



PUBLIC

Document Version: 2608 – 2026-08

Transaction Monitoring API

Content

Transaction Monitoring API.	3
About Transaction Monitoring API.	4
Workflow for Transaction Monitoring API.	5
Performance of the Prerequisite Steps for Using the Transaction Monitoring API.	6
Use of the Transaction Monitoring API.	6
Interpretation of the Query Response.	13
Error Messages Returned By Transaction Monitoring API.	17
Version History for the Transaction Monitoring API.	19
API-Specific Disclaimers and Legal Information.	20

Transaction Monitoring API

This guide is for developers seeking to use the Transaction Monitoring API to monitor transactions in SAP Business Network

Note

With the 2211 (November 2022) release, the Transaction Monitoring API will be deprecated. It will be decommissioned one year from the date of deprecation, and will cease to function. The Integration Event Monitoring Query API for Procurement provides equivalent functionality to customers who are already using or are planning to use the Transaction Monitoring API.

Buyers enabled for SAP Business Network for Supply Chain can monitor SAP Business Network transactions across the network, allowing a high degree of visibility and traceability into associated business processes and documents.

This guide applies to:

- SAP Business Network for Supply Chain
- SAP Business Network

Related Information

[Integration Event Monitoring Query API for Procurement](#)

About Transaction Monitoring API

The Transaction Monitoring API offers the following capabilities:

- **End-to-end lineage query:** Searches can be configured on Correlation ID to trace the lineage of a transaction. This allows for end-to-end visibility of transactions.
- **Time-bound query:** Searches can be configured based on specified time periods. This returns all transactions within the specified time range.
- **Non-time-bound query:** Searches can be configured based on searchable fields like Document ID and Correlation ID. This returns all transactions corresponding to the specified searchable fields.
- **Pagination support:** Pagination can be enabled to support larger search windows for time-bound queries with high data volumes. This displays the transactions in multiple pages, each with a predefined number of transactions.

To ensure granular visibility, SAP Business Network has implemented a grid of checkpoints that reflect events. Each time a business message passes a checkpoint the respective event changes the document's processing status and all relevant descriptive parameters of the business message are recorded internally. The actual status of the business messages is managed by the Transaction Monitoring Repository.

The Transaction Monitoring API allows external users to access the data records in a secure and efficient way to feed any local application in order to track and trace messages so that they can be correlated with backend data. Additionally, with periodic queries, response data can be used to help with local exception monitoring and root cause analysis.

Prerequisites

Before you can use the Transaction Monitoring API, you must:

- Complete the onboarding process on the SAP Ariba Developer Portal
- Ensure that your application conforms to expected OAuth authentication. Only applications approved by the SAP Ariba APIs administration can execute OAuth authentication.

Restrictions

The Transaction Monitoring API supports monitoring of events for SAP Business Network Buyer accounts only. The API does not support Supplier accounts.

Considerations When Using the Transaction Monitoring API

- If there are multiple purchase orders (PO) for a single ship notice, the Transaction Monitoring API creates a separate record for each PO. These records are created at intervals of one nanosecond. This is done to prevent the previous record being overwritten by the next one, as both records would have the same time of creation.
- SAP Business Network does not support Correlation IDs for some documents. The Transaction Monitoring API does not monitor transactions for such documents.
- The Transaction Monitoring API monitors events that are identified by SAP Business Network as public events. It does not monitor internal events. If an event has to be monitored, administrators must mark the event as public.
- The Transaction Monitoring Repository stores data for the preceding fifteen days only. If a document that is linked to others by the same Correlation ID has a date of creation outside this period, it is removed from the database.

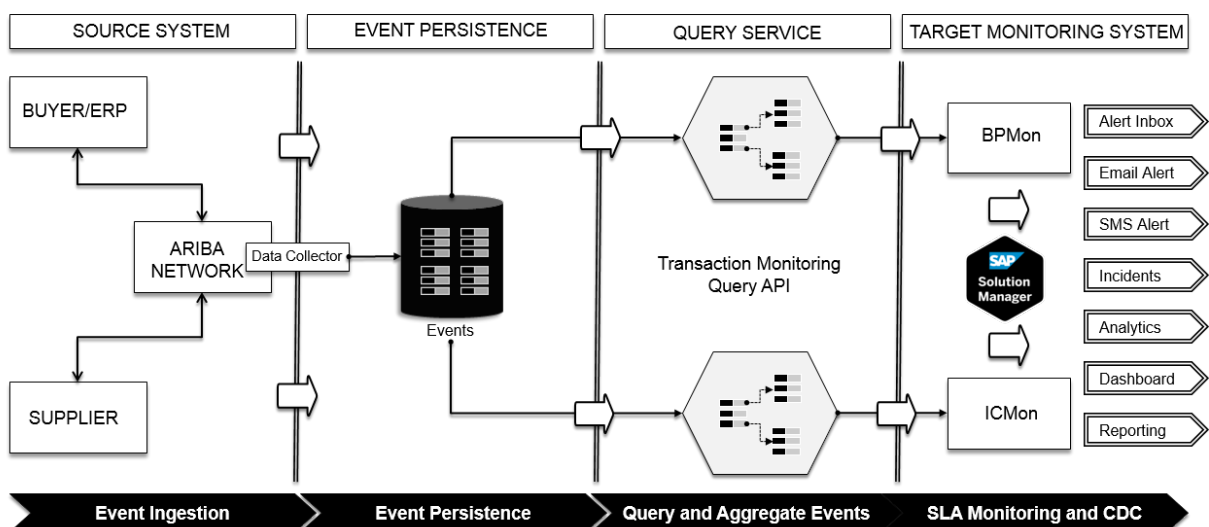
[Workflow for Transaction Monitoring API \[page 5\]](#)

Workflow for Transaction Monitoring API

The high-level workflow for this feature covers the query, response, and filtering steps.

1. The customer's client application must query the SAP Business Network Transaction Monitoring Repository using the Transaction Monitoring API with a defined set of parameters.
2. In response to the query, the API returns a JSON data set that contains all matching records in a formatted structure. A particular business message is described by multiple events and a collected set of details.
3. The receiving application then filters and processes the data.

Transaction Monitoring REST API Architecture



[Performance of the Prerequisite Steps for Using the Transaction Monitoring API \[page 6\]](#)

[Use of the Transaction Monitoring API \[page 6\]](#)

Performance of the Prerequisite Steps for Using the Transaction Monitoring API

The following steps are required to start using the Transaction Monitoring API with an SAP Ariba cloud solution.

Steps for SAP Business Network

The following steps are required to use the Transaction Monitoring API with SAP Business Network:

1. Create SAP Ariba OpenAPI account.
To create your account, access the SAP Ariba Open API portal (<https://developer.ariba.com/api/login>), click **Don't have an account yet?** and complete the **Registration Request Form**. Alternatively, contact your SAP Ariba Customer Success Manager or Account Manager to create the account.
2. Provision BPM OpenAPI and BPM integration (client-ID to ANID mapping).
To create the client-ID mapping, contact your SAP Ariba Customer Success Manager or Account Manager.

Use of the Transaction Monitoring API

Transaction Monitoring API Specification

This section lists the available REST APIs defined in the Transaction Monitoring API specification.

HTTP methods

The Transaction Monitoring API uses HTTP verbs for every action.

Method	Description
GET	Used for retrieving resources.

Query filters

Query Filter	Requirement	Description
start_date_time	Mandatory	Start date and time of the event date range. Date and time is specified in epoch format, in seconds. This indicates the start of a time interval to be used as a filter for the search query.
end_date_time	Mandatory	End date and time of the event date range. Date and time is specified in epoch format, in seconds. This indicates the end of the time interval applied as a search filter.

Query Filter	Requirement	Description
<code>alert_type</code>	Optional	Event status (error or success).
<code>process_name</code>	Optional	The specific business document that is tracked in an event. Processes relating to the following documents are tracked: - TRANSACTION_TRACKING_PURCHASEORDER - TRANSACTION_TRACKING_CONFIRMATIONDOCUMENT - TRANSACTION_TRACKING_SHIPNOTICEDOCUMENT - TRANSACTION_TRACKING_INVOICEDetailREQUEST
<code>sender_id</code>	Optional	Identity of the sender of the business document. This is a unique SAP Business Network ID for the sender. For example, the sender's SAP Business Network ID.
<code>receiver_id</code>	Optional	Identity of the receiver of the business document. This is a unique SAP Business Network ID for the receiver. For example, the receiver's SAP Business Network ID.
<code>transaction_id</code>	Optional	The unique identifier of the transaction. This is a globally unique and globally shared transaction ID for the particular process. Payload ID is the unique identifier for a given transaction in SAP Business Network. Payload ID is mapped to <code>transaction_id</code> in the Transaction Monitoring API. You can use <code>transaction_id</code> to link to backend systems, to see an end-to-end view of a transaction.
<code>groupby</code>	Optional	The field used to trace the lineage of a document. This field is used to identify documents related to each other. The following fields are supported for use with the <code>groupby</code> filter: <code>transaction_id</code>, <code>receiver_id</code>, <code>event_name</code>, <code>correlation_id</code>, <code>document_id</code>, <code>alert_type</code>, and <code>sender_id</code>.

The Transaction Monitoring API supports a single primary parameter, *rsql/filter* which embeds all parameters. Embedded parameters can be combined with an *and* clause by using the 'and' keyword or ';' in accordance with the rsql standard. For examples of how the primary parameter is used, see [Transaction Monitoring API Request Examples \[page 11\]](#).

How to Use the Transaction Monitoring API

API requests are made by accessing the following URL on any HTTP client or REST API test:

```
https://openapi.riba.com/api/bizmonservice/v2/prod
```

The Transaction Monitoring API comprises REST services on the SAP Ariba Open API platform. Contact your SAP Ariba Customer Success Manager or Account Manager for information about setting up OAuth to authenticate applications on the SAP Ariba Developer Portal.

How to Use Pagination in Time-Bound Searches

The pagination feature to support large data volumes is available with the following URL:

```
https://openapi.ariba.com/api/bizmonservice/v2/prod
```

This feature controls the number of transactions returned per page, in a JSON response to a time-bound query. The default page size is 10,000.

You can limit the page to a manageable size using the PageSize and Offset HTTP headers. You send the HTTP header PageSize with an integer value for the required page size. You send the HTTP header Offset with the starting point for the subsequent page. This value is obtained from the response to the first request. For example, if the first request specified the PageSize as *n*, the response will have the HTTP header 'Offset: *n*+1'. Therefore, the second request will be sent with the HTTP header Offset value '*n*+1'.

The following example shows the use of the HTTP header Offset. In this example, the request has the Offset header set to 101. This would have been the value returned in the response to the previous request, sent with the Offset header value as 100.

```
"requests": [
  {
    "id": "b9e12707-2bb4-f10b-ae6c-b8862694b093",
    "headers": "Authorization: Bearer
49f435ed-1da9-47e4-8271-79e6bc274e98\n//Offset: 101\n",
    "url": "https://openapi.ariba.com/api/bizmonservice/v2/prod?
rsqlfilter=(start_date_time== 1504591680 and end_date_time== 1504591980 and
groupby== correlation_id,transaction_id )",
    ...
  }
]
```

[Transaction Process Monitoring Event Data Elements \[page 8\]](#)

[Transaction Monitoring API Request Examples \[page 11\]](#)

Transaction Process Monitoring Event Data Elements

Data is recorded for events in the Transaction Monitoring Repository.

Element Name	Description
date_happened	The event date, which is the date and time when the event was triggered. Dates are in epoch format.
text	This represents comments that can be added to an event by a cloud application to provide additional information about the event. For example, this might contain an explanation of why a process failed.
changed_by	The name of the application/node of the published event.

Element Name	Description
process_name	The specific business document that is tracked in an event. Processes relating to the following documents are tracked: - TRANSACTION_TRACKING_COMPONENTRECEIPTREQUEST - TRANSACTION_TRACKING_COMPONENTSHIPNOTICEDOCUMENT - TRANSACTION_TRACKING_APPROVALREQUEST - TRANSACTION_TRACKING_COMPONENTCONSUMPTIONREQUEST - TRANSACTION_TRACKING_CONFIRMATIONDOCUMENT - TRANSACTION_TRACKING_COPYREQUEST - TRANSACTION_TRACKING_INVOICEDetailREQUEST - TRANSACTION_TRACKING_PURCHASEORDER - TRANSACTION_TRACKING_ORDERSTATUSREQUEST - TRANSACTION_TRACKING_PRODUCTACTIVITYMESSAGE - TRANSACTION_TRACKING_PRODUCTREPLENISHMENTMESSAGE - TRANSACTION_TRACKING_QUALITYINSPECTIONDECISIONREQUEST - TRANSACTION_TRACKING_QUALITYINSPECTIONREQUEST - TRANSACTION_TRACKING_QUALITYINSPECTIONRESULTREQUEST - TRANSACTION_TRACKING_QUALITYNOTIFICATIONDOCUMENT - TRANSACTION_TRACKING_RECEIPTREQUEST - TRANSACTION_TRACKING_SALESORDERREQUEST - TRANSACTION_TRACKING_SHIPNOTICEDOCUMENT - TRANSACTION_TRACKING_STATUSUPDATEREQUEST - TRANSACTION_TRACKING_TRANSPORTCONFIRMATION

Element Name	Description
event_name	The event name that indicates the document processing stage. It can be one of the following values: - DOCUMENT_RECEIVE_FAILED - DOCUMENT_RECEIVED - PROCESSING_SUCCESSFUL - PROCESSING_DELAYED - PROCESSING_FAILED - GATEWAY_DELIVERY_SUCCESSFUL - GATEWAY_DELIVERY_DELAYED - GATEWAY_DELIVERY_FAILED - DELIVERY_SUCCESSFUL - DELIVERY_DELAYED - DELIVERY_FAILED - DELIVERY_READY - CUSTOMER_CONFIRMED - CUSTOMER_RECEIVED - CUSTOMER_FAILED - POSTED_TO_UI_EXTRACTION - UI_EXTRACTION_INITIATED - UI_EXTRACTION_IN_PROGRESS - UI_EXTRACTION_DELAYED - UI_EXTRACTION_FAILED - UI_EXTRACTION_COMPLETED
sender_id	The identity of the sender of the business document. This is a unique SAP Business Network ID for the sender. For example, the sender's SAP Business Network ID.
receiver_id	The identity of the receiver of the business document. This is a unique SAP Business Network ID for the receiver. For example, the receiver's SAP Business Network ID.
transaction_id	The unique identifier of the transaction. This is a globally unique and globally shared transaction ID for the particular process. Payload ID is the unique identifier for a given transaction in SAP Business Network. Payload ID is mapped to the transaction_id parameter in the Transaction Monitoring API. So transaction_id can be used to link to backend systems to see an end-to-end view of a transaction.
document_id	The unique identifier of the business document that is being tracked in an event. This could contain the value of one of the following identifiers: OrderID, ConfirmID, ShipmentID, and InvoiceID.
system_id	The unique identifier used by SAP Business Network to identify the ERP system.
order_date	The date on which the order was created in the source system. This field is mapped either to the Document Date or the Reference Document Date. It is mapped to Reference Document Date to trace the lineage in PO change instances as PO change does not contain orderReference information.

Element Name	Description
correlation_id	The PO ID from the source system. This helps in easy identification of the PO numbers that are affected. This parameter is specified only when the document is a change PO, a ship notice, or an order confirmation.
reference_payload_id	The cXML Payload ID for the PO. This helps associate the PO to other associated business documents, such as an order confirmation or ship notice.
status	The business document's processing status. This indicates the success or failure of document processing. It also includes warnings, if there are any.
error_message	The error information from SAP Business Network, which contains more information about why an error has occurred. This information enables the user to quickly identify corrective actions.
source	The source of the event.
comment	The comment generated from the transaction checkpoint.
title	The title is a combination of process name and event name.

Transaction Monitoring API Request Examples

The HTTP REST API requests in this section use endpoints with filters to get a response from the server.

Request Examples for Time-Bound Queries

The following request examples are for time-bound queries with different filters:

Filter By	API Request Example
Start date and end date	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180)</pre>
Sender	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and sender_id==ANBPM001)</pre>
Receiver	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and receiver_id==ANBPM001)</pre>
Alert type	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and alert_type==FAILURE)</pre>

Filter By	API Request Example
Alert type with in clause	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and alert_type=in=(FAILURE,WARNING))</pre>
Transaction ID	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504362300 and transaction_id==Txn-1456236345637)</pre>
Document ID	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1501595688 and end_date_time==1501595692 and document_id==POC914198)</pre>
Process name	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1498631024 and end_date_time==1498631024 and process_name==TRANSACTION_TRACKING_PURCHASEORDER)</pre>
Group by	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1501078325 and end_date_time==1501078330 and groupby==correlation_id)</pre>
Process name and group by	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1501078325 and end_date_time==1501078330 and process_name==TRANSACTION_TRACKING_PURCHASEORDER and groupby==correlation_id)</pre>
Sender and process name	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and sender_id==ANBPM001 and process_name==TRANSACTION_TRACKING_PURCHASEORDER)</pre>
Receiver and process name	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and receiver_id==ANBPM001 and process_name==TRANSACTION_TRACKING_CONFIRMATIONDOCUMENT)</pre>
Alert type and process name	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and alert_type==FAILURE and process_name==TRANSACTION_TRACKING_PURCHASEORDER)</pre>

Filter By	API Request Example
Alert type, receiver, and process name	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(start_date_time==1504362300 and end_date_time==1504371180 and alert_type==SUCCESS and receiver_id==ANBPM001 and process_name==TRANSACTION_TRACKING_PURCHASEORDER)</pre>

Request Examples for Non-Time-Bound Queries

The following request examples are for non-time-bound queries with different filters, including search by document ID:

Filter By	API Request Example
Document ID	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(document_id==P037529)</pre>
Correlation ID	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(correlation_id==P01-1204756317410)</pre>
Document ID and group by	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(document_id==POC1-15635343463635 and groupby==alert_type)</pre>
Document ID and alert type with <code>in</code> clause	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(document_id==POC1-15635343463635 and alert_type=in=(FAILURE,WARNING))</pre>
Correlation ID and group by	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(correlation_id==P01-15635343463635 and groupby==document_id)</pre>
Correlation ID and alert type	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(correlation_id==P01-15635343463635 and alert_type==FAILURE)</pre>
Correlation ID and alert type with <code>in</code> clause	<pre>https://openapi.ariba.com/api/bizmonservice/v2/prod? rsqlfilter=(correlation_id==P01-15635343463635 and alert_type=in=(FAILURE,WARNING))</pre>

Interpretation of the Query Response

HTTPS Status Codes

Status Code	Description
200	OK
400	Bad request
401	Unauthorized
404	URL not found
405	Method not allowed
500	Internal error

Transaction Monitoring API Error Codes

The following is the list of application error codes returned from the Transaction Monitoring API:

Error Code	Message
1000	Error encountered executing responsibility
1100	Exception while getting CustomerId from OAuth token.
1200	Exception while processing RSQLFilter. It could be because RSQLFilter could not be parsed successfully.
1250	Exception while validating RSQLFilter. It could be because of validation failure for RSQLFilters or unsupported filter in RSQLFilters.
1251	start_date_time and end_date_time is invalid.
1252	start_date_time is invalid.
1253	end_date_time is invalid.
1254	Invalid fields in RSQL filter.
1300	Exception while encrypting input data.
1400	Exception while executing query on database.
1500	Exception while decrypting field.

[Transaction Monitoring API Response Examples \[page 14\]](#)

Transaction Monitoring API Response Examples

This topic shows JSON response examples for time-bound queries and lineage queries.

JSON Response for a Time-Bound Query

The following example shows a JSON response from a server based on the requested filters.

```
{
  "transactions": [
    {
      "transaction_id": "LQ_24_08_18_46_PayloadId",
      "document_id": "LQ_24_08_18_46_DocId",
      "sender_id": "ANBPM00001",
      "receiver_id": "ANBPM00002",
      "process_name": "TRANSACTION_TRACKING_PURCHASEORDER",
      "changed_by": "Community WebApp-SUPPLIER",
      "comment": "[INFO] - DOCUMENT_IN_PO_DISPATCHER (1260)",
      "system_id": "AN-ERP-System",
      "events": [
        {
          "correlation_id": "LQ_24_08_18_46_DocId",
          "order_date": "1504390256000",
          "document_date": "1504390256000",
          "alert_type": "SUCCESS",
          "event_name": "DOCUMENT_IN_PROPAGATION_PROCESSOR",
          "event_timestamp": "1503580605139",
          "event_id": "8f4c6ddc-d612-41e8-bb2a-9b9184b4c354",
          "host": "tmapp.tmhost.com",
          "date_happened": "1503580605139",
          "reference_payload_id": "",
          "error_message": "",
          "source": "AN",
          "attachment_document_type": ""
        }
      ]
    }
  ]
}
```

When a single ship notice contains multiple POs

This is a special case of the time-bound query.

The following example shows a JSON response from a server based on the requested filters.

```
{
  "transactions": [{
    "transaction_id": "ASN_LQ_19_33_D21_D30",
    "document_id": "ASN_For_PO_21_30",
    "sender_id": "ANBPM00001",
    "receiver_id": "ANBPM00002",
    "process_name": "TRANSACTION_TRACKING_SHIPNOTICEDOCUMENT",
    "changed_by": "Community WebApp-SUPPLIER",
    "comment": "[INFO] - DOCUMENT_IN_PROPAGATION (1110)",
    "system_id": "AN-ERP-System",
    "events": [
      {
        "correlation_id": "\"LQ_24_08_19_21_DocId\"",
        "order_date": "\"1476035524000\"",
        "document_date": "\"1476035524000\"",
        "alert_type": "FAILURE",
        "event_name": "DOCUMENT_IN_PROPAGATION",
        "event_timestamp": "1503583515906",
        "event_id": "34e5aa2f-6ee0-4d0d-b4ee-3a273740128c",
        "host": "tmapp.tmhost.com",
        "date_happened": "1503583515906",
        "reference_payload_id": "\"LQ_24_08_19_21_PayloadId\"",
        "error_message": "The quantity for this line item
falls outside the range (0 - 20.00)"
      }
    ]
  }]
}
```

```

        allowed by the buyer. Please
contact support with the Error Reference
        Number:
ANERR-500000000000000000340397015 for more details",
    "source": "AN",
    "attachment_document_type": ""
  },
  {
    "correlation_id": "\"LQ_24_08_19_30_DocId\"",
    "order_date": "\"1476035524000\"",
    "document_date": "\"1476035524000\"",
    "alert_type": "SUCCESS",
    "event_name": "DOCUMENT_IN_PROPAGATION",
    "event_timestamp": "1503583515906",
    "event_id": "87d4bffe-bd0e-4999-a3c5-c4912ebd985e",
    "host": "tmapp.tmhost.com",
    "date_happened": "1503583515906",
    "reference_payload_id": "\"LQ_24_08_19_30_PayloadId\"",
    "error_message": "",
    "source": "AN",
    "attachment_document_type": ""
  }
]
}

```

JSON Response for a Lineage Query

The following example shows a JSON response from a server based on the requested filters.

```

{
  "correlation_id": "PO-9122231728",
  "transaction_ids": [
    {
      "transaction_id": "20170827205704192+0530-9122231728",
      "document_id": "PO-9122231728",
      "sender_id": "ANBPM00001",
      "receiver_id": "ANBPM00002",
      "process_name": "TRANSACTION_TRACKING_PURCHASEORDER",
      "changed_by": "Community WebApp-BUYER",
      "comment": "[INFO] - PROCESSING_SUCCESSFUL (1200)",
      "system_id": "AN-ERP-System",
      "events": [
        {
          "correlation_id": "PO-9122231728",
          "order_date": "1476035524000",
          "document_date": "1476035524000",
          "alert_type": "FAILURE",
          "event_name": "DOCUMENT_RECEIVED",
          "event_timestamp": "",
          "event_id": "d774ec24-a136-4fe7-8138-f3d6cc9c9067",
          "host": "tmapp.tmhost.com",
          "date_happened": "1503847625775",
          "reference_payload_id": "",
          "error_message": "The quantity for this line item falls
outside the range (0 - 20.00)
        allowed by the buyer. Please contact support with
the Error Reference
        Number: ANERR-500000000000000000340397015 for more
details",
          "source": "AN",

```

```

        "attachment_document_type": ""
    }
}
},
{
    "transaction_id": "20170827205704192+0530-9122231728_asn",
    "document_id": "ASN-9122231728",
    "sender_id": "ANBPM00001",
    "receiver_id": "ANBPM00002",
    "process_name": "TRANSACTION_TRACKING_SHIPNOTICEDOCUMENT",
    "changed_by": "Community WebApp-SUPPLIER",
    "comment": "[INFO] - PROCESSING_SUCCESSFUL (1200)",
    "system_id": "AN-ERP-System",
    "events": [
        {
            "correlation_id": "PO-9122231728",
            "order_date": "1476035524000",
            "document_date": "1476035524000",
            "alert_type": "SUCCESS",
            "event_name": "DOCUMENT_RECEIVED",
            "event_timestamp": "",
            "event_id": "0d73452b-cd8e-4c36-adad-ee7ad8df83a8",
            "host": "tmapp.tmhost.com",
            "date_happened": "1503847649722",
            "reference_payload_id": "20170827205704192+0530-9122231728",
            "error_message": "",
            "source": "AN",
            "attachment_document_type": ""
        }
    ]
}
]
}
}

```

Error Messages Returned By Transaction Monitoring API

Use this information to troubleshoot your queries to the Transaction Monitoring API.

HTTP Response Code	API Response Code	Message	Additional Information
400		The API key passed in the request is not valid.	The API key passed in the query headers does not match the API key in the OAuth token.
	1100	Exception while getting CustomerId from OAuth token.	
	1200	Exception while processing RSQFilter.	RSQFilter could not be parsed successfully.
	1250	Exception while validating RSQFilter.	Failed to validate RSQFilter or filter is unsupported.

HTTP Resposne Code	API Response Code	Message	Additional Information
	1251	start_date_time and end_date_time is invalid.	
	1252	start_date_time is invalid.	
	1253	end_date_time is invalid.	
	1254	Invalid fields in RSQL filter.	
429		Rate limit error	You have exceeded the maximum permitted query rate. Refer to the developer portal discovery page for this API for the applicable query rates
500	1000	Error encountered executing responsibility.	
	1300	Exception while encrypting input data.	
	1400	Exception while executing query on database.	
	1500	Exception while decrypting field.	
504		Gateway Time-out	The server timeout time has been exceeded, possibly due to network traffic or reliability issues. Retry your query.

Version History for the Transaction Monitoring API

The Transaction Monitoring API is deprecated and might be decommissioned at a later date. We recommend the Integration Event Monitoring Query API for Procurement as an alternative to customers who are already using or planning to use the Transaction Monitoring API.

API Version	Status	As of Release
1	Deprecated	2211 (November 2022)

Related Information

[API versioning and deprecation on the developer portal](#)

API-Specific Disclaimers and Legal Information

The SAP Ariba Developer Portal included in the SAP Ariba APIs product and the APIs made available on this site are provided solely at the discretion of SAP without warranty of any kind, and SAP may change, suspend, or cancel any or all features or functions of the SAP Ariba APIs product or revise the website at any time. Any production use of or commercialization of applications containing any APIs provided on this website is prohibited without a written agreement between your company and SAP governing such activities.

Access to this API is available to you as a subscriber to this solution as part of the SAP Cloud Service Level Agreement. However, it is not considered part of the solution. Use of this API is purely optional and is subject to restrictions stated in the documentation, including the Terms of Use and the documentation found at the SAP Ariba Developer Portal (see <https://developer.ariba.com/api>). If you wish to connect a third-party service using this API, first confirm that the company is participating in the SAP partner program and is authorized to provide a connection to this solution using this API. You will be required to submit written consent to SAP to authorize the exchange of data with the third-party service.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

Copyright © 2026 Ariba, Inc. All rights reserved.

This documentation, as well as the Ariba solutions, software and/or services described in it, contain proprietary information. They are provided under a license or other agreement containing restrictions on use and disclosure and are also protected by copyright, patent and/or other intellectual property laws. Except as permitted by such agreement, no part of the document may be reproduced or transmitted in any form by any means, electronic, mechanical or otherwise, without the prior written permission of Ariba, Inc.

Ariba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the documentation. The information contained in the documentation is subject to change without notice.

Ariba and Ariba products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Ariba, Inc. in the United States and other countries. Please see <http://www.ariba.com/legal/trademarks> for additional trademark information and notices.

Ariba Sourcing solutions (On Demand and software) are protected by one or more of the following patents, including without limitation: U.S. Patent Nos. 6,199,050; 6,216,114; 6,223,167; 6,230,146; 6,230,147; 6,285,989; 6,408,283; 6,499,018; 6,564,192; 6,871,191; 6,952,682; 7,010,511; 7,072,061; 7,130,815; 7,146,331; 7,152,043; 7,225,152; 7,277,878; 7,249,085; 7,283,979; 7,283,980; 7,296,001; 7,346,574; 7,383,206; 7,395,238; 7,401,035; 7,407,035; 7,444,299; 7,483,852; 7,499,876; 7,536,362; 7,558,746; 7,558,752; 7,571,137; 7,599,878; 7,634,439; 7,657,461; 7,693,747; 8,364,577; and 8,392,317. Patents pending.

Other Ariba product solutions are protected by one or more of the following patents:

U.S. Patent Nos. 6,199,050, 6,216,114, 6,223,167, 6,230,146, 6,230,147, 6,285,989, 6,408,283, 6,499,018, 6,564,192, 6,584,451, 6,606,603, 6,714,939, 6,871,191, 6,952,682, 7,010,511, 7,047,318, 7,072,061, 7,084,998; 7,117,165; 7,225,145; 7,324,936; 7,536,362; 8,364,577; and 8,392,317. Patents pending.

Certain Ariba products may include third party software or other intellectual property licensed from a third party. For information regarding software or other intellectual property licensed from a third party, go to <http://www.ariba.com/copyrights.cfm>.

