



QAD Adaptive Applications

# Technical Reference Outbound REST Integration Configuration

70-3509-2024.1

QAD QMS Applications version 2024.1

September 2024

# Copyright

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

This document contains trademarks owned by QAD Inc. and other companies.

Copyright © 2024 by QAD Inc.

QAD Inc.

100 Innovation Place

Santa Barbara, CA 93108

Phone: +1 (805) 566-6100

<http://www.qad.com>

---

	<b>1</b>
<b>Copyright</b> .....	<b>2</b>
<b>Introduction</b> .....	<b>6</b>
About This Guide .....	6
<b>Prerequisites for Integrations</b> .....	<b>6</b>
User Roles .....	6
Access and Prerequisite Information Needed .....	6
Web Services .....	6
Call Service (Outbound/Push) .....	6
Create Service (Inbound) .....	6
REST APIs .....	7
<b>Capabilities of QAD EQMS REST Integration Services</b> .....	<b>7</b>
Compatibility Constraints .....	7
<b>Differences Between REST and SOAP</b> .....	<b>7</b>
<b>Limitations of QAD EQMS REST Integration Services</b> .....	<b>8</b>
<b>Instructions on Creating a Sample REST Service</b> .....	<b>10</b>
Creating a New REST Service .....	10
Step 1 .....	10
Step 2 .....	10
Step 3 .....	11
Step 4 .....	12
Step 5 .....	13
Step 6 .....	14
Step 7 .....	15

# Outbound REST Integration Configuration Change Summary

The following table summarizes significant differences between this document and previous versions.

Date/Version	Description	Reference	Changed By
SEPT 2022/v2022.1	Initial upload	--	NQC
MAR 2023/v2023	Updated versioning; Modified Prerequisites for Integrations; Modified Capabilities of QAD EQMS REST Integration Services; Modified Step 1; Modified Step 4; Modified Step 5	p. 6, p. 7, p. 10, p. 12, p.13	NQC
SEPT 2023/v2023	Removed obsolete information from Prerequisites for Integrations, as well as Capabilities of QAD EQMS REST Integration Services; Modified image and URL in Step 2	p. 6, p. 7, p. 10	NQC
MAR 2024/v2024	Updated versioning	--	NQC
SEPT 2024/v2024.1	Updated versioning	--	NQC

Chapter 1

# Introduction

*Introduction...6*

*Prerequisites for Integrations...6*

*Capabilities of QAD EQMS REST Integration Services...7*

*Differences Between REST and SOAP...7*

*Limitations of QAD EQMS REST Integration Services...8*

## Introduction

This document is meant to serve as a guide for setting up and configuring outbound REST capabilities in QAD EQMS 2024.1. In this guide, you will see an example for creating a Google Calendar event using Outbound REST in QAD EQMS.

### About This Guide

This user guide focuses on:

- Prerequisites for setting up Outbound REST integrations
- Limitations of REST integration services in QAD EQMS
- Integration capabilities for QAD EQMS and REST Integrations
- Difference between REST and SOAP and determining which to use
- Creating a sample REST event in QAD EQMS using Google Calendar Web API

## Prerequisites for Integrations

### User Roles

Users should be capable of installing QAD EQMS and following a proper SDLC process. Users should have an intermediate knowledge of EQMS Integration Configuration, IIS, and Microsoft SQL Server.

### Access and Prerequisite Information Needed

Users must know both the third party inbound REST Service URLs, along with the authentication type and credentials needed to log into those applications.

### Web Services

Web Services are designed to move data to and from QAD EQMS on a push/pull basis.

#### Call Service (Outbound/Push)

A call service can be configured to call a REST service from an external service or application.

Mappings can be established between the service's input and output parameters to and from the QAD process making the call.

Call services are executed (called) on save for the process for which they are configured.

#### Create Service (Inbound)

A create service can be configured to create QAD REST web services that can be called from an external service or application.

Methods can be configured for insert, update, insert or update, remove, and select actions. Input parameters can be defined and mappings can be configured.

## REST APIs

Every QAD business object (or process, as they are referred to within QAD) has exposed REST APIs.

For each API, the following request types are exposed:

- Search Results (a collection of records)
  - Get
- Details (a specific record)
  - Get
  - Put (update existing)
  - Post (add new)
  - Delete

Each request requires an authentication token, which can be retrieved through a call to the QAD EQMS authentication service. Once obtained, the token (a JWT token) is stored on the client for the length of the session and passed with any subsequent REST service requests. All user permissions are applied based on the authentication token.

Error codes are sent, but detailed errors are not sent for security purposes.

Despite the fact that the application uses REST APIs for its own architecture, these APIs are not optimized for use other than by the QAD EQMS application.

## Capabilities of QAD EQMS REST Integration Services

- All processes in QAD EQMS will automatically generate an inbound API that can be called to insert, update, or remove data from other applications.
- Outbound (aka "call") REST services can be created by user interface similar to normal call SOAP service.
- Outbound REST service supports either Basic or OAuth 2.0 authentication.
- Outbound REST service supports the following methods: GET, POST, PUT, DELETE, PATCH.

## Compatibility Constraints

The outbound REST service currently only supports the JSON (JavaScript Object Notation) format.

## Differences Between REST and SOAP

- SOAP is language, platform, and transport independent (REST requires use of HTTP, SOAP can work over HTTP, SMTP, and other protocols).
- SOAP is more standardized and extensible due to its longevity.
- REST is more efficient and associated with a smaller learning curve (SOAP uses XML for all messages, REST can use smaller message formats like JSON, CSV, and RSS).
- A helpful guide: <https://smartbear.com/blog/test-and-monitor/understanding-soap-and-rest-basics/>

## **Limitations of QAD EQMS REST Integration Services**

Please note the following limitation of QAD EQMS REST Integration Services:

Outbound REST Service in QAD EQMS currently only supports JSON format (JavaScript Object Notation).

## Chapter 2

# Creating a Sample REST Service

*Instructions on Creating a Sample REST Service...10*

*Step 1...10*

*Step 2...10*

*Step 3...11*

*Step 4...12*

*Step 5...13*

*Step 6...14*

*Step 7...15*

# Instructions on Creating a Sample REST Service

## Creating a New REST Service

REST integration has many variables that will change based on the third party you are integrating with. Instead of going over every variable, this guide will demonstrate REST integration using the Google Calendar API. Please note that the focus here is on the QAD EQMS side of the configuration and is not meant to demonstrate every detail for setting up integration with Google Calendar.

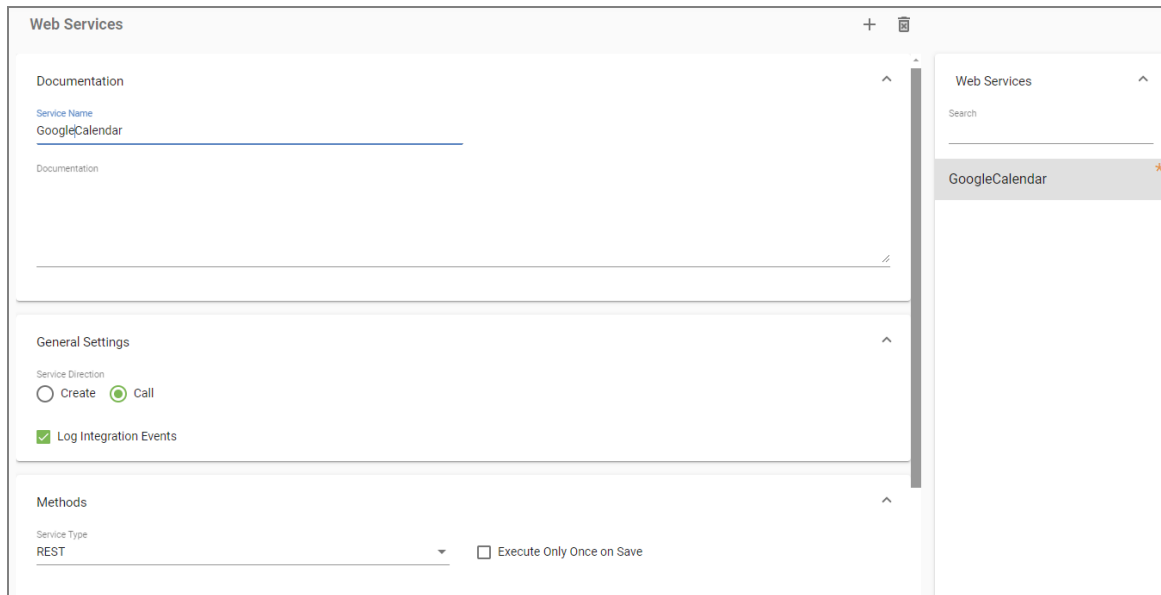
### Step 1

Configuring REST Service in QAD EQMS

1. From your chosen process in AppExtender, navigate to the Integration tab.
2. Select the Web Service tab and select the + button to create a new service.
3. Give your service a name and modify Service Direction to Call. In the example below, the service is named "GoogleCalendar".
4. Select REST under Service Type.

Below is a screen shot from the Web Services section. A new web service has been created.

**Fig. 1: Web Services screen**



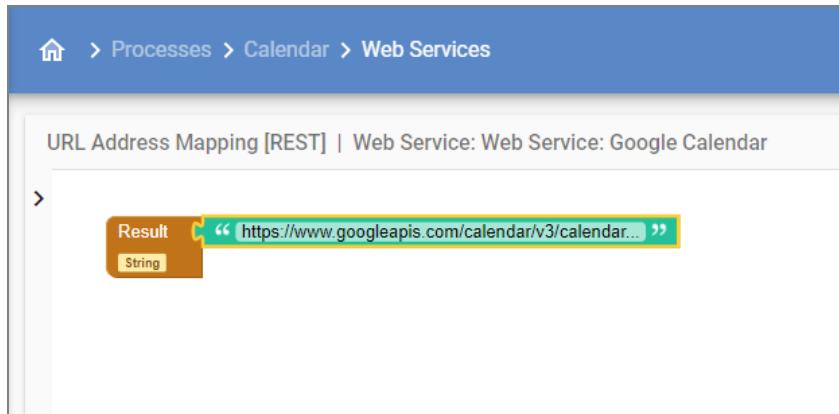
You are now ready to configure your new REST outbound service.

### Step 2

1. Under REST Settings, navigate to the General tab.
2. Under the URL section, click Edit Expression.
3. Your expression should consist of a text constant populated with the URL to the third party with which you are integrating. The third party should provide this information.

4. In this example, the Google Calendar API is being used. The URL is: “https://www.googleapis.com/calendar/v3/calendars/[calendarId]/events”.

**Fig. 2: Web Services screen, URL Edit Expression**



### Step 3

1. Under HTTP Method, select POST.

**Note:** This is dictated by the documentation or API from the third party. In this example, Google has specified to use "POST".

2. Navigate to the Authorization tab. The values on this tab must be populated based on the requirements from the third party. Continuing with the Google Calendar example, note that in the screen shot below, OAuth2 authentication is configured.
3. Once your values are populated on the Authorization tab, choose the Get Token button to obtain the access token and refresh token. If authentication is successful, then these values should populate in their respective fields.

**Fig. 3: Web Services screen, Authorization tab**

General	Authorization	Parameters	Headers	Request	Response	Json Types
Authorization Type						
OAuth2						
OAuth 2 Flow						
Authorization Code						
Authorization URL				Access Token URL		
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent				https://www.googleapis.com/oauth2/v4/token		
Client				Client Secret		
517214164012-3cvns59tfqhlid15f9kbfmk3askr8in.apps.googleusercontent.com				RnY3V5cApeExpa02BFQpzlGi		
Scope						
https://www.googleapis.com/auth/calendar						
<input type="checkbox"/> Use current user credentials						
Access Token				Access Token Expiry Date		
RnY3V5cApeExpa02BFQpzlGiRnY3V5cApeExpa02BFQpzlGiRnY3V5cApeExpa0:				9/30/2022 12:00 AM		
Refresh Token						
cApeExpa025cApeExpRnY3Va02BFQpzlGiRnY5cApeExp3V5cApeExpa02BFQpzlGi						

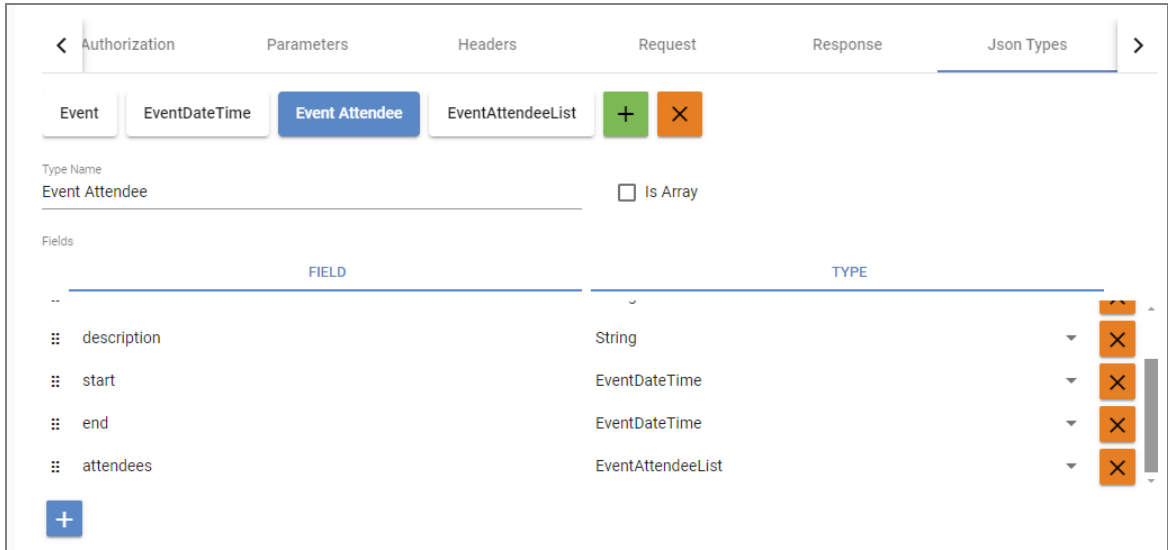
#### Step 4

In this document's example, no parameters or headers are required, so the next step is to navigate to the Request Body tab. Note that parameters and headers would be used in the "GET" REST type.

1. Under Content Type, select JSON.
2. Select Content. Under Type, select Event.
3. Navigate to the JSON Types tab.

**Note:** The type definitions will come from the third party. In this document's example, working with the Google Calendar APIs, a type has been created named Event Attendee.

**Fig. 4: Web Services screen, Json Types tab**

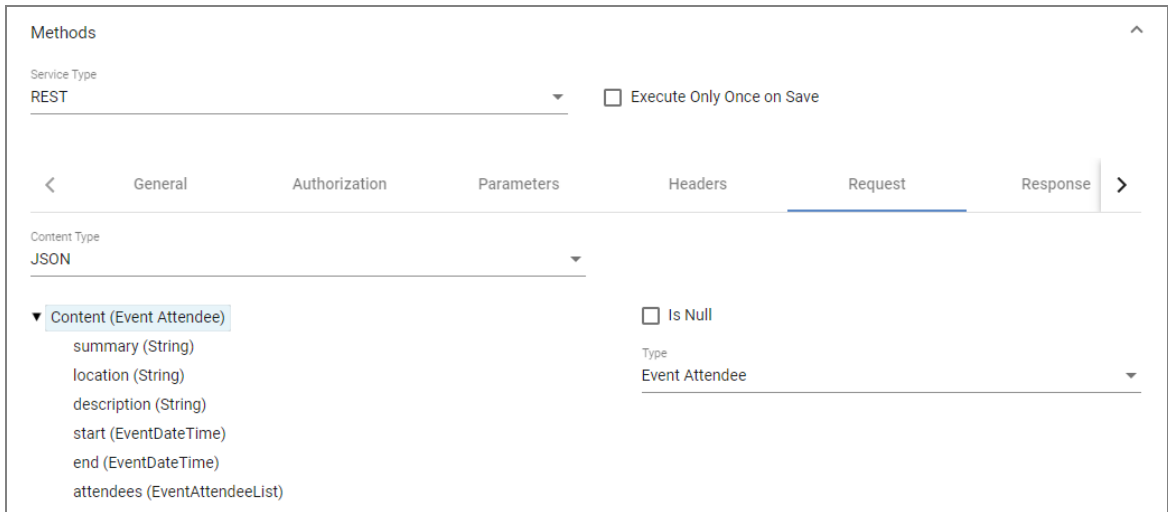


**Step 5**

1. Navigate to the Request tab.

Remember, the content is going to depend on the third party. An example is below.

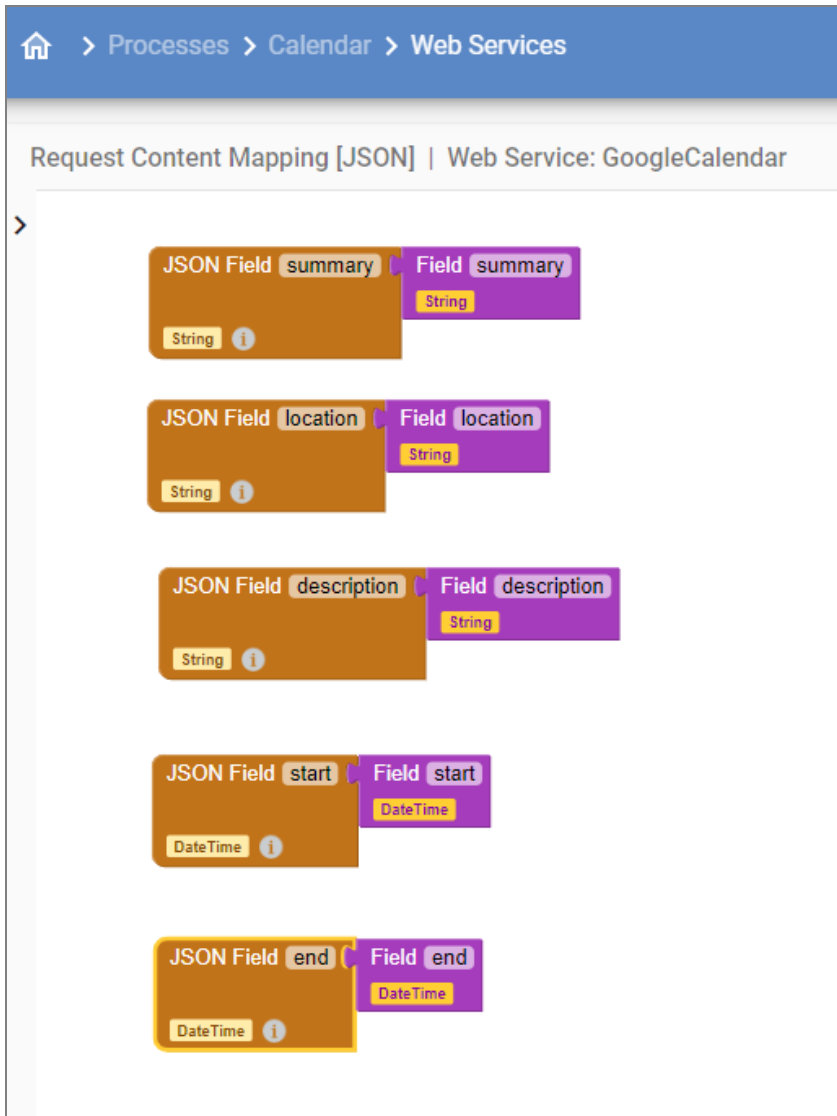
**Fig. 5: Web Services screen, Request tab**



2. Under Content Mapping, click Edit Expression.
3. Map the appropriate fields from the JSON field to the QAD EQMS field in your process.

Below is a screen shot of the Request Content mapping from QAD EQMS. The JSON values are mapped to the respective EQMS fields within this process.

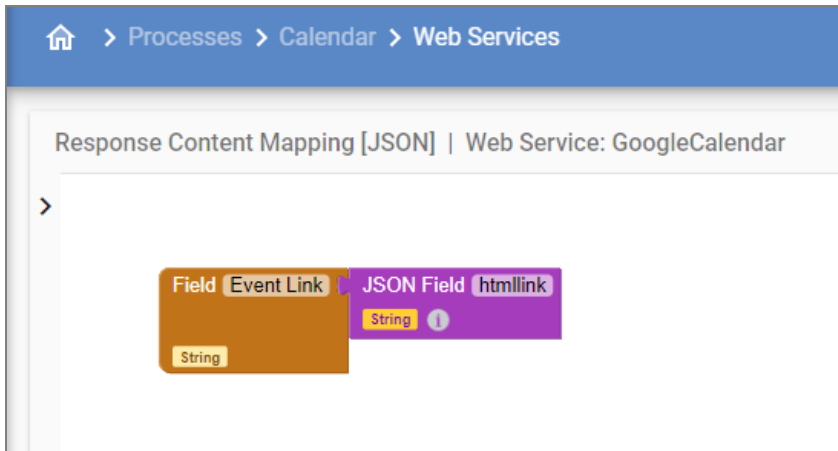
Fig. 6: Web Services screen, Content Mapping Edit Expression

**Step 6**

1. Navigate to the Response tab.
2. Select the appropriate content type and response type (see third party for required types).
3. Click Edit Expression.

The example below uses a text field on the process titled Event Link.

**Fig. 7: Web Services screen, Response Edit Expression**



**Step 7**

Your final step is to publish and deploy the process containing your new web service.

In this document's example, the integration event is tied to a command button. However, REST integration can be set up to execute like any other integration type.