONTACT INFORMATION

US: + 1.800.445.3311

International: + 1.978.441.2200

Sales Email

US: sales@datawatch.com

Europe: sales_euro@datawatch.com

Asia Pacific: sales_apac@datawatch.com

SUPPORT CONTACT INFORMATION

Customer Portal: <https://support.datawatch.com>

Email: support@datawatch.com

US: +1 800.988.4739

Canada: +1 978.275.8350

Europe, Middle East, Africa: +44 (0) 8081 892481