SAP Ariba

Feature at a Glance

Analytical Reporting API Speed and Throughput Enhancement

Andy Rubinson, SAP Procurement Product Success
Target GA: May, 2022

PUBLIC

THE BEST RUN SAP

Ease of implementation Low touch / simple
Geographic relevance Global

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Customer challenge

Current Analytical Reporting API severely throttles the data extraction, resulting in slow performance that negatively impacts customer experience.

Meet that challenge with SAP Ariba

Async calls to SAP Ariba Spend Analysis database will deliver up to 50K records per zip file for non-vector views. Enables users to get more data in less time for their customer managed analytics.

Experience key benefits

Faster analytical reporting API will allow bulk exports with relaxed throttles, resulting in improved satisfaction and faster time to value.

Solution area

SAP Ariba developer portal
SAP Ariba APIs
SAP Ariba Sourcing
SAP Ariba Contracts
SAP Ariba Invoice Management
SAP Ariba Catalog
SAP Ariba Buying and Invoicing
SAP Ariba Buying
SAP Ariba Spend Analysis
SAP Ariba Supplier Information and Performance Management

Implementation information

This feature is automatically on for all customers with the applicable solutions but requires **customer configuration**.

Prerequisites and Restrictions

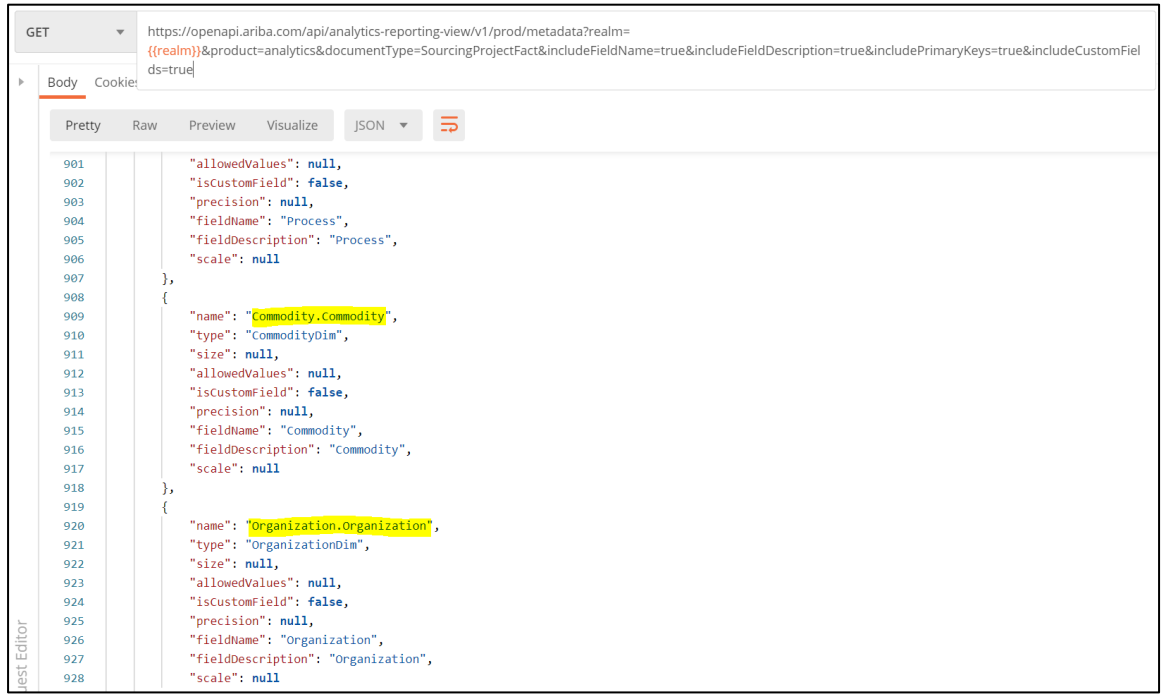
- Applies to ASYNC Analytical Reporting API
- Applies to views that do not have vectors, i.e., multi-value list fields

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Detailed feature information

- The ASYNC Analytical Reporting API output depends on whether the view contains vector or non-vector fields.
- If the view has a vector field, it will deliver a maximum of 5,000 records. If there are no vector fields in the view, then it will deliver a maximum of 50,000 records per zip file.
- Vector fields can be identified by format “field.field” in the selectFields section of the Metadata API response, as highlighted here.
- SourcingProjectFact vector fields are highlighted in this example. If you use the SourcingProjectFactSystemView template you will get the vector fields and the standard 5,000 records per zip file for your ASYNC call. If the vector fields are not needed, and you wish to boost performance, then create a custom view without these fields.

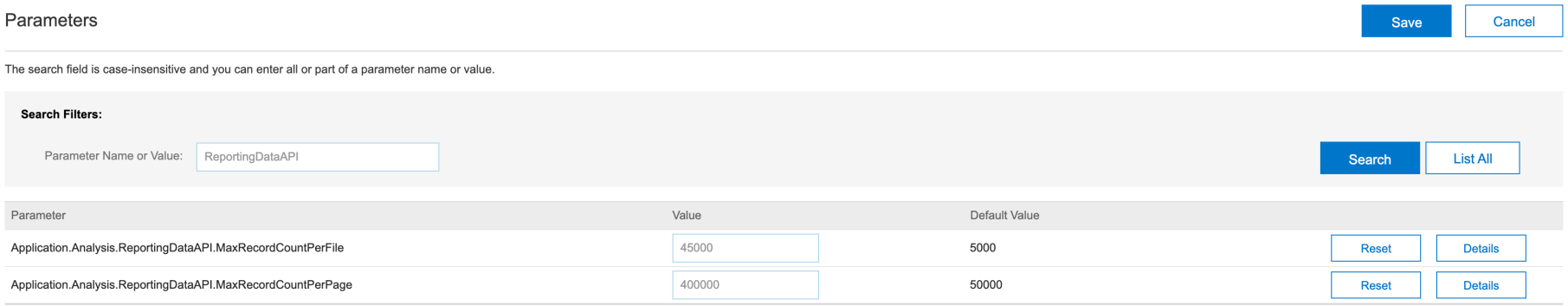



```
"selectAttributes" : ["ContractMonths", "AllOwners.AllOwners", "LoadCreateTime", "Region.Region",
"ProcessStatus", "Description", "DueDate", "AwardJustification", "Suppliers.Suppliers", "Origin",
"DependsOnProject", "Process", "ProjectId", "Commodity.Commodity", "AclId", "EndDate",
"PlannedEventType", "SourceSystem", "ActualSaving", "PlannedEndDate", "ContainerProject", "Status",
"PlannedStartDate", "OnTimeOrLate", "ProjectReason", "ProjectInfo", "TargetSavingsPct",
"SourcingMechanism", "IsTestProject", "ResultsDescription", "Owner", "Organization.Organization",
"Duration", "BaselineSpend", "State", "EventType", "ContractEffectiveDate", "LoadUpdateTime",
"ExecutionStrategy", "BeginDate"]
```

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Maximum record count per page and per file in non-vector analytical reporting API request



- **Application.Analysis.ReportingDataAPI.MaxRecordCountPerFile** parameter specifies the maximum number of records saved in one ZIP file for a non-vector analytical reporting API request.
  - Default is 5,000
  - Range for the parameter is between 1,000 and 50,000
  - Customers can increase the value up to 50,000 to improve the speed and throughput of non-vector analytical reporting API requests

ID	Application.Analysis.ReportingDataAPI.MaxRecordCountPerFile
Name	**Maximum record count per file in non-vector analytical reporting API request**
Default value	5000

- **Application.Analysis.ReportingDataAPI.MaxRecordCountPerPage** parameter specifies the maximum number of records per page that can be requested per non-vector analytical reporting API request.
  - Default is 50,000
  - Range for the parameter is between 10,000 and 500,000
  - Customers can increase the value up to 500,000 to improve the speed and throughput of non-vector analytical reporting API requests

ID	Application.Analysis.ReportingDataAPI.MaxRecordCountPerPage
Name	**Maximum record count per page in non-vector analytical reporting API request**
Default value	50000

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Performance Impact – 10x or more*

Analytical reporting API mode	Number of records to extract	Number of pages needed (jobs)	Number of zip files to download	Time to extract all data
				With standard rate limits (submit up to 8 jobs per hour and up to 40 jobs per day)
Before: Existing performance	10M	200	2000	5 days
After: With feature toggle enabled and parameters set to 500K records per page and 50k records per file	10M	20	200	5 hours

* Example improvement seen in testing. Improvement may vary based on customer configuration and landscape

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Pagination allows greater throughput

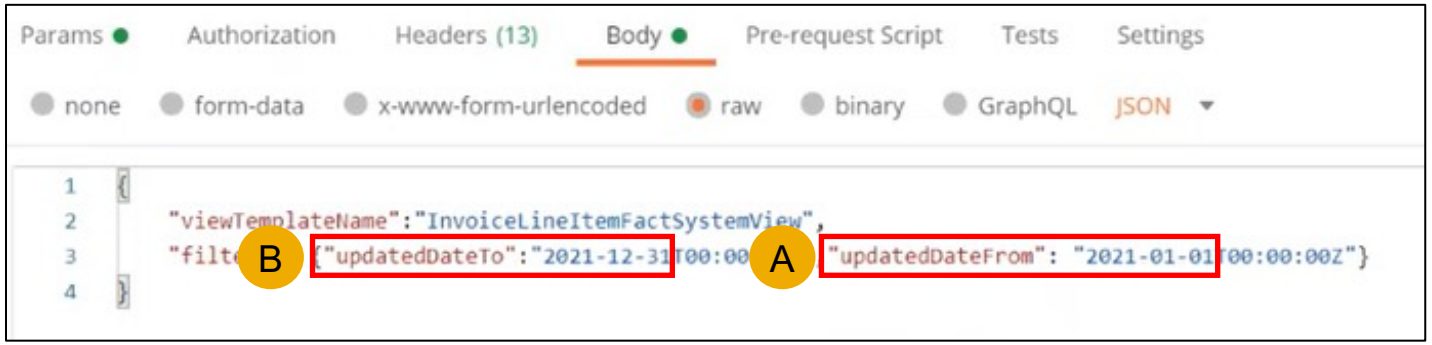
- When the response to an asynchronous reporting API call contains greater than the system maximum records, all records can still be retrieved with pagination.
- Records in the results set are separated into pages, with each page containing 50,000 records, and each page can be requested in a separate API call.
- The pagination feature must be used when more than 50,000 records are included in order to retrieve all records via API.
- Customers will be provided a page token that can be used to submit a job to get the next page (in case the number of records are more than a single page size).
- The response to each API call contains the one page of records, a page token that can be passed in the next query to retrieve the next page of records, and several new fields to help navigate and enumerate the results set.
- In addition, the reporting API result response will also contain a summary of the total number of records like number of pages, number of files to be downloaded, number of records, current page, etc.
- Pagination is supported for SYNC APIs as well and they follow the same record count limits as Async APIs. Current changes of reading the new ICM parameter are made only for non-vector Analytical Async APIs. All other reporting APIs still read the system parameter
  - System.ReportingDataAPI.MaxRecordsPerFile
  - System.ReportingDataAPI.MaxRecordsPerRequest
- More details on pagination available in the SAP Ariba Applications Q4 2019 release guide.

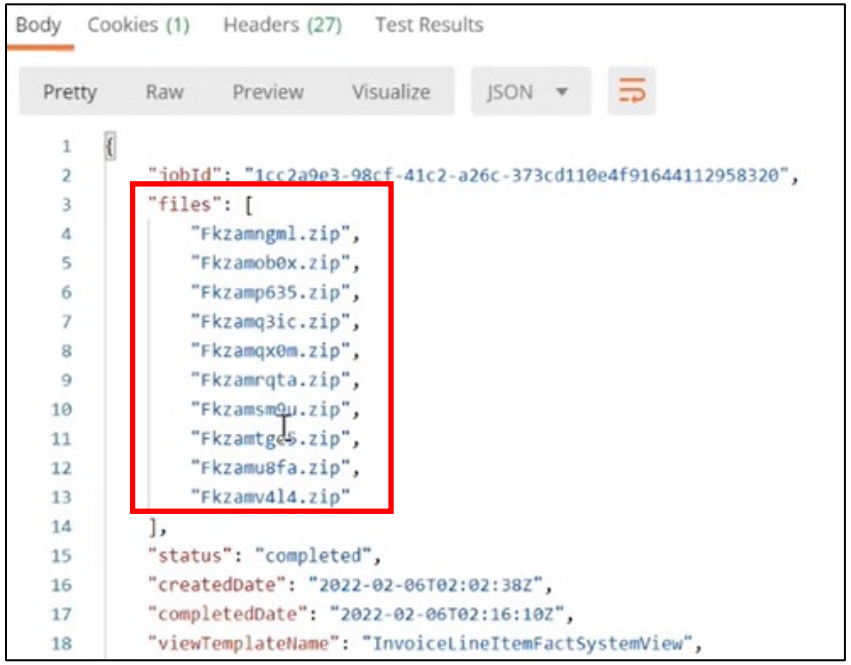
Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Pulling data via API using pagination

- Once you are authorized, you may submit a job to retrieve the desired information
  - Image shows user specifying all invoices from Jan 1, 2021 (A) – Dec 31, 2021 (B)
- Once pull is done processing, the list of zip files containing the requested records is shown.





Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Pulling data via API using pagination

- In addition to showing the document requested, InvoiceLine ItemFact in this case, you can also see a listing of the different fields included in the records contained in the zip files.

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Pulling data via API using pagination

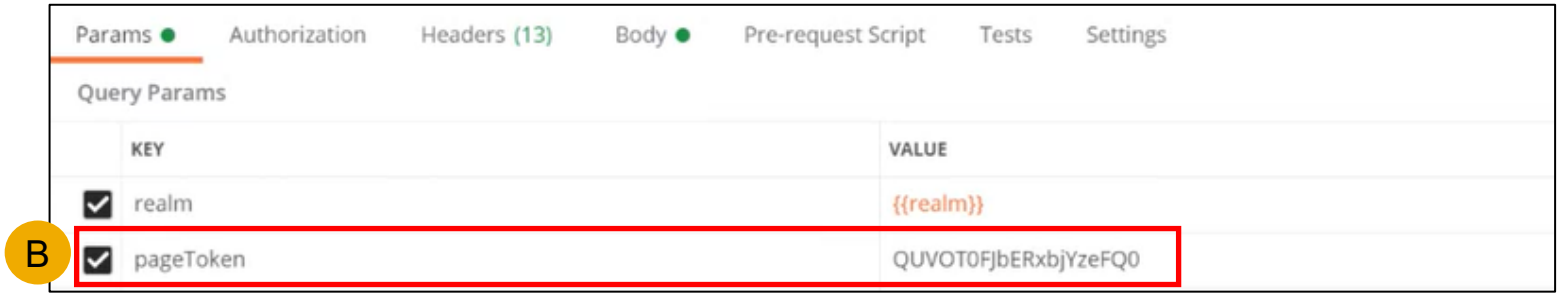
- At the end of the API processing information, some key information is shared:
  - “pageToken” – used for requesting subsequent pages in the pull (see next slide)
  - “totalNumOfRecords” – total records pulled – 12,694,443
  - “currentPageRecordsCount” – indicates the max limit of 500,000 as described on slide 4
  - “totalNumOfPages” – indicates a total of 26 pages were pulled
  - “currentPageNum” – indicates where you are in the pulling of pages. Once that number reaches 26, you’re at the end
  - “totalNumOfFiles” – indicates the total number of zip files pulled across the 26 pages
- Once each zip file is downloaded, you can continue on by requesting the next page as shown in the following slide.

Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Pulling data via API using pagination

- Using the page token (A) shown on the prior slide, you can submit the next analytical reporting API job, copying and pasting the page token, in a new POST call (B) and sending.

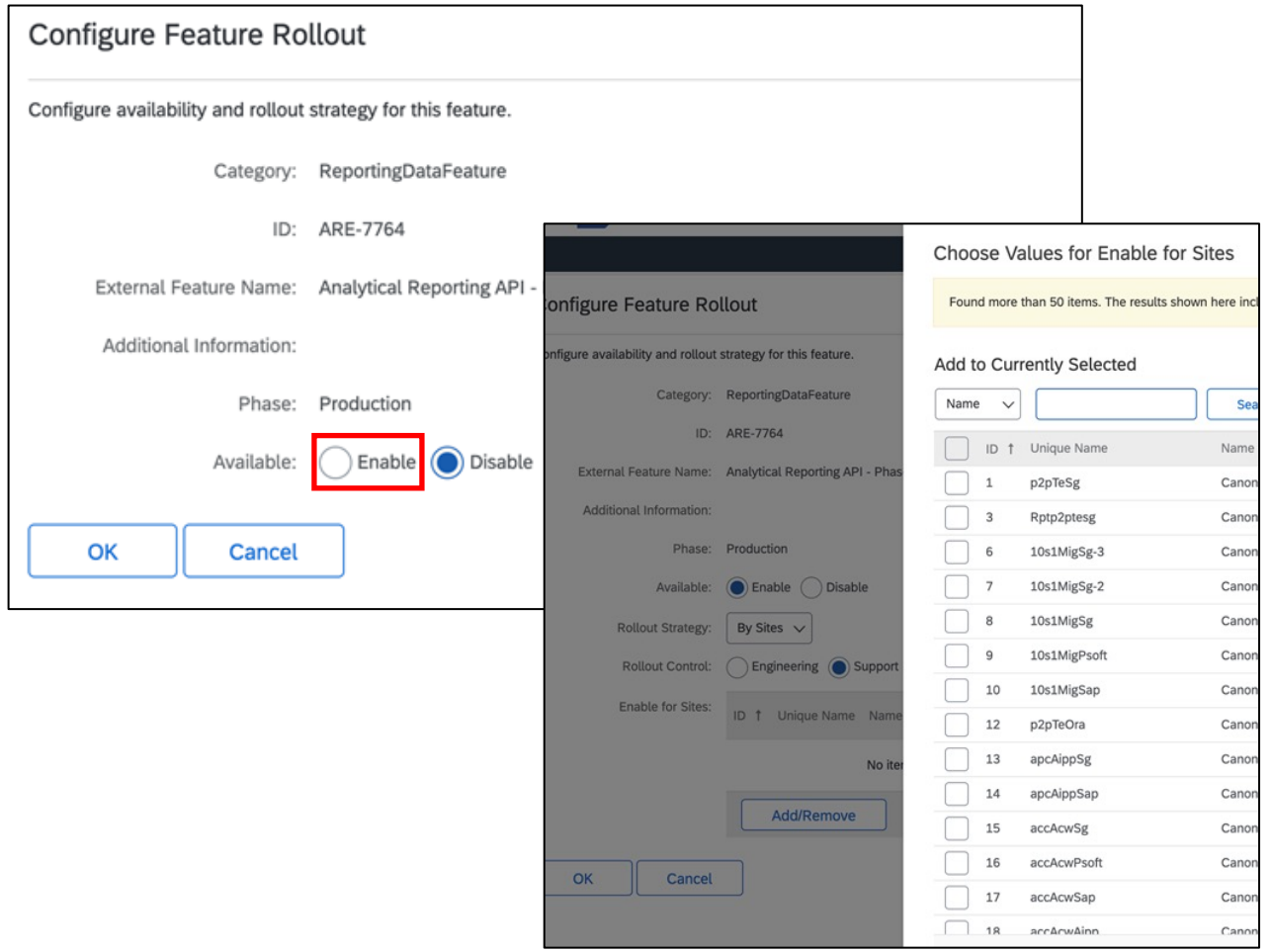


Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Toggle and Parameters

- By selecting Enable in the top left image, the feature will become visible.
- After selecting Enable, users may search for realms and add.
- Please note:
  - The toggle is only used before release to allow customers to access for Early Adopter Care.
  - Once released, the toggle will be on for all customers and will no longer be necessary to configure.



Feature at a Glance

Introducing: **Analytical Reporting API Speed and Throughput Enhancement**

Enablement

Feature Toggle:

```
id = "ARE-7764"
name = "Analytical Reporting API – Phase 1 – Faster extraction for non-vector facts/dims"
```

ICM Parameters:

```
   <parameter name="Application.Analysis.ReportingDataAPI.MaxRecordCountPerPage" type="int"
defaultValue="50000" minVal="10000" maxVal="500000" acl="selfservice">
        <description>
            <![CDATA[<P>This section defines the max records that can be requested per non-vector analytical
reporting API request</P>]]>
        </description>
   </parameter>

   <parameter name="Application.Analysis.ReportingDataAPI.MaxRecordCountPerFile" type="int" defaultValue="5000"
minVal="1000" maxVal="50000" acl="selfservice">
        <description>
            <![CDATA[<P>This section defines the max records saved in one file for non-vector analytical
reporting API request</P>]]>
        </description>
   </parameter>
```