

## MQ REST API: Technical and Troubleshooting Notes

<https://www.ibm.com/support/pages/node/6208422>

Date last updated: 23-Jun-2020

Angel Rivera - [rivera@us.ibm.com](mailto:rivera@us.ibm.com)  
IBM MQ Support

### +++ Objective

To provide a good starting point for MQ Administrators to learn more about the new feature:

REST API

The chapters are:

Chapter 1: Overview (FAQs, history, references)

Chapter 2: Installation (effects on /opt/mqm)

Chapter 3: Setup and configuration (effects on /var/mqm and queue managers)

Chapter 4: Diagnostics review and what doc to gather (runmqras)

Chapter 5: Common errors and how to resolve them

### ++ Related documentation

The corresponding document for the MQ Web Server and Web Console is:

<https://www.ibm.com/support/pages/node/6208289>

MQ Web Server and Web Console: Technical and Troubleshooting Notes

The following tutorials provide very detailed instructions.

<https://www.ibm.com/support/pages/node/6208006>

Using the MQ 9.1.5 CD REST API in Linux with no security (for Testing environments)

<https://www.ibm.com/support/pages/node/6209110>

MQ 9.1 REST API, remote administration via a gateway queue manager

+++++ Chapter 1: Overview +++++

The MQ REST API is available by enabling the MQ Web Server, which is based on WebSphere Liberty Profile (WLP) which is included with the MQ Server.  
The MQ Web Server provides an HTTP based administration facilities via the MQ Web Console and the MQ REST API.

The following presentation provides excellent information regarding these features:

[https://www.mqtechconference.com/sessions\\_v2018/MQTC\\_v2018\\_MQAdmin\\_Console\\_REST.pdf](https://www.mqtechconference.com/sessions_v2018/MQTC_v2018_MQAdmin_Console_REST.pdf)

MQ Administration, the Web Console, & REST API

Sam Goulden, IBM MQ L3 Service,

+ Slide 16: What is REST?

- REpresentational State Transfer
- HTTP is an example of a RESTful architecture
- HTTP defines resources (URL/URIs) and the operations (HTTP verbs) which can use them
  - Originally used for serving web-pages
  - Work really well for APIs too
- Generally light-weight and relatively simple to use, much simpler than SOAP webservices
- Have become incredibly common in recent years
- MQ has taken the approach of following best-practice, and adherence to the various w3c standards when defining its REST API

+ Slide 17: MQ REST API

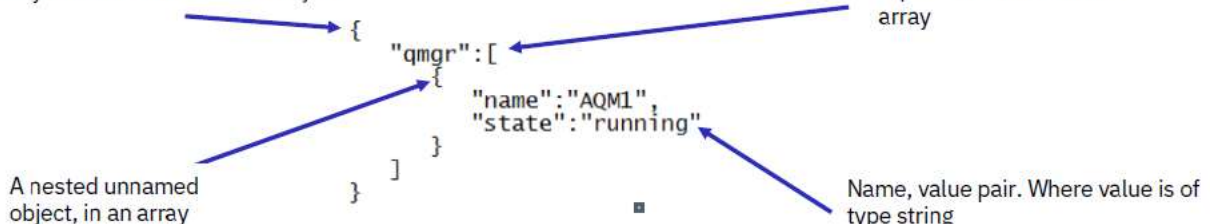
- An administrative API for managing MQ via REST
- It is much more intuitive to use than PCF and makes it easier to create MQ tooling, that is, a selfservice web-browser based MQ portal using JavaScript
  - „ No need for an MQ client!
  - „ Callable from any language which can invoke an HTTPS endpoint
  - „ Many languages now have built in, or easily added, support for REST
- Payload format is JSON (JavaScript Object Notation)
- Human readable, not a binary format

### Payload format is JSON (JavaScript Object Notation)

- Human readable, not a binary format

Curly bracket denotes JSON object

Square bracket denotes JSON array



## ++ FAQs

+ Do customers need to buy an extra license to use the WebSphere Liberty Profile?

Answer:

No. The purchase of the MQ Server license includes already the use of the WLP. Similarly with the JRE that is shipped with MQ, the use of WLP (and JRE) is restricted for MQ usage (that is, do not copy the WLP or JRE to a host that does not have MQ in order to use those features outside the scope of MQ).

+ What is the mapping between the MQ version.release.level and the version of WLP?

The following is obtained from the header of console.log or messages.log:

MQ 9.0.4 CD:

Launching mqweb (WebSphere Application Server 17.0.0.2/wlp-1.0.17.cl170220170523-1818) on IBM J9 VM, version pwa6480sr4fp11-20170823\_01 (SR4 FP11) (en\_US)

MQ 9.0.5 CD:

Launching mqweb (WebSphere Application Server 17.0.0.4/wlp-1.0.19.201712061531) on IBM J9 VM, version 8.0.5.10 - pwa6480sr5fp10-20180214\_01(SR5 FP10) (en\_US)

MQ 9.1.0.4 LTS:

Launching mqweb (WebSphere Application Server 19.0.0.9/wlp-1.0.32.cl190920190905-0148) on IBM J9 VM, version 8.0.5.40 - pwa6480sr5fp40-20190807\_01(SR5 FP40) (en\_US)

MQ 9.1.0.5 LTS:

Launching mqweb (WebSphere Application Server 19.0.0.12/wlp-1.0.35.cl191220191120-0300) on IBM J9 VM, version 8.0.6.5 - pxa6480sr6fp5ifix-20200228\_01(SR6 FP5+IJ23014) (en\_US)

MQ 9.1.5 CD:

Launching mqweb (WebSphere Application Server 19.0.0.12/wlp-1.0.35.cl191220191120-0300) on IBM J9 VM, version 8.0.6.5 - pxa6480sr6fp5ifix-20200228\_01(SR6 FP5+IJ23014) (en\_US)

+ Does the MQ REST API have the same limitation of the MQ Web Console where only Qmgrs from the SAME host and installation can be managed?

No!

The MQ REST API does not have the same limitation of the MQ Web Console.

You can have a “gateway” queue manager from which you could remotely administer queue managers in other installations in the same server, or in remote servers.

The restriction is that this gateway queue manager must be in the SAME installation as the MQ Web Server.

#### + Regarding Multi-Instance queue managers

There is no equivalent to a `connectionNameList` when accessing the REST API for Multi-Instance queue managers.

- That is, there is no automatic reconnection from CURL or another tool/language that connects to the MQ Web Server to deal with the failover or switchover and move the URL from host-1 to host-2

- It might be possible to use a floating IP address to isolate the REST API for knowing the actual running host.

#### ++ History

- + MQ 9.0.1 CD added support for a number of HTTP-based administration capabilities

- Focus on low barrier to entry and ease of use
- MQ Console, a web-browser based graphical administration tool
- MQ REST API, a programmatic administration API
- Enhanced further during CD deliverables 9.0.x and 9.1.x

- + MQ 9.0.4 CD added support for administration of remote queue managers.

#### + MQ 9.1

[V9.1.0 Jul 2018]

You can use the administrative REST API to administer IBM® MQ objects, such as queue managers and queues, and Managed File Transfer agents and transfers.

Information is sent to, and received from, the administrative REST API in JSON format.

These RESTful APIs can help you to embed IBM MQ administration into popular DevOps and automation tooling.

## ++ References

+ Main page from the online KC:

[https://www.ibm.com/support/knowledgecenter/SSFKSJ\\_9.1.0/com.ibm.mq.adm.doc/q127590\\_.htm](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.adm.doc/q127590_.htm)

IBM MQ 9.1.x / IBM MQ / Administering /  
Administration using the REST API

## Subtopics

- [V 9.1.0 Getting started with the administrative REST API](#)  
Get started quickly with the administrative REST API and try out a few example requests by using cURL to create, update, view, and delete a queue.
- [V 9.1.0 Remote administration using the REST API](#)  
You can use the REST API to administer remote queue managers, and the IBM MQ objects that are associated with those queue managers. This remote administration includes queue managers that are on the same system, but not in the same IBM MQ installation as the mqweb server. Therefore, you can use the REST API to administer your entire IBM MQ network with only one installation that runs the mqweb server. To administer remote queue managers, you must configure the administrative REST API gateway so that at least one queue manager in the same installation as the mqweb server acts as a gateway queue manager. Then, you can specify the remote queue manager in the REST API resource URL to perform the specified administrative action.
- [V 9.1.0 REST API time stamps](#)  
When date and time information is returned by the administrative REST API, it is returned in Coordinated Universal Time (UTC), and in a set format.
- [V 9.1.0 REST API error handling](#)  
The REST API reports errors by returning an appropriate HTTP response code, for example 404 (Not Found), and a JSON response. Any HTTP response code that is not in the range 200 - 299 is considered an error.
- [V 9.1.0 REST API discovery](#)  
Documentation for the REST API is available within the IBM Knowledge Center and in Swagger format. Swagger is a commonly used approach for documenting REST APIs. The Swagger documentation for the REST API can be viewed by enabling the API discovery feature on the mqweb server.
- [V 9.1.0 REST API national language support](#)  
The REST API supports, with certain qualifications, the ability to specify national languages as part of an HTTP request.
- [V 9.1.0 REST API versions](#)  
At IBM MQ version 9.1, the REST API is at version 1. From IBM MQ Version 9.1.5, the REST API is at version 2. This version number forms part of the base URL for REST requests. For example,

`https://localhost:9443/ibmmq/rest/v2/admin/installation`. The version number is used to isolate clients from changes to the REST API that might be introduced in future releases.

#### + Administrative REST API reference

[https://www.ibm.com/support/knowledgecenter/SSFKSJ\\_9.1.0/com.ibm.mq.ref.adm.doc/q127980.htm](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.ref.adm.doc/q127980.htm)

IBM MQ 9.1.x / IBM MQ / Reference / Administration reference /  
Administrative REST API reference

Reference information about the administrative REST API.

For more information about using the administrative REST API, see [Administration using the REST API](#).

For more information about configuring the administrative REST API, see [Configuring the REST API](#).


For more information about securing the administrative REST API, see [Securing the REST API](#).

#### *Subtopics*

- [REST API resources](#)

This collection of topics provides reference information for each of the administrative REST API resources.

- [REST API and PCF equivalents](#)

 For most REST API optional query parameters and attributes, an equivalent PCF parameter or attribute exists. Use these topics to understand these equivalents.

[https://www.ibm.com/support/knowledgecenter/SSFKSJ\\_9.1.0/com.ibm.mq.dev.doc/q130960\\_.htm](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.dev.doc/q130960_.htm)

IBM MQ 9.1.x / IBM MQ / Developing applications / Developing REST applications with IBM MQ / Messaging using the REST API /  
Using the messaging REST API

[https://www.ibm.com/support/knowledgecenter/SSFKSJ\\_9.1.0/com.ibm.mq.ref.dev.doc/q130740\\_.htm](https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.1.0/com.ibm.mq.ref.dev.doc/q130740_.htm)

IBM MQ 9.1.x / IBM MQ / Reference / Developing applications reference / Messaging REST API reference / REST API resources /  
/messaging/qmgr/{qmgrName}/queue/{queueName}/message

You can use the HTTP POST method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to put messages to the specified queue on the specified queue manager. You can use the HTTP DELETE method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to get messages from the specified queue on the specified queue manager. **V 9.1.3** You can use the HTTP get method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to browse messages from the specified queue on the specified queue manager.

### Subtopics

- [POST](#)

**V 9.1.0** You can use the HTTP POST method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to put messages to the specified queue on the specified queue manager.

- [GET](#)

You can use the HTTP GET method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to browse messages from the associated queue manager and queue.

- [DELETE](#)

**V 9.1.0** You can use the HTTP DELETE method with the /messaging/qmgr/{qmgrName}/queue/{queueName}/message resource to get messages from the associated queue manager and queue.

+ IBM INTERNAL

<https://supportcontent.ibm.com/support/pages/node/6208288>

MQ Skill Transfer: Web Server, Web Console and REST API - IBM INTERNAL

+++++  
+++ Chapter 2: Installation  
+++++

See:

Chapter 2: Installation (effects on /opt/mqm)

From related document:

<https://www.ibm.com/support/pages/node/6208289>

MQ Web Server and Web Console: Technical and Troubleshooting Notes



+++++  
+++ Chapter 3: Setup and configuration  
+++++

See:

Chapter 3: Setup and configuration (effects on /var/mqm and queue managers)

From related document:

<https://www.ibm.com/support/pages/node/6208289>

MQ Web Server and Web Console: Technical and Troubleshooting Notes

+ Ports used:

mqm@orizaba1.fyre.ibm.com: /home/mqm

\$ dspmqweb

MQWB1124I: Server 'mqweb' is running.

URLS:

<https://orizaba1.fyre.ibm.com:9443/ibmmq/rest/>

<http://orizaba1.fyre.ibm.com:9080/ibmmq/rest/>

<https://orizaba1.fyre.ibm.com:9443/ibmmq/console/>

<http://orizaba1.fyre.ibm.com:9080/ibmmq/console/>

```
+++++
+++ Chapter 4: Diagnostics review and what doc to gather
+++++
```

See:

Chapter 4: Diagnostics review and what doc to gather (runmqras)

From related document:

<https://www.ibm.com/support/pages/node/6208289>

MQ Web Server and Web Console: Technical and Troubleshooting Notes

++ The 2 most important diagnostic files are:

console.log  
messages.log

++ Where to find the files at the MQ server at the customer side

You can ask the customer to go into the web directory under /var/mqm

```
$ cd /var/mqm/web
```

Then go into

```
$ cd installations/Installation1/servers/mqweb
```

```
$ ls -l
```

```
-rw-rw-r-- 1 mqm mqm 858 Aug 28 2019 jvm.options
drwxrws--- 4 mqm mqm 128 May 8 08:02 logs
-rw-rw---- 1 mqm mqm 891 Aug 28 2019 mqwebuser.original.xml
-rw-rw---- 1 mqm mqm 2871 May 8 08:02 mqwebuser.xml
-r--r----- 1 mqm mqm 2831 May 8 08:01 no_security.xml
drwxrwsr-x 3 mqm mqm 22 Mar 25 09:20 resources
-rw-r----- 1 mqm mqm 589 Aug 28 2019 server.xml
drwxrwsr-x 6 mqm mqm 176 May 8 08:02 workarea
```

```
$ cd logs
```

(Full directory: cd installations/Installation1/servers/mqweb/logs)

```
-rw-rw-r-- 1 mqm mqm 1678 May 8 08:09 console.log
drwxrwsr-x 2 mqm mqm 176 May 4 05:22 ffdc
-rw-rw-r-- 1 mqm mqm 17968 May 8 08:00 messages_20.05.08_08.02.35.0.log
-rw-rw-r-- 1 mqm mqm 15156 May 8 08:09 messages.log
drwxrwsr-x 2 mqm mqm 64 May 8 08:02 state
-rw-rw-r-- 1 mqm mqm 761 May 8 08:02 status.xml
```

```
$ cd ffdc
```

(Full directory: cd installations/Installation1/servers/mqweb/logs/ffdc)

```
-rw-rw-r-- 1 mqm mqm 622 Apr 8 11:24 exception_summary_20.04.08_10.27.59.0.log
-rw-rw-r-- 1 mqm mqm 608 May 8 07:18 exception_summary_20.05.04_05.22.49.0.log
-rw-rw-r-- 1 mqm mqm 27566 Apr 8 10:27 ffdc_20.04.08_10.27.58.0.log
-rw-rw-r-- 1 mqm mqm 16571 May 4 05:22 ffdc_20.05.04_05.22.49.0.log
```

`++ runmqras`

For “runmqras”, starting with MQ 9.1.0, there is a new value for “section”:

[V9.1.0 Jul 2018] **mqweb**

Gathers trace and configuration data for the mqweb server.

General format:

`runmqras -section mqweb -caseno TS12345 -ftp IBM`

Real example:

Host “orizaba1” (Linux) has MQ 9.1.5 CD and the MQ Web Server was enabled.

The ticket TS003658930 is going to be used

The following will be the “baseline” without any sections, in order to allow a comparison when doing the section for the mqweb (next step):

`$ runmqras -caseno TS003658930 -ftp IBM`

File name:

`TS003658930.runmqras_202005081111-orizaba1.fyre.ibm.com.zip`

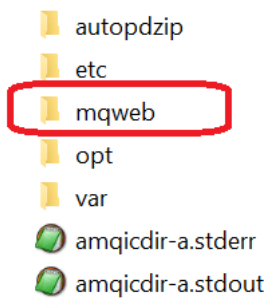
Now let’s use the “-section mqweb”:

`$ runmqras -section mqweb -caseno TS003658930 -ftp IBM`

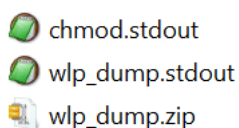
File name:

`TS003658930.runmqras_202005081114-orizaba1.fyre.ibm.com.zip`

After unzipping the zip files, there is only ONE item that is different:  
a subdirectory named “mqweb”



Inside the directory “mqweb” there are 3 entries:



Notice: due to the long names in directories, it might be possible to reach a limit on the full path names in Windows, preventing you from unzipping:

wlp\_dump.zip

You may need to copy the wlp\_dump.zip file into a shorter overall directory, such as:

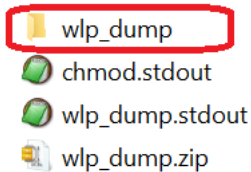
C:\temp

And then unzip it there, which will give a directory called:

wlp\_dump

You may need to move that directory “wlp\_dump” into the original directory tree under the unzipped runmqras for consistency.

Name



Even though there will be MANY files under “wlp\_dump”, there are only a handful that are of interest to MQ Support:

UNZIPPED-DIRECTORY-FROM-RUNMQRAS\mqweb\wlp\_dump\logs

```
C:\runmqras-mqweb-orizaba1.fyre.ibm.com\mqweb\wlp_dump\logs> dir
05/08/2020 11:23 AM          1,842 console.log
05/08/2020 11:23 AM              0 dump.log
05/08/2020 02:29 PM    <DIR>          ffdc
05/08/2020 11:23 AM     15,494 messages.log
05/08/2020 08:00 AM     17,968 messages_20.05.08_08.02.35.0.log
05/08/2020 02:29 PM    <DIR>          state
05/08/2020 08:02 AM          761 status.xml
```

```

+++++
+++ Chapter 5: Common errors and how to resolve them
+++++

```

+ For some URIs you need to specify the versioning (v1 or v2):

```

mqm@orizaba1.fyre.ibm.com: /var/mqm/web/installations/Installation1/servers/mqweb
$ curl -k https://orizaba1.fyre.ibm.com:9443/ibmmq/rest/admin/qmgr -X GET
{"error": [{
  "action": "Resubmit the request using a valid URI.",
  "completionCode": 0,
  "explanation": "There is no corresponding REST interface on the provided URI.",
  "message": "MQWB0116E: The URI cannot be invoked as it does not correspond to an
existing REST interface.",
  "msgId": "MQWB0116E",
  "reasonCode": 0,
  "type": "rest"
}]}
```

But by adding the versioning “v1”, it works fine:

```

mqm@orizaba1.fyre.ibm.com: /var/mqm/web/installations/Installation1/servers/mqweb
$ curl -k https://orizaba1.fyre.ibm.com:9443/ibmmq/rest/v1/admin/qmgr -X GET
{"qmgr": [
  {
    "name": "QMORI915",
    "state": "endedImmediately"
  },
  {
    "name": "QMDEMO",
    "state": "running"
  },
  {
    "name": "QMORI",
    "state": "running"
  }
]}
```

+ You need to escape double quotes that are inside double quotes

```
mqm@orizaba1.fyre.ibm.com: /home/mqm
$ curl -s -k "https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QMORI/queue" -X POST -H
"Content-Type: application/json" --data "{ \"name\":\"Q2\" }"
{"error": [{
  "action": "Resubmit the request using the correct format and syntax.",
  "completionCode": 0,
  "explanation": "The REST API request failed as the data in the request payload could not be
parsed.",
  "message": "MQWB0107E: Unable to parse the request data due to exception 'Unexpected
character ('n' (code 110)): was expecting double-quote to start field name'.",
  "msgId": "MQWB0107E",
  "reasonCode": 0,
  "type": "rest"
}]}
```

The following has the correction: using escaped double quotes

```
mqm@orizaba1.fyre.ibm.com: /home/mqm
$ curl -s -k "https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QMORI/queue" -X POST -H
"Content-Type: application/json" --data "{ \"name\":\"Q2\" }"
mqm@orizaba1.fyre.ibm.com: /home/mqm
(1068) echo $?
0
```

+ You try to use the REST API for a queue manager using MQ 8.0 (such as QM80)

```
mqm@orizaba1.fyre.ibm.com: /home/mqm
```

```
$ curl -s -k "https://orizaba1.fyre.ibm.com:9443/ibmmq/rest/v1/admin/qmgr/QM80" -X GET
```

```
{"error": [{  
  "action": "Define the required queues on the target queue manager and resubmit the  
request.",  
  "completionCode": 2,  
  "explanation": "The REST API request failed as a queue required to handle PCF  
requests is not defined on the queue manager.",  
  "message": "MQWB0111E: REST API queue 'SYSTEM.REST.REPLY.QUEUE' is not defined  
on queue manager 'QM80'.",  
  "msgId": "MQWB0111E",  
  "reasonCode": 2085,  
  "type": "rest"  
}]}
```

Workaround: You could exploit the remote administration by means of a gateway queue manager.

+ Error handling, for example, trying to delete a queue that does not exist

```
mqm@orizaba1.fyre.ibm.com: /home/mqm
$ curl -s -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QMORI/queue/Q2?purge -X
DELETE
{"error": [{
  "action": "Resubmit the request with the name of an existing queue, or with no queue
name to retrieve a list of queues.",
  "completionCode": 2,
  "explanation": "The REST API was invoked specifying a queue name which cannot be
located.",
  "message": "MQWB0037E: Cannot find the queue 'Q2'. The queue manager reason code
is 2085 : 'MQRC_UNKNOWN_OBJECT_NAME'.",
  "msgId": "MQWB0037E",
  "reasonCode": 2085,
  "type": "rest"
}]}
```

The solution is to specify a queue that exists.

The objective of this example was to show the type of error message that you get in this situation.

+++ end