Infor IDF REST APIs User Guide

Publication Information

Release: Infor IDF REST APIs User Guide

Publication date: April 7, 2026

Contents

Contents **3**

About this guide **5**

Contacting Infor 5

Chapter 1 Developing Rest APIs **6**

Accessing REST APIs 6

Creating a REST API 6

REST API details 7

Viewing REST API details 7

Selecting business objects for a REST API 8

Removing selected business objects from a REST API 8

Selecting business object attributes for a REST API 8

Changing the visibility of a selected business object attribute 9

Resetting business object attribute usage 9

Removing selected business object attributes from a REST API 10

Selecting business object REST Calls for a REST API 10

Deleting selected business object REST Calls from a REST API 12

Changing Selected Business Object REST Calls in a REST API 12

Standard REST Calls in a REST API 12

Selecting Business Object Text Objects for a REST API 13

Changing the visibility of a selected text object 13

Deleting selected text objects from a REST API 14

Viewing an API Overview for a REST API 14

REST API deployment areas 14

Deploying a REST API 15

Downloading the OpenAPI Specification document....15
Undeploying a REST API....16
Suspending a REST API....16
Activating a REST API....16
REST API Simulator....17
Save to Host for a REST API....17
Saving to Host....17
Cancel Pending Changes for a REST API....18
Exporting REST APIs....18
Importing REST APIs....19

Chapter 2 Using Deployed Rest APIs....20

REST API Runtime Server....20
REST API requests....20
REST API request format....21
Caching....21
Licensing....21
Logging and Diagnostics....23
Logging Unexpected Errors....26
Infor OS API Gateway integration....27
Adding a target endpoint to the API Gateway....27
Authorized Apps....29
Creating a backend service authorized app....29

About this guide

This document provides an overview for developing and accessing REST APIs using the REST API Designer within the Integrator application. The document explains requirements for launching the REST API designer, specifies licensing requirements, and identifies the required security permissions.

This guide is intended for developers who create and deploy APIs.

Contacting Infor

If you have questions about Infor products, go to Infor Concierge at https://concierge.infor.com/ and create a support case.

The latest documentation is available from docs.infor.com or from the Infor Support Portal. To access documentation on the Infor Support Portal, select **Search > Browse Documentation**. We recommend that you check this portal periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Developing Rest APIs

This section explains creating and configuring REST APIs.

Accessing REST APIs

The REST API Designer can be launched from the REST APIs object, which has been added to the Integrator application card.

The REST API Designer is available only when using Net-Link. If you attempt to run the REST API object from Power-Link, this error message is displayed: REST APIs are only available in Net-Link. Additionally, you must have a valid non-expired REST API license installed in your IDF environment to access the REST APIs object.

REST APIs are subject to the same security requirements (“OBJECT”, “ADMIN”) as other Integrator components, including business objects and deployment profiles.

See the *Infor IDF REST API Runtime Server Installation and Administration Guide* for comprehensive information on installing, deploying, and running the runtime server.

Creating a REST API

Complete these steps to create a new REST API:

1 Select **Integrator > REST APIs**.

2 Select **Maintain > Create**.

3 Specify this information:

API title

Specify a title for the API.

API name

Specify a name for the API.

API version

Specify the version for the API.

The combination of the API name and version attributes must be unique across all REST APIs within the IDF environment. The name and version are included in the URL that end users call when making requests against this REST API. For example: `http://<HOST>:<PORT>/infor-idf/XA/<Environment>/rest-api/api/<API name>/<API version>/<API Request>)`.

4 Click **Create**.

If you skip **Preview before create**, then an empty REST API is created with nothing deployed. To add API Calls to deploy, see *Adding Business Objects to a REST API*.

REST API details

The **REST API details** page contains this information:

- General
- Notes
- Related information
  - Business Objects
  - API Overview

The **Business Objects** tab contains two sections: Selected and Available. The Available section shows a list of all business objects that can be selected for a REST API. The Available list filters the Integrator business objects list to show these objects that can be added on the main browser:

- Primary business objects
- Code File business objects
- Current revision or Date Effective revision business objects
- Text business objects

To select Business Objects for your REST API, see *Selecting business objects for a REST API*.

Viewing REST API details

Complete these steps to view details for a REST API:

1 Select **Integrator > REST APIs**.

2 Double-click the REST API record to view.

Selecting business objects for a REST API

When you select a business object, the key attributes for that object are automatically selected. To select other attributes, see *Selecting Business Object Attributes for a REST API*. Complete these steps to select Business Objects to be deployed on a REST API:

1. Select **Integrator > REST APIs**.
2. Double-click the REST API record to modify.
3. On the **Business Objects** tab, select one or more business objects from the **Available** list, then click the **Add** icon above the list.
4. Click **Continue**.

Removing selected business objects from a REST API

Complete these steps to delete specified business objects from a REST API:

1. Double-click the REST API record to modify.
2. Select one or more business objects from the **Selected** list on the **Business Object** tab.
3. Click the **Remove** icon on the Selected list card.

Selecting business object attributes for a REST API

When a business object attribute is selected, the **Attribute Usage** automatically aligns with the default attribute usage assigned to both primary and extension attributes belonging to the business object. For **Query Scope** attributes, the Attribute Usage is always set to **Display**, regardless of any default settings. Key attributes are automatically selected and cannot be removed.

Complete these steps to select Business Object Attributes to be deployed on a REST API:

1. Double-click the REST API record or select the **REST API details** action.
2. Select the business object to modify in the **Selected** list on the **Business Object** tab.
3. Double-click the record or select the **REST API Business Object details** button.
4. Click the **Attributes** tab.

   The **Available** list includes these types of attributes:

   - Non-hidden attributes from the business object
   - Non-hidden attributes from all installed Secondary extensions of the business object

- Non-hidden attributes from all installed related business objects that are included in the business object's Query Scope

5 Select one or more business object attributes in the **Available** list, then click the **Add** toolbar button.

When you add business object attributes, only attributes not already in the **Selected** list are added. Duplicate attributes cause an error or are ignored if selected in a group.

Changing the visibility of a selected business object attribute

When deploying a maintainable business object attribute in a REST API, you can deploy the attribute as display-only instead of maintainable.

Complete these steps to change the attribute usage of deployed business object attributes in a REST API:

1 Double-click the REST API record or select the **REST API details** action.

2 Select the business object to modify in the **Selected** list on the **Business Objects** tab.

3 Double-click the selected business object record or select the **REST API Business Object details** option.

4 Click the **Attributes** tab.

5 Select one or more REST API attributes in the **Selected** list.

6 Click the **Make displayable** icon on the Selected list card.

You can identify changes to selected business object attributes by choosing the **Overridden** subset (available in Grid Settings) or by verifying which REST API attributes have **Yes** in the **Overridden** column.

Resetting business object attribute usage

Complete these steps to return the usage value for an attribute in a REST API back to the original value:

1 Select one or more REST API attributes in the Selected list.

2 Click the **Reset to default** icon on the Selected list card.

Removing selected business object attributes from a REST API

Complete these steps to remove selected business object attributes from a REST API:

1. Double-click the REST API record or select the **REST API details** action.
2. Select the business object to modify on the **Business Objects** tab.
3. Double-click the record or select the **REST API Business Object details** action.
4. Click the **Attributes** tab.
5. Select one or more REST API attributes.
6. Click the **Remove** icon on the Selected list card.

Selecting business object REST Calls for a REST API

Business objects with header and detail relationships are objects such as Purchase Order, Purchase Order Item, and Purchase Order Release. When you call one of these objects, REST API calls, such as Display Purchase Order Details, automatically return the Purchase Order with its related Items and Releases in one response. This behavior applies to the create actions Display Object, Display Options, and Maintainable Object. For example, with a REST API Purchase Order create action, users can submit the full Purchase Order structure in their request.

Complete these steps to deploy business object REST Calls to a REST API:

1. Double-click the REST API record or select the **REST API details** action.
2. Select the Business Object to modify on the **Business Objects** tab.
3. Double-click the record or select the **REST API Business Object details** action.
4. Navigate to the **REST Calls** tab.

   The Available section shows every action that can be used as a REST Call for the specified business object. This list is generated by filtering the Integrator Business Objects Actions for that business object, according to specific criteria:

   - Display lists of Business Object records if permitted in the main browser
   - Display details action
   - Create action
   - Change action
   - Copy action
   - Delete action
   - Object Transaction actions

- Display Related List actions
- Display Related Object action

Note that actions show only on the **Available** list if they are supported by the Integrator Business Object.

5 Select a business object action from the **Available** list.

6 Click the **Add** icon on the Available list card.

7 Click **Yes** if you receive a message.

8 Specify this information:

API path

Specify an **API path**. This value must be unique across all REST Calls within the Business Object. A unique default value will be supplied.

API summary

Specify an **API summary**. A default value will be supplied.

Apply subset

Select an **Apply subset** value to apply an optional REST API subset when deploying a Display List or a Display Related action.

Apply sort

Select an **Apply sort** value to apply an optional REST API sort when deploying a Display List or Display Related action.

Apply template

Select an **Apply Template** value to apply an optional template when deploying a Create, Change, Copy, or Object Transaction action that supports templates.

9 Click **Create**.

The selected business object action is included in the Selected list as a REST Call.

If multiple business object actions are selected, then each action is added to the Selected list with the default values of the action without additional prompts.

You can use the same action to set up several REST API calls. For example, you can deploy the Purchase Orders list action with no subset and the default primary key sort to query all orders. You can then deploy with a specific subset and a different sorting method, like querying active orders sorted by order date in descending order.

Deleting selected business object REST Calls from a REST API

Complete these steps to delete, or remove, selected Business Object REST Calls from a REST API:

1. Double-click the REST API record or select the **REST API details** action.
2. Navigate to the **Business Objects** tab.
3. In the **Selected** list, select the business object to modify.
4. Double-click the record or select the **REST API Business Object details** action.
5. Navigate to the **REST Calls** tab.
6. Select one or more REST Calls in the Selected list.
7. Select the **Delete** icon on the Selected list card.

Changing Selected Business Object REST Calls in a REST API

Complete these steps to modify selected Business Object REST Calls from a REST API:

1. Double-click the REST API record or select the **REST API details** action.
2. Select the business object to modify in the Selected list on the **Business Objects** tab.
3. Double-click the record or select the **REST API Business Object details** action.
4. Select the REST Calls in the Selected list to modify on the **REST Calls** tab.
5. Double-click the record or select the **Change** action.
6. Update the maintainable fields.
7. Click **Update**.

Standard REST Calls in a REST API

A REST API automatically deploys several standard REST API calls that do not show in the standard list.

1. **/Sessions/create**: Log in to the REST API Runtime Server. This call returns a login token that must be passed on subsequent API Calls.
2. **/Sessions/delete**: Logout of the REST API Runtime Server.

3 **/Reasons/query**: Returns a list of available Reason codes and descriptions. This call is deployed if at least one business object has **Reason Tracking** turned on at the time the REST API is deployed.

4 **/openapi.json**: Returns the current version of the OpenAPI Specification document for the REST API.

Selecting Business Object Text Objects for a REST API

When a text object is selected, the Usage is configured automatically to correspond with the default settings for that Text Object. You can modify the usage of a selected Text Object.

Text objects are presented as variable-length string properties to users of the REST API. Their inclusion in REST API calls depends on the relevant Display Object and Display Option parameters. Additionally, text objects may be used within any maintenance-oriented REST API requests.

Complete these steps to select Business Object Text Objects to be deployed on a REST API:

1 Double-click the **REST API** record or select the **REST API details** action.

2 Navigate to the.

3 Select the business object to modify in the Selected list on the **Business Objects** tab.

4 Double-click the record or select the **REST API Business Object details** action.

5 Click the **Text Objects** tab.

6 Select one or more Text Objects in the Available list.

7 Click the **Add** toolbar button.

Selected text Objects are added to the Selected list. If a text object is already in the list, then an error message is shown. When you select multiple Text Objects, any duplicates are ignored.

Changing the visibility of a selected text object

When deploying a maintainable text object in a REST API, we recommend that you deploy the business object as display-only instead of as maintainable.

Complete these steps to change the usage of deployed text object in a REST API:

1 Double-click the REST API record or select the **REST API details** action.

2 Select the business object to modify in the Selected list on the **Business Objects** tab.

3 Click the **Text Objects** tab.

4 Select one or more REST API Text Objects in the Selected list.

5 Select the **Make displayable** icon on the Selected list card.

6 To reset the usage of a selected text object to the original value, select one or more REST API Text Objects in the selected list.

7 Select the **Reset to default** icon.

You can identify which selected text objects have been modified by choosing the **Modified** subset from the Selected list or by finding REST API Text Objects with **Yes** marked in the **Modified** column.

Deleting selected text objects from a REST API

Complete these steps to remove selected text objects from a REST API:

1 Double-click the REST API record or select the **REST API details** action.

2 Select the business object to modify in the Selected list on the **Business Objects** tab.

3 Double-click the record or select the **REST API Business Object details** action.

4 Select one or more REST API Text Objects in the Selected list on the **Attributes** tab.

5 Click the **Delete** icon in the Selected list.

Viewing an API Overview for a REST API

Complete these steps to display an Overview of a REST API:

1 Double-click the REST API record or select the **REST API details** action.

2 Click the **API Overview** tab.

The API Overview shows all the business objects that you have selected. Each selected business object includes lists of the REST calls, attributes, and text objects associated with the object.

REST API deployment areas

On the REST APIs List View, these two separate deployment areas are displayed: **Release** and **Test**. Which one is used depends on the current state of the REST API. If your REST API has Pending Changes, Deploy deploys to **Test**. If your REST API has no Pending Changes, **Deploy** deploys to **Release**.

Deploying to Test gives you a separate area to test your REST API Pending Changes before deciding to Deploy to Release. Typically, when creating a REST API, you make changes, then deploy to Test to verify the changes are correct. This action can be repeated multiple times. After you are satisfied with the changes, you use the **Save to Host** action on the REST API to make the changes permanent. Then you deploy the REST API using the Release deployment. Deploying to Release undeploys the REST API from the Test area and deploys the REST API to the Release area.

The REST API list contains **Deployed (release)** and **Deployed (test)** properties that show you where the REST API is currently deployed. Note that a REST API can be deployed to **Release** and **Test** at the same time. If the property is blank, then this REST API has not been deployed to that area.

The REST API keeps track of whether all current changes have been deployed. The **Deployed (test) and Deployed (release)** properties show an orange background if the properties are out-of-date:

- **Deployed (test)** shows an orange background if you have made new Pending Changes since the last time you deployed to **Test**
- **Deployed (release)** shows an orange background if you have run a **Save to Host** action, without choosing the Deploying REST APIs option, since the last time you deployed to **Release**.

Deploying a REST API creates an OpenAPI Specification (Swagger) document that describes the REST API. This document includes the URL to access the REST API. By default, this URL contains the IBM i name plus the IDF Kernel Gateway port. In cases where you have created a public (external) REST API and do not want to directly expose this URL, you can specify an API server base URL (override) value that is published instead in the Swagger document.

Deploying a REST API

Deploying a REST API generates the information needed so that end users can make REST API calls for your REST API.

Complete these steps to deploy a REST API.

1 Select the REST API record to deploy.

2 Select **Maintain > Deploy**.

Downloading the OpenAPI Specification document

Complete these steps to download the OpenAPI Specification document.

3 Double-click the REST API record or select the **REST API details** action.

4 Select **Display > Download OpenAPI Specification**

Undeploying a REST API

Undeploying a REST API permanently removes the information needed for this REST API so that end users can no longer make REST API calls for your REST API.

If you only want to temporarily undeploy a REST API, you can use the Suspend action instead.

Complete these steps to undeploy a REST API.

1 Select the REST API record to deploy.

2 Select **Maintain > Undeploy**.

Suspending a REST API

Suspending a REST API temporarily removes the information needed for this REST API so that end users can no longer make REST API calls for your REST API.

Complete these steps to suspend a REST API.

1 Select the REST API record to suspend.

2 Select **Maintain > Suspend**.

The **Deployed (test) and Deployed (release)** properties show a red background if they have been suspended. After a REST API has been suspended, you can reactivate the REST API using the **Activate** action. If you want to permanently undeploy a REST API, you can use the **Undeploy** action.

Activating a REST API

Activating a suspended REST API restores the information needed for this REST API so that end users can again make REST API calls for your REST API.

Complete these steps to activate a suspended REST API.

1 Select the REST API to activate.

2 Select **Maintain > Activate**.

REST API Simulator

After you have deployed a REST API, you can use the Rest API with any standard method of calling them. We provide an internal Swagger editor called the REST API Simulator that you can use to manually exercise the deployed REST API.

Complete these steps to run the REST API Simulator on a REST API.

1 Select the REST API record to run the simulator on.

2 Select **Navigation > REST API Simulator**.

Refer to https://swagger.io/tools/swagger-editor/ for information on the Swagger Editor.

Save to Host for a REST API

Rest APIs have a similar structure as other Integrator objects, such as Business Object and Deployment Profile. When you make changes to a REST API, they are marked as **Pending** instead of being immediately applied. These changes are visible only to you.

The **Save to Host** dialog box contains a Subset list that indicates which REST API records are to be processed. The value for the Subset property is determined by these conditions:

- If the user has selected multiple records in the list, then (selected records) are to be set
- If the user has selected a single record in the list, then (selected record) are to be set
- If no records are selected, then (pending deletes only) are to be set

Note that any Pending Delete records are always processed during a **Save to Host** operation regardless of whether the user chooses (selected records) or (pending deletes only).

When you click **Save to Host** for REST APIs, these actions are taken:

- Permanently apply all Pending Change records and set them to Modified
- Permanently remove any Pending Delete records

Saving to Host

Complete these steps to run the Save to Host action.

1 Select one or more REST APIs. This step is optional.

2 Select the **File > Save to Host** menu action.

If you select more than one REST API, then the optional **Deploy REST APIs** action is available. This option automatically deploys the REST API to release when the **Save to Host** task is complete.

See *Deploying a REST API*.

Cancel Pending Changes for a REST API

As described in the Save to Host for a REST API task, any changes you make to a REST API shows as Pending Change and are only visible to you.

When you reset the selected REST APIs, they return to the state they were in after the last Save to Host action. If an API was deleted but not yet saved to host (pending delete), then that deletion will be undone. If you cancel pending changes on a REST API that has never been saved to host, then it will be permanently removed.

Use this task to discard all pending changes and restore the API to its state after the last save to host.

1 Optionally, select one or more REST APIs.

2 Select **File > Cancel Pending Changes**.

The **Cancel Pending Changes** dialog box contains a **Subset** attribute that indicates that REST API records are to be processed. These conditions determine the value for the Subset property:

- If the user selects multiple records in the list, **(selected records)** is set
- If the user selects a single record in the list, **(selected record)** is set
- If no records are selected **(pending deletes only)** is set

3 Click **Continue**.

Exporting REST APIs

The selected REST APIs, along with any REST API sorts, subsets, and templates that they reference, are packaged into an export file upon export. To import these REST APIs in another environment, access the target environment and run a REST API import action. See *Importing REST APIs*.

Use this task to export REST APIs for use in another environment.

1 Select one or more REST APIs.

2 Select **File** > **Export REST APIs** menu action.

3 Specify this information:

Subset

Select a subset to indicate which REST API records is to be processed. The value for the Subset property is determined by these conditions:

- If the user has selected multiple records in the list, then (selected records) will be set.
- If the user has selected a single record in the list, then (selected record) will be set.

File name

Specify a file name.

4 Click **Export**.

Importing REST APIs

Note: if the REST API being imported depends on new Integrator Business Object changes in the environment you exported from, you must import those Integrator Business Object changes before you import the REST API.

During the import, if any REST APIs in the import file already exist in the target environment, then they are verified to ensure these conditions exist:

- They do not have any pending changes
- They are not currently checked out to another user

In addition, all REST APIs in the import file verify this information:

- The API name and API version combination is not already in use by a different REST API.
- Any Integrator Business Objects that are referenced by the REST API exist and are installed in the target environment.

If any of the verifications fail, then the import is aborted. No changes are applied, and you are presented with a list of errors.

If all the verifications pass, then the REST APIs in the import file are loaded into the target environment. All import changes are marked as **Pending change**. After the import, you can use the **Save to Host** action to make the changes permanent or use the **Cancel Pending Changes** action to remove them.

Complete these steps to import REST APIs from another environment:

1 Select one or more REST APIs.

2 Select the **File > Import REST APIs** menu action.

3 In the **Import REST APIs** dialog box, specify the **File Name** of the export file to be imported.

4 Click **Import**.

Chapter 2 Using Deployed Rest APIs

This section explains how to use deployed REST APIs.

REST API Runtime Server

The REST API Runtime Server is a dedicated Java server that processes REST API requests for REST APIs that are built and deployed by the REST API object.

Each IDF environment has its own REST API Runtime Server to process all requests for all deployed REST APIs for that environment.

REST API requests

A normal API client flow using the REST API Runtime server includes these steps:

1. A log in (Sessions/create) API request is initiated in the API client and, upon successful completion, a loginToken is received.
2. One or more REST API requests are initiated in the API client, including the loginToken value in the X-Infor-Login-Token header.
3. To end a session, a log out (Sessions/delete) API request is sent to the API client, including the loginToken in the X-Infor-Login-Token header. If the client does not log out manually, then inactive sessions are reclaimed by the REST API Runtime server. A session is considered inactive if the REST API Runtime Server does not receive a request within the **`maxIdleMinutes`** time that was specified on the log in request.

You can bypass the Sessions/create call and make REST API requests directly from Infor ION using a JWT Token for authentication by the ION API Gateway. See *Infor IO API Gateway Integration*.

Each REST API session (loginToken) can accept multiple concurrent requests; however, only one request can be processed at a time. Any additional requests received during an active process are queued in order of arrival. To enable parallel processing by the REST API Runtime Server, multiple REST API sessions can be established, with requests distributed across these sessions.

REST API request format

The REST APIs are generated according to the Open API Specification version 3.0.3 standard. String, integer, and boolean properties use the normal standard, which is how other IDF data types are handled:

1. Fixed-point decimals are accepted and returned as Strings (type: string format: decimal), for example, “54.30”.
2. Dates/Times/Timestamps are accepted/returned per the RFC3339 standard.
   a. Dates are accepted or returned as strings (type: string format: date). For example, 2025-02-23. IDF empty dates are passed as null.
   b. Times are accepted or returned as strings (type: string format: time-local). For example, 12:14:15. Fractional seconds and time zone information are ignored as they are not supported by IDF.
   c. Timestamps are accepted or returned as strings (type: string: date-time-local). For example, 2025-02-23T12:14:15. Fractional seconds and time zone information are ignored as they are not supported by IDF.

   Null-capable fields in IDF are accepted and returned as **null**.

Caching

The REST API Runtime Server has a smart caching feature. IFS files are cached for performance in the REST API Runtime Server. Every minute, any of these actions are verified to see if they have been taken for the requested REST API:

- Deployed
- Undeployed
- Suspended
- Activated

If one of the above changes is detected by the server, then the REST API files are removed from the REST API Runtime Server’s cache, then reloaded. From the API Designer perspective, running any of the above actions takes effect in the REST API Runtime Server within one minute.

Licensing

Infor ships a general Product License (**5I33IDF V0R0 5184** for a system license or **5I34IDF V0R0 5184** for a partition license). This license is verified on REST API Runtime Server startup and these conditions are verified:

- License must exist on partition
- License must be valid and not expired
- Usage limit must be > 0

If any of the verifications are unsuccessful, then the REST API Runtime Server fails to start, and all errors are documented in the **`server.log`** process log file. See the *Infor IDF Kernel Installation and Administration Guide*.

Infor also ships a Concurrent License (**5C33IDF V0R0 5184** for a system license or **5C34IDF V0R0 5184** for a partition license). This license is verified on REST API Runtime Server startup and these conditions are verified:

- License must exist on partition
- License must be valid and not expired

If one or more of the verifications do not pass, then the REST API Runtime Server fails to start, and any errors are shown in the process log file.

The REST API Runtime Server uses a Peak Usage Per Hour licensing model, limiting total REST API requests per hour across all servers in a partition (partition license) or across partitions (system license). Customers are responsible for distributing licenses among their deployed servers.

When starting the REST API Runtime Server, use the licenses property to set the maximum requests per hour for the server. This number of licenses is allocated by the server. If you do not have enough available licenses, then the server cannot start and errors are displayed in the log file. The total license count is shared across all servers in a partition or system. The sum of the number of licenses for all running servers must not exceed the concurrent license limit, or some servers may fail to start.

After the REST API Runtime Server is running, the incoming requests are limited to x number of licenses per hour. Any excess requests receive a 429 (Too Many Requests) error.

```
[{“messageType”: “error”,

“detail”: “Please wait and try again at the top of the hour.”’,

“message”: “You have exceeded 6000 requests this hour.”

}]
```

A 429 response includes a Retry-After header that tells the client how many seconds to wait before trying again; for example, Retry-After: 720.

The request count resets automatically at the beginning of each hour. When you start a REST API Runtime Server, the number of requests permitted is pro-rated for the first partial hour. For example, if you request 1000 licenses and the server starts at 10:45 AM, then you have 250 requests available until 11:00 AM. After that, you will get the full 1000 requests for the next hour.

On REST API Runtime Server shutdown, all allocated licenses are released and are available to be used by other server instances.

Logging and Diagnostics

The REST API Runtime Server has several available logging levels:

- OFF
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- ALL

The REST API Runtime Server logging level is set to INFO when it starts. Setting a logging level logs information for that level and levels above it. For example, setting the logging level to INFO logs all INFO, WARN, and ERROR messages.

If you need to change the logging level for advanced error diagnostics, go to Infor Concierge at https://concierge.infor.com/ and create a support case.

API clients can enable troubleshooting by adding an X-Infor-Diagnostics-Token header to a REST API call. When detected, the REST API Runtime Server logs detailed diagnostic data for that request, using the token value to identify the request in the log file. For example, sending Buyers/query with X-Infor-Diagnostics-Token: myToken generates this log entry specific to that call:

```
[DEBUG][2023-12-11T08:44:01.665784900][idf-1-actors-11][D:myToken] [POST
/Buyers/query] Forwarding request to SessionActor

[DEBUG][2023-12-11T08:44:01.665784900][idf-1-actors-11][D:myToken] [POST
/Buyers/query] Start

[DEBUG][2023-12-11T08:44:01.665784900][idf-1-actors-11][D:myToken] [POST
/Buyers/query]  Transforming Request: Buyers/POST-query

…

[DEBUG][2023-12-11T08:44:01.774897200][idf-1-actors-2][D:myToken] [POST
/Buyers/query]  Transforming Response: Buyers/POST-query

…

[DEBUG][2023-12-11T08:44:01.787905400][idf-1-actors-2][D:myToken] [POST
/Buyers/query] End - Elapsed Time: 122ms
```

Important things to note in this call log:

- Each line starts with the [D:myToken] prefix to identify relevant log entries for the request.
- Each line incorporates the [POST /Buyers/query] prefix string to specify clearly the API call being referenced.
- The End call provides the total elapsed time (End – Elapsed Time: 122ms). The End call measures the time from when the REST API Runtime Server begins processing the request to when the server finishes and sends a response.

- Each line includes a system timestamp [2023-12-11T0*:44:01.665784900] for internal diagnostics, allowing for full analysis of time spent on each step in the request. For example, queue time and processing time can be calculated from the system timestamps.
- Diagnostics are only written to the log if the logging level is set to DEBUG, TRACE, or ALL.

If you include the X-Infor-Diagnostics-Token header in the login (Sessions/create), then all REST API calls for that session are logged.

In addition, long-running requests are automatically logged regardless of whether an X-Infor-Diagnostics-Token header is included in the request:

```
[WARN ][2025-08-20T12:24:48.352][torq-affinity-thread-4][PERF@28944332] [POST
/SMUTest/1.1.0/Buyers/query] Start

…

[WARN ][2025-08-20T12:24:48.352][torq-affinity-thread-4][PERF@28944332] [POST
/SMUTest/1.1.0/Buyers/query]  Transforming Request: Buyers/POST-query

…

[WARN ][2025-08-20T12:24:48.517][torq-affinity-thread-4][PERF@28944332] [POST
/SMUTest/1.1.0/Buyers/query]  Transforming Response: Buyers/POST-query

…

[WARN ][2025-08-20T12:25:53.079][torq-affinity-thread-4][PERF@28944332] [POST
/SMUTest/1.1.0/Buyers/query] End - Elapsed Time: 64727ms
```

Important things to note in this log:

- These log entries are written with type WARN
- The logging id is PERF@<token>
- The long-running requests are not written to the log if the logging level is set to OFF or ERROR.

In addition to request logging, we periodically write out overall REST API Runtime Server statistics to the process log:

```
[INFO ][2025-10-23T10:19:36.278][t1][restApiActor.STATS]
================================================================================
==========

[INFO ][2025-10-23T10:19:36.279][ t1][restApiActor.STATS] Memory (Used: 27 MB
Free: 40 MB Allocated: 68/8116 MB)

[INFO ][2025-10-23T10:19:36.279][ t1][restApiActor.STATS]

[INFO ][2025-10-23T10:19:36.279][ t1][restApiActor.STATS] Total Requests
Processed: 15

[INFO ][2025-10-23T10:19:36.279][t1][restApiActor.STATS]  Queued Requests: 0

[INFO ][2025-10-23T10:19:36.282][t1][restApiActor.STATS]  Logins Processed: 1
(Avg: 1395ms Min: 1395ms Max: 1395ms)
```

```
[INFO ][2025-10-23T10:19:36.282][t1][restApiActor.STATS]  ION SSO Logins
Processed: 0

[INFO ][2025-10-23T10:19:36.283][t1][restApiActor.STATS]  OpenApi Specs
Processed: 0

[INFO ][2025-10-23T10:19:36.283][t1][restApiActor.STATS]

[INFO ][2025-10-23T10:19:36.283][t1][restApiActor.STATS] Sessions (1/0):

[INFO ][2025-10-23T10:19:36.283][t1][restApiActor.STATS] Session: 9922eb5b-
4b93-41af-8011-7143606ad3f7 (0d 0h 4m 52s)

[INFO ][2025-10-23T10:19:36.283][t1][restApiActor.STATS]  Queued Requests: 0

[INFO ][2025-10-23T10:19:36.284][t1][restApiActor.STATS]  Requests Processed:
7

[INFO ][2025-10-23T10:19:36.284][t1][restApiActor.STATS]   Avg Response Time:
456ms

[INFO ][2025-10-23T10:19:36.284][t1][restApiActor.STATS]   Min Response Time:
202ms (POST /SMUTest/1.1.0/Buyers/query)

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS]   Max Response Time:
1348ms (POST /SMUTest/1.1.0/Buyers/query)

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS]

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS] Total Requests (this
period): 750

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS] Max Requests (this
period): 1000

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS] Available Requests
(this period): 250

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS]

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS] ResourceActor Queued
Requests: 0

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS] IonSSOActor Queued
Requests: 0

[INFO ][2025-10-23T10:19:36.285][t1][restApiActor.STATS]
```

Important things to note in this statistics log:

- These log entries are written with type INFO
- The logging id is STATS
- Statistics are not written to the log if the logging level is set to OFF, ERROR, or WARN.

Logging Unexpected Errors

Expected Error messages are returned by a standard 400 (Bad Request) HTTP response code in this format:

```
{
"messageType": "error",
"message": "Invalid Login Token.",
"detail": "The Login Token you provided is either incorrect or has expired."
}
```

Unexpected Error messages are returned by a standard 500 (Internal Server Error) HTTP response code in this format:

```
{
"traceID": "TRACE-1",
"message": "Unexpected error ResourceActor.onRetrieveAllFilesFor
ApiCallRequest"
}
```

For unexpected errors, a traceID value is also returned. Looking at the REST API Runtime Server log file, you will find a matching entry with the same traceID [TRACE-1] that contains all the relevant support information about the error:

```
[ERROR][2025-10-10T10:29:45.714][torq-affinity-thread-0][resourceActor]
[TRACE-1] Unexpected
error ResourceActor.onRetrieveAllFilesForApiCallRequest:
java.lang.NullPointerException: Cannot invoke "String.toString()" because
"resourceName" is null
at com.infor.idf.server.restapi@1.1.1-
SNAPSHOT/com.infor.idf.server.restapi.transform.ResourceActor.onRetrieveAllFi
lesForApiCallRequest(ResourceActor.java:151)
at com.infor.idf.server.restapi@1.1.1-
SNAPSHOT/com.infor.idf.server.restapi.transform.ResourceActor.onRequest(Resou
rceActor.java:126)
at com.infor.idf.kernel.server@1.1-
SNAPSHOT/com.infor.idf.kernel.server.BasicActor.onMessage(BasicActor.java:34)

at org.torqlang.local@1.0-
rc1/org.torqlang.local.AbstractActor$Dispatcher.run(AbstractActor.java:231)
at org.torqlang.local@1.0-
rc1/org.torqlang.local.AffinityThreadExecutor$Worker.run(AffinityThreadExecut
or.java:55)
```

```
at java.base/java.lang.Thread.run(Thread.java:840)
```

Note that detail is optional depending on the specific error.

Infor OS API Gateway integration

Infor OS API Gateway is a robust platform designed to manage and optimize API interactions within an organization. The API Gateway provides a common base URL path and a unified authentication mechanism, simplifying access and enhancing security for API consumers.

By centralizing API management, API Gateway facilitates seamless integration across systems. The API Gateway enables administrators to enhance functionality and ensure consistent communication among diverse applications.

After REST APIs are deployed, you can create them as Target API servers for which API Gateway is acting as a proxy. Rather than clients directly talking to these servers, client applications talk to the gateway, and the gateway talks to those target servers and adds value while doing so.

Note: Target API servers should be reachable by API Gateway. If these APIs are running on a local network, then you can connect to them by Enterprise Connector.

Adding a target endpoint to the API Gateway

1. Log in to Infor OS and select **API Gateway**.
2. Select **Available APIs** in API Gateway. The **Available APIs** page shows all available API suites that the tenant is authorized to use.
3. Click **Add New API Suite**.
4. Specify this information:

   Application Name

   Specify a name for the application.

   Suite Name

   Specify a name for the suite.

   Description

   Specify a description for the API Suite.

   API Context

   Specify the API context information.
5. To select a suite icon, click **Chose** icon, then complete the setup:

   a Select an icon.

b Select an icon color.

c Click **Save**.

6 Click **Add Endpoint**, then specify this information:

Target Endpoint URL

Specify the URL. The Target Endpoint URL should match the server URL in the Open API specification for the REST API deployed.

Target Endpoint Description

Specify the description for the target endpoint.

Authentication Type

Select **JWT Target Authentication**.

JWT Header

Specify the JWT Header as **`{ "alg": "RS256","kid": "KeyID"}`**.

JWT Payload

Specify this information for the JWT payload:

```
{
"Tenant": "$(context.user.Tenant}",
"Identity2": "$(context.user.Identity2},
"iss": https://mingle-sso.inforcloudsuite.com:4443,
"scope":"signature impersonation",
"MaxIdleMinutes":10
}
```

`Tenant` is always set to `${context.user.Tenant},` and Identity2 to `$(context.user.Identity2}`.

The value for **`iss`** is the Base URL for calling authorization swerver for this tenant/environment.

`Scope` is always set to signature impersonation.

`MaxIdleMinutes` refers to the period of inactivity permitted before an IDF login session expires.

Expires in (seconds)

Optionally, specify the token expiration in seconds.

Key ID

Click **Generate** icon to generate the new Key ID. The Key ID (kid) in the JWT header must correspond to the generated Key ID. Click the **Download** icon to obtain the JWT Verifier file.

Algorithm

Specify the algorithm from RS256, RS384, RS512. The algorithm specified in the JWT header must correspond to the chosen algorithm.

7 Click the **Save** button in the upper left corner.

Authorized Apps

All Infor OS portal users who are also IDF users are required to have their **ERP Person ID** configured. The ERP Person ID serves as the identifier for the ERP Application User. The API Gateway includes the portal user's unique identifier as the Identity2 claim in the JWT payload. The REST API server uses this claim to call an IFS API and retrieve the ERP Person ID. Authentication for the IFS API call is handled by a bearer token from a registered Authorized App.

Creating a backend service authorized app

Create a backend service authorized application to enable secure machine-to-machine authentication using the client credentials grant flow. This app provides the client ID and client secret required to obtain access tokens from API Gateway.

1 Navigate to **OS > API Gateway > Authorized Apps**.

2 Click **Add** to create a new authorized app.

3 Specify this information:

Name

Specify a unique name for the authorized app. This field is required.

Type

Select **Backend Service**. This field is required.

Description

Specify a brief description of the purpose of the application. This field is required.

Grant Type

Select **Client Credentials**. This field is required.

4 Click **Save**.

5 Click **Download Credentials**.

6 Enable the **Create Service Account** option.

7 Select the name of a user to associate with the service account. The **User Admin** role must be assigned to this user.

8 Click **Download**.

See the *Infor IDF REST API Runtime Server Installation and Administration Guide* for instructions on uploading the downloaded JWT Verifier and IONAPI file.