The MQ Console and REST API

Matt Leming

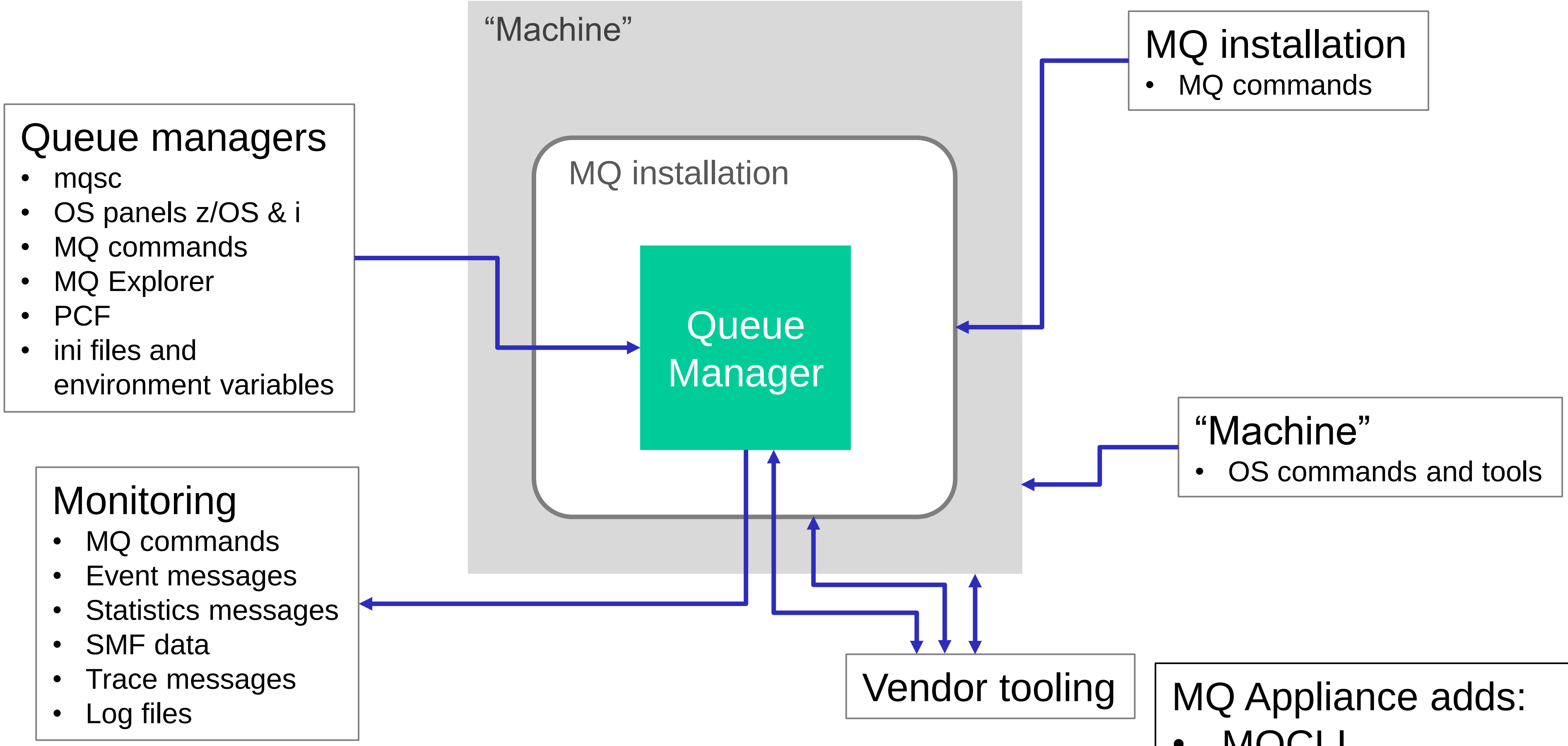
lemingma@uk.ibm.com

Agenda

- **Existing capabilities**
- **What's new?**
  - The mqweb server
  - The MQ REST API
  - The MQ Console

Existing capabilities

Administering software MQ



“Machine”

MQ installation

Queue Manager

Queue managers
- mqsc
- OS panels z/OS & i
- MQ commands
- MQ Explorer
- PCF
- ini files and environment variables

MQ installation
- MQ commands

“Machine”
- OS commands and tools

Monitoring
- MQ commands
- Event messages
- Statistics messages
- SMF data
- Trace messages
- Log files

Vendor tooling

MQ Appliance adds:
- MQCLI
- MQ Console
- Appliance Console

From a tooling perspective PCF is key

Why we need more

- **While PCF is very powerful, it is not that easy to use**
  - Requires an MQ client, and a supported programming language
  - Binary format
  - Multiple messages generated per request
  - There are tools to make this easier

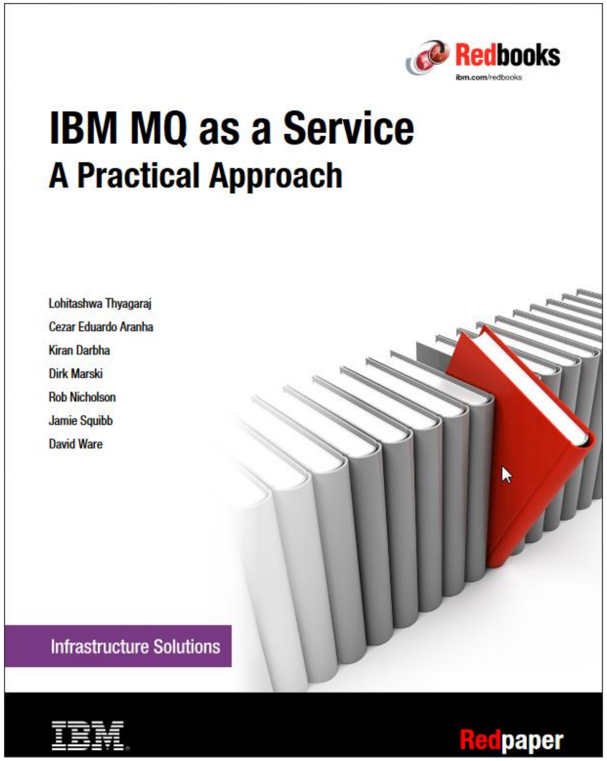
```
**** Message ****

 length - 724 of 724 bytes

 00000000: 080A 4103 0000 0000 5744 5220 0200 0000 '..A.....WDR ....'
 00000010: 8800 0000 6700 0000 514D 4752 315F 3230 'ˆ...g...QMGR1_20'
 00000020: 3135 2D31 302D 3239 5F30 392E 3431 2E31 '15-10-29_09.41.1'
 00000030: 3620 2020 2020 2020 2020 2020 2020 2020 '6               '
 00000040: 2020 2020 2020 2020 514D 4752 3120 2020 ' QMGR1          '
 00000050: 2020 2020 2020 2020 2020 2020 2020 2020 '                '
 00000060: 2020 2020 2020 2020 2020 2020 2020 2020 '                '
 00000070: 2020 2020 2020 2020 0000 0000 0000 0000 ' ........       '
 00000080: 58CA 0000 0000 0000 0000 0000 0000 0000 'X...............'
 00000090: 644E 4656 2116 4656 3230 3135 2D31 302D 'dNFV!.FV2015-10-'
 000000A0: 3239 2020 0000 0000 3039 2E34 312E 3233 '29 ....09.41.23 '
 000000B0: 0100 0000 4D51 4D4D 0000 0000 3038 3030 '....MQMM....0800'
 000000C0: 3030 3034 0000 0000 434C 5553 5445 5231 '0004....CLUSTER1'
 000000D0: 2E51 4D47 5231 2020 2020 2020 0B00 0000 '.QMGR1 ....
 000000E0: 0800 0000 0200 0000 2020 2020 2020 2020 '........
 000000F0: 2020 2020 2020 2020 2020 2020 2020 2020 '
```

- **There is a growing need for the ability to administer MQ from**
  - Any environment
  - Any programming language
  - By users who are not expert in MQ

- **Lots of customers are writing self-service web-portals for managing their infrastructure, including MQ**



Redbooks
ibm.com/redbooks

IBM MQ as a Service
A Practical Approach

Lohitashwa Thyagaraj
Cezar Eduardo Aranha
Kiran Darbha
Dirk Marski
Rob Nicholson
Jamie Squibb
David Ware

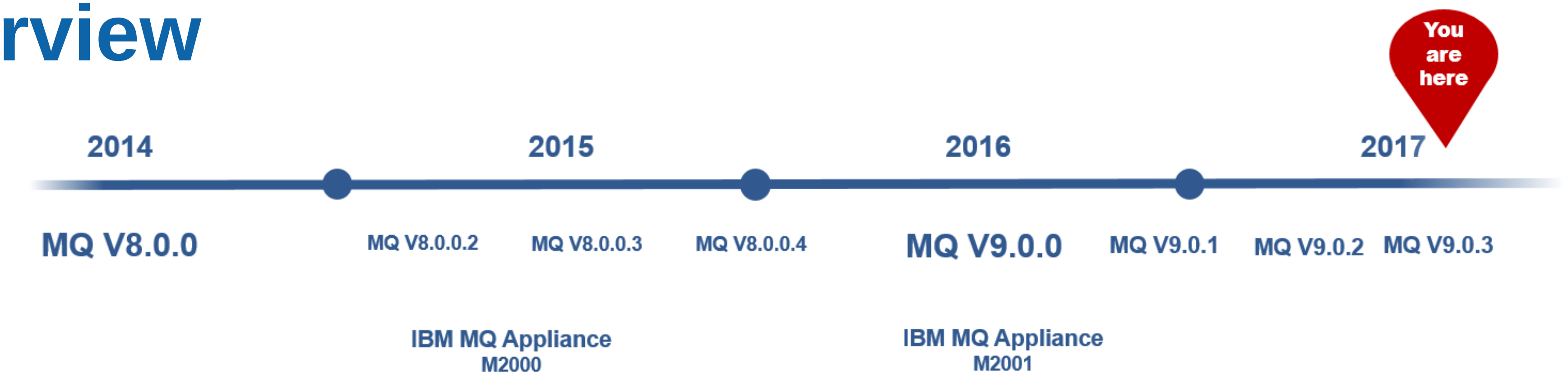
Infrastructure Solutions

IBM

Redpaper

What’s new?

Overview



You are here

2014 2015 2016 2017

MQ V8.0.0 MQ V8.0.0.2 MQ V8.0.0.3 MQ V8.0.0.4 MQ V9.0.0 MQ V9.0.1 MQ V9.0.2 MQ V9.0.3

IBM MQ Appliance M2000

IBM MQ Appliance M2001

- **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 in 9.0.2 and 9.0.3

- **As 9.0.1 is a CD release these capabilities are supported on a subset of platforms**
  - Windows, Linux and z/OS

- **Need a web-serving environment to run in**
  - The mqweb server

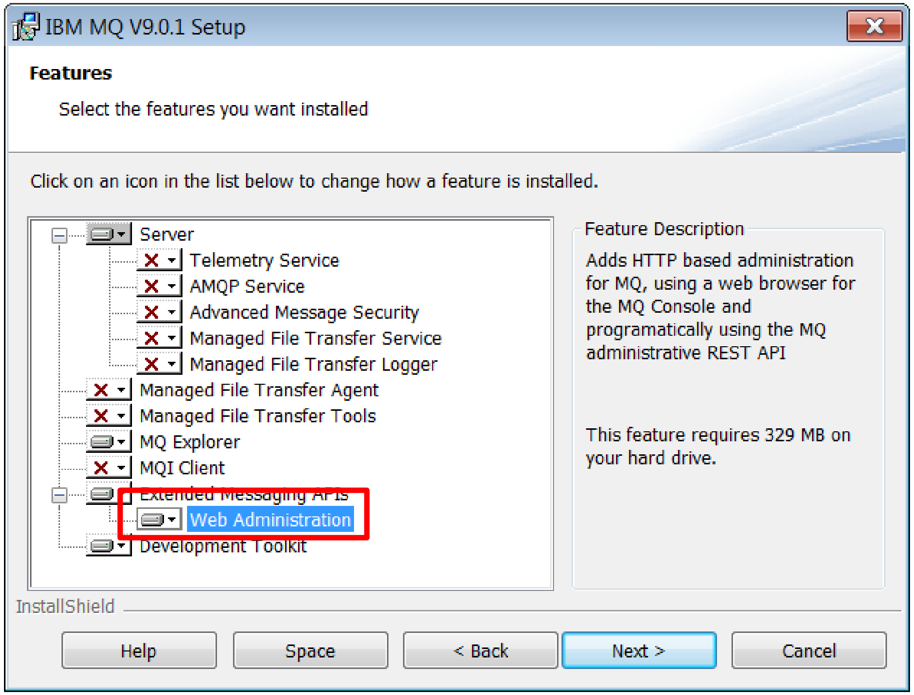
V9.0.1

The mqweb server

V9.0.1

Web component

- **A new optional install component**
- **Contains the MQ Console, MQ administrative REST API plus prereqs**
  - WebSphere Liberty Profile which runs the mqweb server
- **New USS FMID on z/OS**
  - JMS9016



IBM MQ V9.0.1 Setup

Features

Select the features you want installed

Click on an icon in the list below to change how a feature is installed.

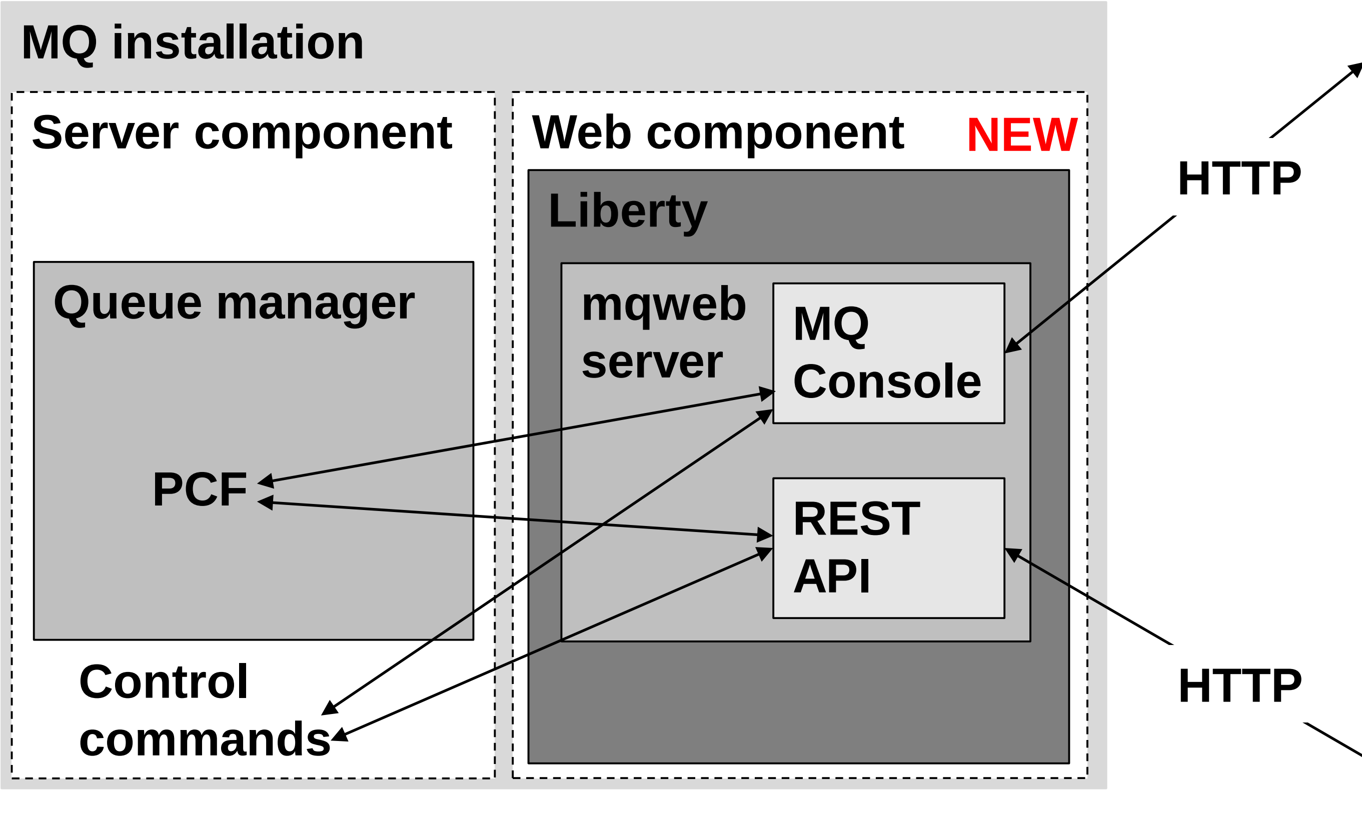
- Server
  - ✗ Telemetry Service
  - ✗ AMQP Service
  - ✗ Advanced Message Security
  - ✗ Managed File Transfer Service
  - ✗ Managed File Transfer Logger
- ✗ Managed File Transfer Agent
- ✗ Managed File Transfer Tools
- MQ Explorer
- ✗ MQI Client
- Extended Messaging APIs
  - Web Administration
- Development Toolkit

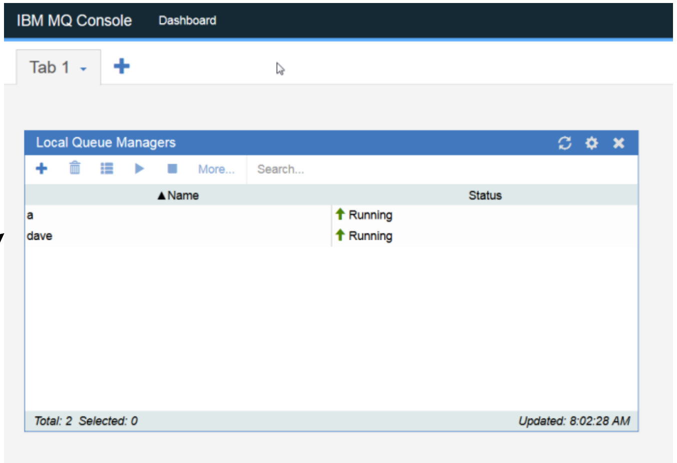
Feature Description

Adds HTTP based administration for MQ, using a web browser for the MQ Console and programatically using the MQ administrative REST API

This feature requires 329 MB on your hard drive.

Perhaps a picture would help?

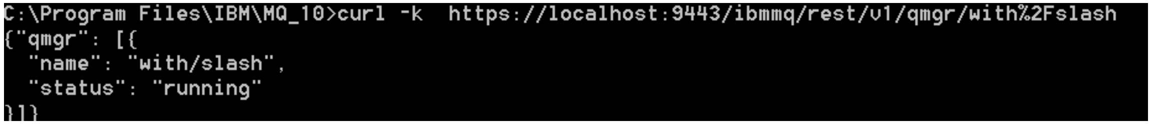




▲Name	Status
a	Running
dave	Running

Total: 2 Selected: 0

Updated: 8:02:28 AM



```
C:\Program Files\IBM\MQ_10>curl -k  https://localhost:9443/ibmmq/rest/v1/qmgr/with%2Fslash
{"qmgr": [{
  "name": "with/slash",
  "status": "running"
}]}
```

V9.0.1

The mqweb server

- **The MQ Console and REST API are applications that run in a WebSphere Liberty Profile (WLP) server called mqweb**
  - WLP is provided as part of MQ install
  - mqweb server definition provided out of the box when installing the web component

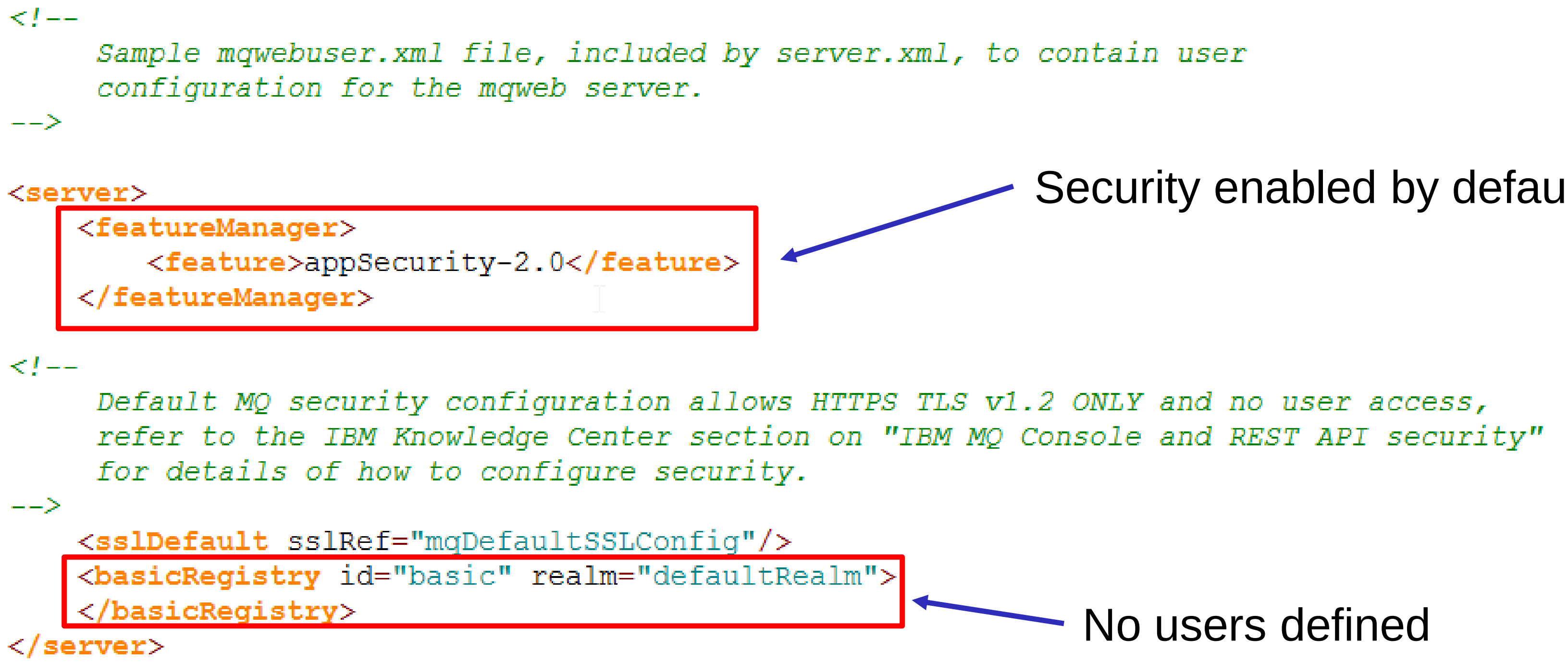
- **Once installed:**
  - MQ Console is enabled
  - REST API is enabled (at 9.0.2, disabled at 9.0.1)
  - HTTPS on localhost only
  - Locked down

```
CWWKE0001I: The server mqweb has been launched.
CWWKG0028A: Processing included configuration resource: C:\Program
Files\IBM\Latest902\web\mq\etc\mqweb.xml
A CWWKG0028A: Processing included configuration resource: C:\Program Files (x86)\IBM\WebSphere
MQ\web\installations\Latest902\servers\mqweb\mqwebuser.xml
CWWKE0002I: The kernel started after 2.493 seconds
CWWKF0007I: Feature update started.
CWWKO0219I: TCP Channel defaultHttpEndpoint-ssl has been started and is now listening for requests on host
127.0.0.1  (IPv4: 127.0.0.1) port 9443.
CWWKZ0018I: Starting application com.ibm.mq.rest.
CWWKZ0018I: Starting application com.ibm.mq.console.
SRVE0169I: Loading Web Module: com.ibm.mq.rest.v1.
SRVE0250I: Web Module com.ibm.mq.rest.v1 has been bound to default_host.
CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/rest/v1/
CWWKZ0001I: Application com.ibm.mq.rest started in 0.518 seconds.
SRVE0169I: Loading Web Module: mqconsole.
SRVE0250I: Web Module mqconsole has been bound to default_host.
CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/console/
SRVE0169I: Loading Web Module: com.ibm.mq.consoleinternal.
SRVE0250I: Web Module com.ibm.mq.consoleinternal has been bound to default_host.
CWWKT0016I: Web application available (default_host): https://localhost:9443/ibmmq/console/internal/
CWWKZ0001I: Application com.ibm.mq.console started in 0.525 seconds.
CWWKF0012I: The server installed the following features: [concurrent-1.0, jsp-2.2, servlet-3.1, ssl-1.0, jndi-1.0,
basicAuthenticationMQ-1.0, websocket-1.0, json-1.0, localConnector-1.0, jaxrs-1.1].
CWWKF0008I: Feature update completed in 2.095 seconds.
CWWKF0011I: The server mqweb is ready to run a smarter planet.
REST023: MQ REST API level: p902-dfct-L170216.1
```

V9.0.1

Configuring mqweb server

- **Currently done by editing xml (standard WLP approach)**
- **File called mqwebuser.xml provided in MQ data directory**
  - This is the only part of the WLP xml configuration that we support customers editing:



```
<!--
     Sample mqwebuser.xml file, included by server.xml, to contain user
     configuration for the mqweb server.
-->

<server>
    <featureManager>
        <feature>appSecurity-2.0</feature>
    </featureManager>

<!--
     Default MQ security configuration allows HTTPS TLS v1.2 ONLY and no user access,
     refer to the IBM Knowledge Center section on "IBM MQ Console and REST API security"
     for details of how to configure security.
-->
    <sslDefault sslRef="mqDefaultSSLConfig"/>
    <basicRegistry id="basic" realm="defaultRealm">
    </basicRegistry>
</server>
```

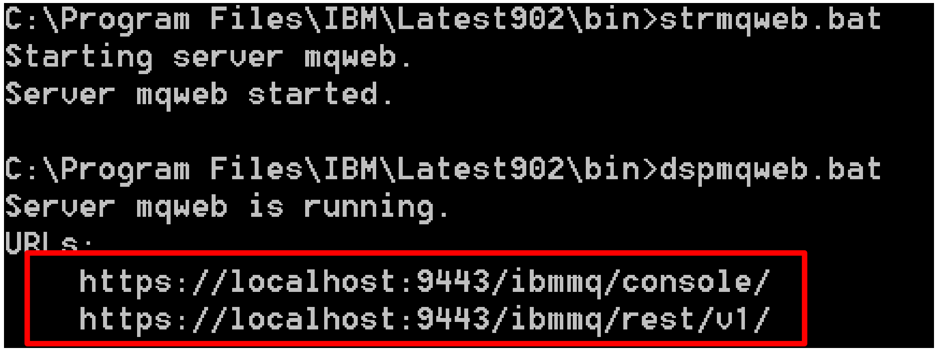
Security enabled by default

No users defined

V9.0.1

Managing mqweb server

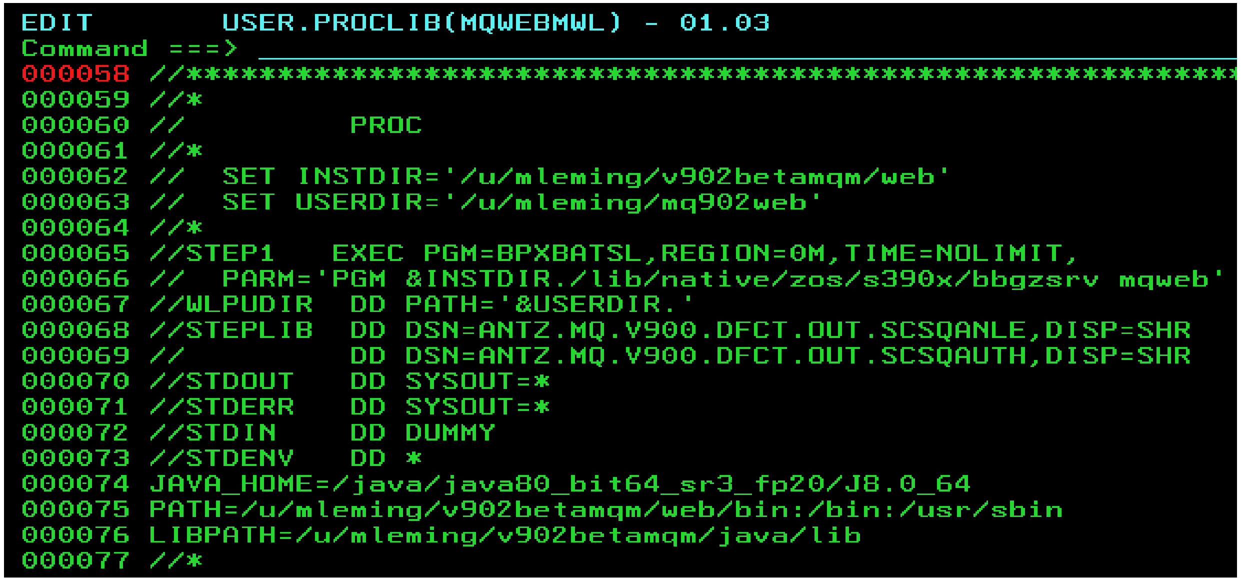
- **Distributed: three new control commands**
  - strmqweb, endmqweb, dspmqweb



```
C:\Program Files\IBM\Latest902\bin>strmqweb.bat
Starting server mqweb.
Server mqweb started.

C:\Program Files\IBM\Latest902\bin>dspmqweb.bat
Server mqweb is running.
URLs:
    https://localhost:9443/ibmmq/console/
    https://localhost:9443/ibmmq/rest/v1/
```

- **z/OS: Sample JCL – CSQ4WEBS – provided**
  - Sets all necessary variables up and then starts up mqweb server



```
EDIT        USER.PROCLIB(MQWEBMWL) - 01.03
Command ===> 
//**********************************************************
//*
//         PROC
//*
//  SET INSTDIR='/u/mleming/v902betamqm/web'
//  SET USERDIR='/u/mleming/mq902web'
//*
//STEP1   EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//  PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv mqweb'
//WLPUDIR  DD PATH='&USERDIR.'
//STEPLIB  DD DSN=ANTZ.MQ.V900.DFCT.OUT.SCSQANLE,DISP=SHR
//         DD DSN=ANTZ.MQ.V900.DFCT.OUT.SCSQAUTH,DISP=SHR
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDIN    DD DUMMY
//STDENV   DD *
JAVA_HOME=/java/java80_bit64_sr3_fp20/J8.0_64
PATH=/u/mleming/v902betamqm/web/bin:/bin:/usr/sbin
LIBPATH=/u/mleming/v902betamqm/java/lib
//*
```

V9.0.1

The MQ REST API

What is REST?

- **REpresentational State Transfer**
  - Term first coined by Roy Fielding in his PhD thesis
  - An architectural style
  - Based off his earlier work defining the HTTP and other web based specifications

- **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 web-services**
  - Have become incredibly common in recent years

- **However there are lots of interpretations of what it means to be RESTful**
  - MQ has taken the approach of following best-practice, and adherence to the various w3c standards when defining its REST API

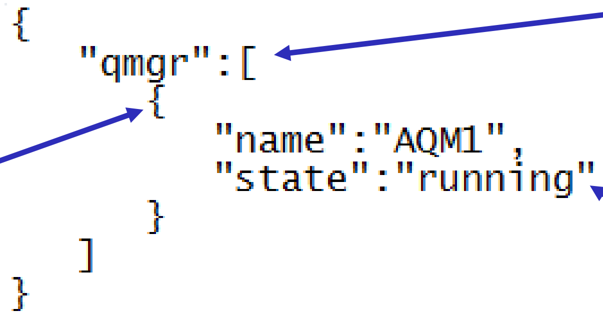
V9.0.1

MQ REST API

- **An administrative API for managing MQ via REST**
- **Is much more intuitive to use than PCF and makes it easier to create MQ tooling, e.g. a self-service 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

Curly bracket denotes JSON object

Square bracket denotes JSON array



```
{
    "qmgr":[
        {
            "name":"AQM1",
            "state":"running"
        }
    ]
}
```

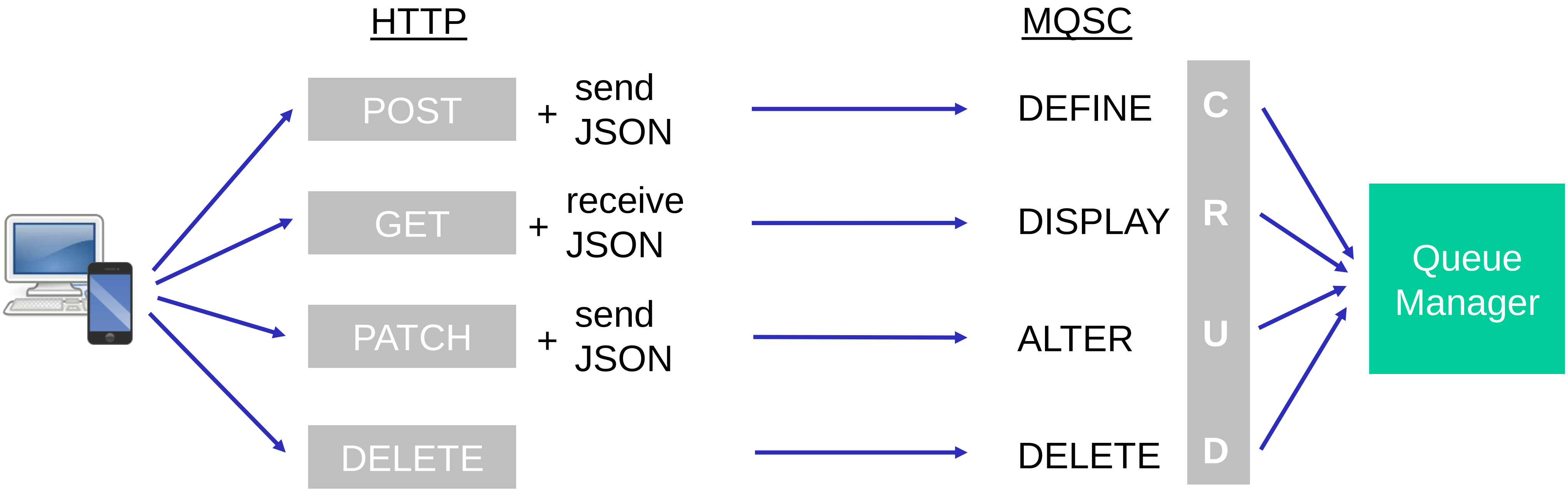
A nested unnamed object, in an array

Name, value pair. Where value is of type string

V9.0.1

MQ REST API

- **Based off underlying MQ capabilities such as PCF and control commands, but adjusted to adhere to RESTful practices**
- **URL represent target object for command**



V9.0.1

Evolution of the MQ REST API

- **Iteratively developed in CD releases**
  - 9.0.1:
    - REST API introduced
    - Contains ability to list queue managers (dspmq) and their installation (dspmqver)
    - Not integrated into mqweb server/MQ security so disabled by default
  - 9.0.2:
    - Integrated into mqweb server and MQ security, enabled by default
    - Contains CRUD for queues and the ability to display queue status
    - Supported on MQ Appliance
  - 9.0.3:
    - Support for subset of DIS QMSTATUS on all platforms including z/OS

- **Same function on Distributed and z/OS**
  - Some minor differences

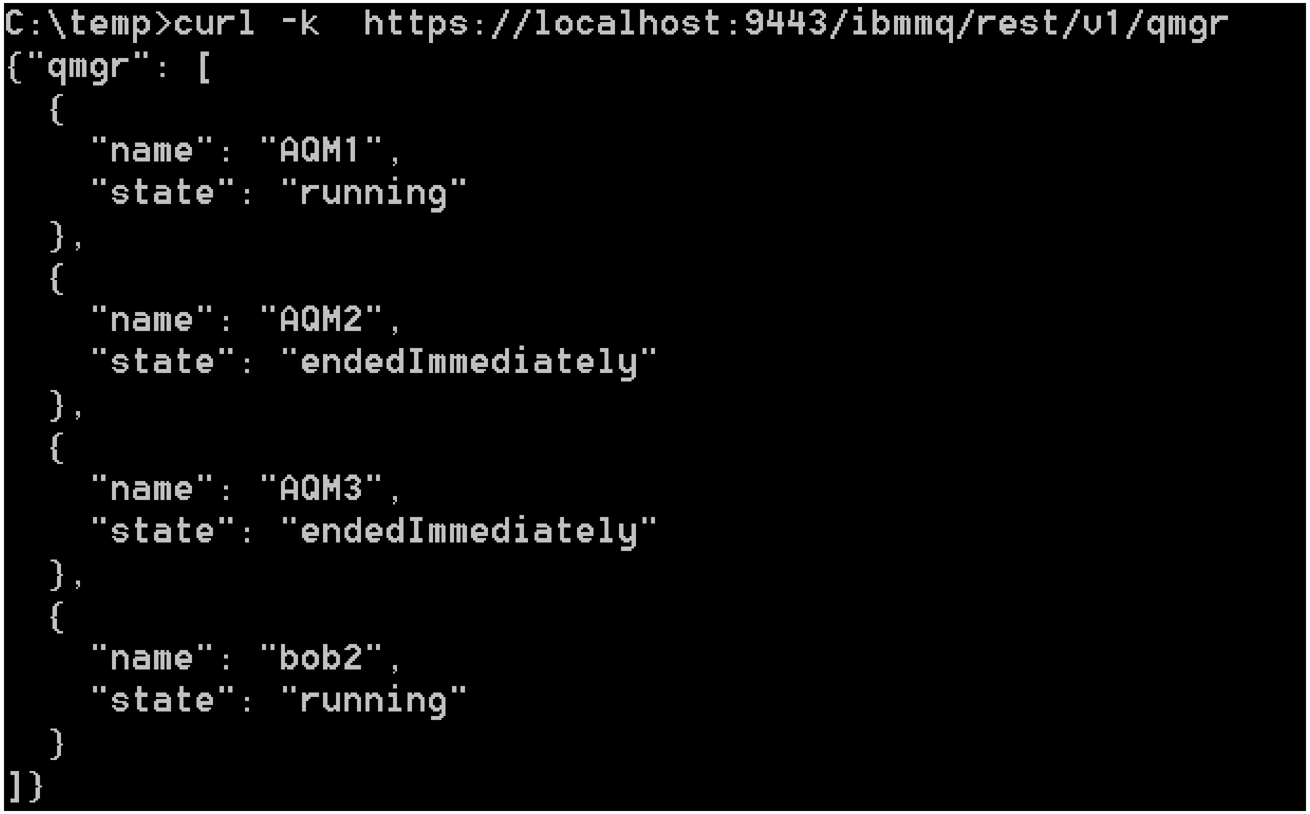
- **Restricted to the installation associated with the MQ install**
  - On z/OS queue managers must be at the same CD level

V9.0.1

GET /ibmmq/rest/v1/qmgr (dspmq)

- **Ability to list queue managers associated with installation**

- **Example below uses curl to list all queue managers**
  - -k flag tells it to ignore the fact that a self-signed certificate is being used on the mqweb server, you don’t want to be doing this in production!



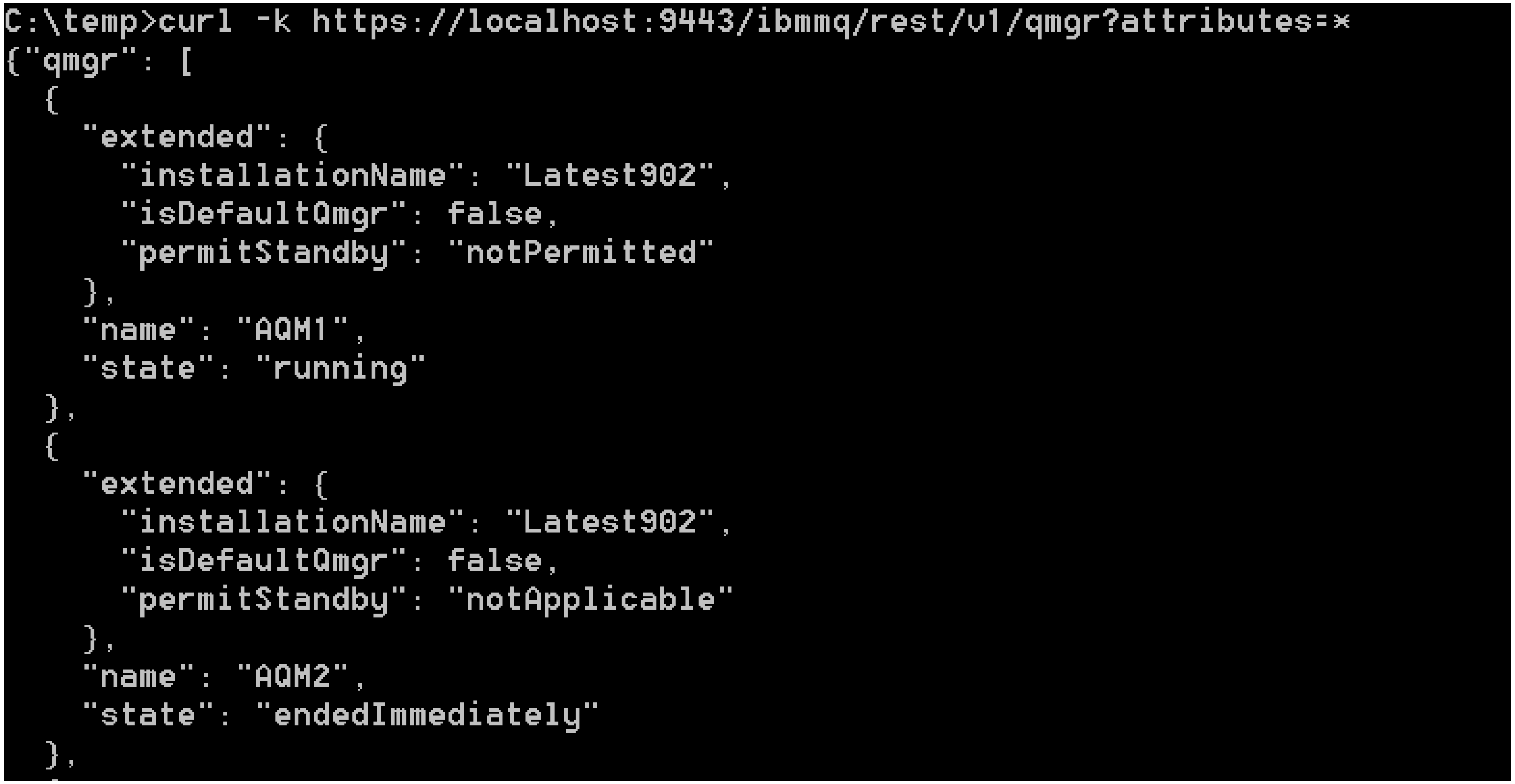
```
C:\temp>curl -k  https://localhost:9443/ibmmq/rest/v1/qmgr
{"qmgr": [
  {
    "name": "AQM1",
    "state": "running"
  },
  {
    "name": "AQM2",
    "state": "endedImmediately"
  },
  {
    "name": "AQM3",
    "state": "endedImmediately"
  },
  {
    "name": "bob2",
    "state": "running"
  }
]}
```

V9.0.1

GET /ibmmq/rest/v1/qmgr (dspmq)

- **Can get information on just a specific queue manager**
  - GET /ibmmq/rest/v1/qmgr/{qmgrName}

- **Can request additional attributes too, or just a sub-set**
  - GET /ibmmq/rest/v1/qmgr?attributes=*

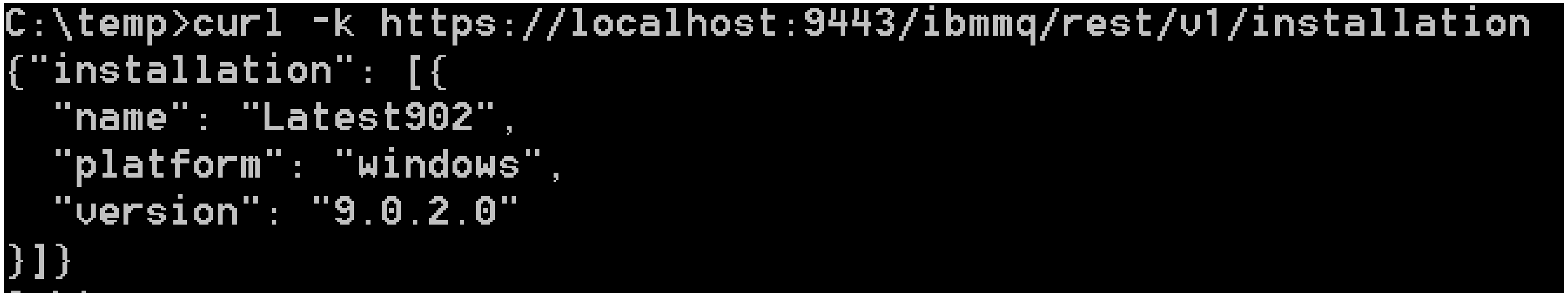


```
C:\temp>curl -k https://localhost:9443/ibmmq/rest/v1/qmgr?attributes=*
{"qmgr": [
  {
    "extended": {
      "installationName": "Latest902",
      "isDefaultQmgr": false,
      "permitStandby": "notPermitted"
    },
    "name": "AQM1",
    "state": "running"
  },
  {
    "extended": {
      "installationName": "Latest902",
      "isDefaultQmgr": false,
      "permitStandby": "notApplicable"
    },
    "name": "AQM2",
    "state": "endedImmediately"
  },
```

V9.0.1

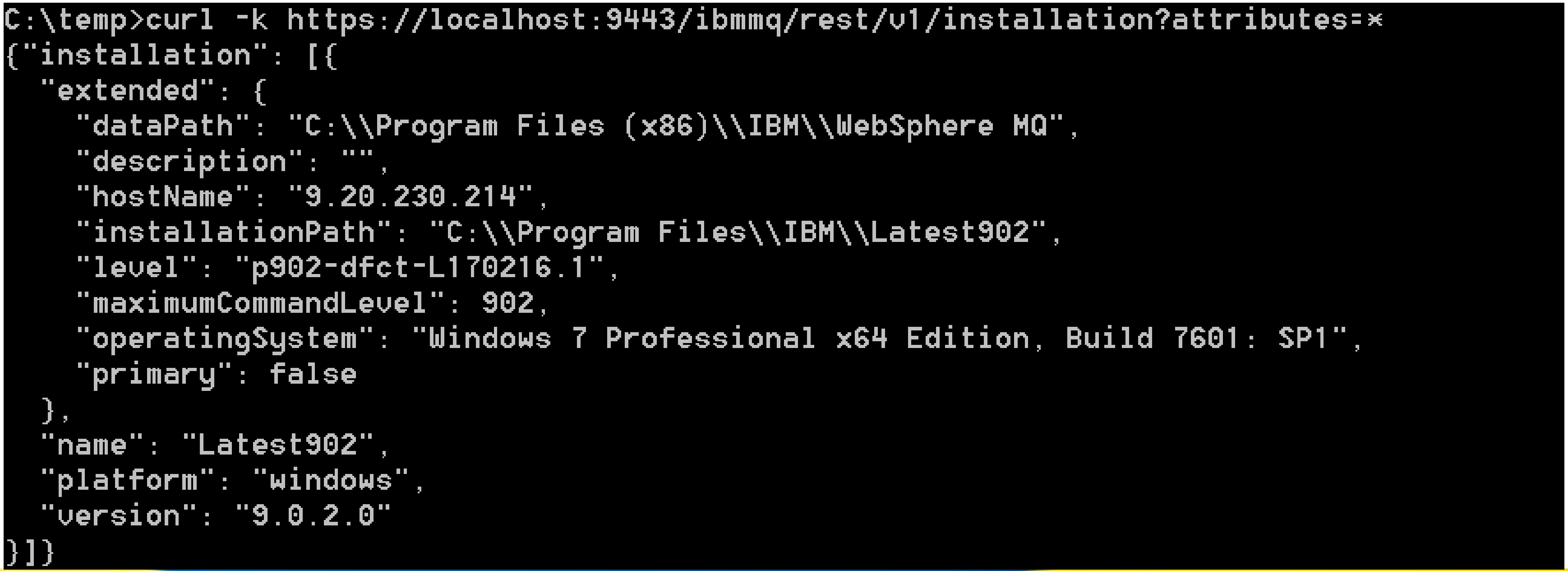
GET /ibmmq/rest/v1/installation (dspmqver)

- **Basic display**



```
C:\temp>curl -k https://localhost:9443/ibmmq/rest/v1/installation
{"installation": [{
  "name": "Latest902",
  "platform": "windows",
  "version": "9.0.2.0"
}]}
```

- **All attributes**



```
C:\temp>curl -k https://localhost:9443/ibmmq/rest/v1/installation?attributes=*
{"installation": [{
  "extended": {
    "dataPath": "C:\\Program Files (x86)\\IBM\\WebSphere MQ",
    "description": "",
    "hostName": "9.20.230.214",
    "installationPath": "C:\\Program Files\\IBM\\Latest902",
    "level": "p902-dfct-L170216.1",
    "maximumCommandLevel": 902,
    "operatingSystem": "Windows 7 Professional x64 Edition, Build 7601: SP1",
    "primary": false
  },
  "name": "Latest902",
  "platform": "windows",
  "version": "9.0.2.0"
}]}
```

V9.0.2

Queues…

- **DEFINE Q***
  - POST to /ibmmq/rest/v1/qmgr/{qmgrName}/queue

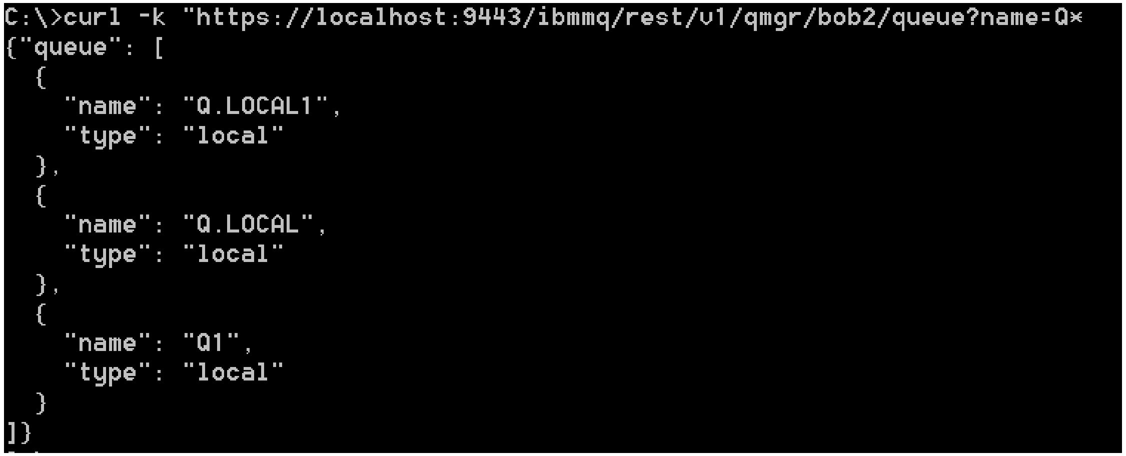
Sending JSON payload

curl -k -X POST -H "Content-Type: application/json" -d "{\"name\":\"Q1\"}" https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue

Queue manager name

Queue definition, very simple in this case

- **DISPLAY Q***
  - GET to /ibmmq/rest/v1/qmgr/{qmgrName}/queue/{queueName}



```
C:\>curl -k "https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue?name=Q*
{"queue": [
  {
    "name": "Q.LOCAL1",
    "type": "local"
  },
  {
    "name": "Q.LOCAL",
    "type": "local"
  },
  {
    "name": "Q1",
    "type": "local"
  }
]}
```

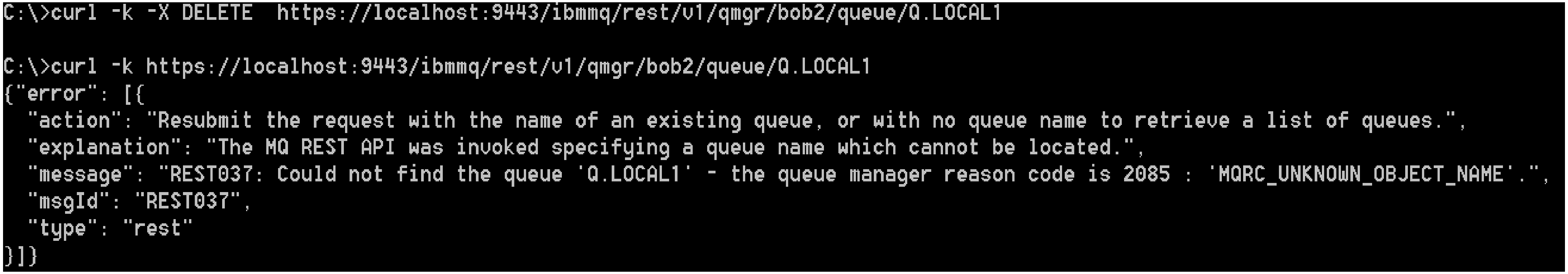
V9.0.2

Queues…

- **ALTER Q***
  - PATCH to /ibmmq/rest/v1/qmgr/{qmgrName}/queue/{queueName}
  - E.g: the following will PUT inhibit Q.LOCAL1

```
curl -k -X PATCH -H "Content-Type: application/json" -d "{\"general\":{\"inhibitPut\": true}}"
https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue/Q.LOCAL1
```

- **DELETE Q***
  - DELETE to /ibmmq/rest/v1/qmgr/{qmgrName}/queue/{queueName}



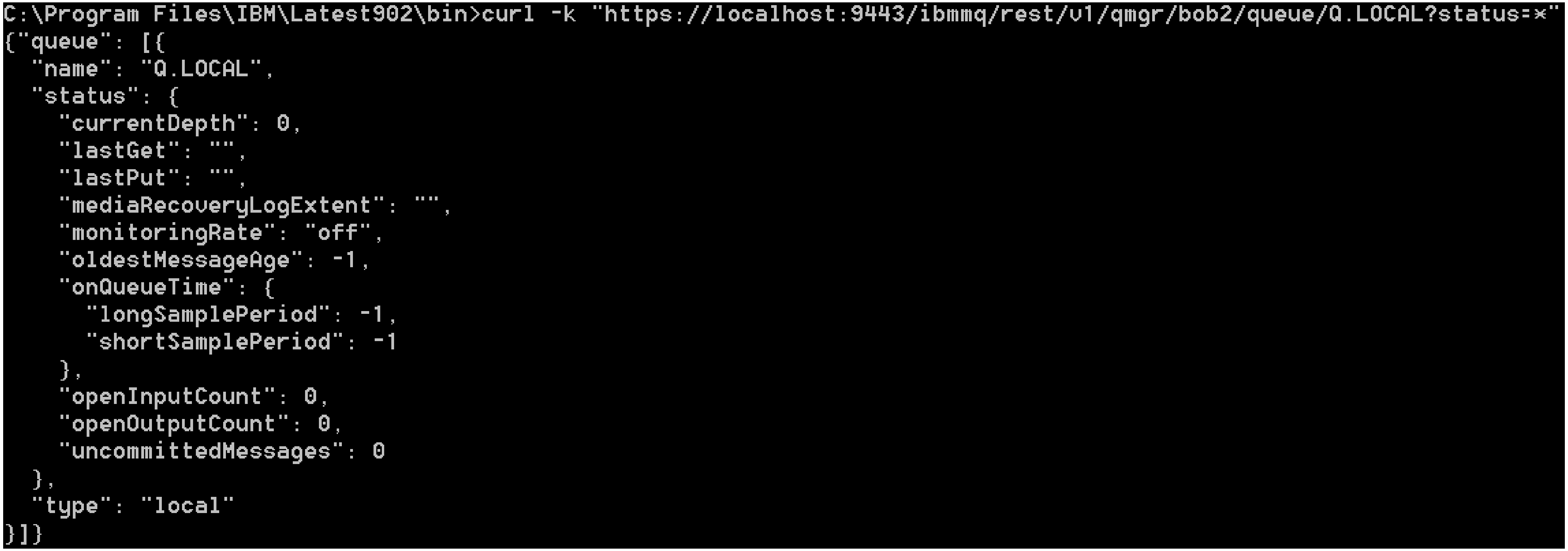
```
C:\>curl -k -X DELETE  https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue/Q.LOCAL1

C:\>curl -k https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue/Q.LOCAL1
{"error": [{
  "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
  "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
  "message": "REST037: Could not find the queue 'Q.LOCAL1' - the queue manager reason code is 2085 : 'MQRC_UNKNOWN_OBJECT_NAME'.",
  "msgId": "REST037",
  "type": "rest"
}]}
```

V9.0.2

Queues…

- **Also possible to issue DISPLAY QSTATUS**
  - GET to /ibmmq/rest/v1/qmgr/{qmgrName}/queue/{queueName}?status=*&applicationHandle=*
  - So you can get both the queue definition and its status at the same time!

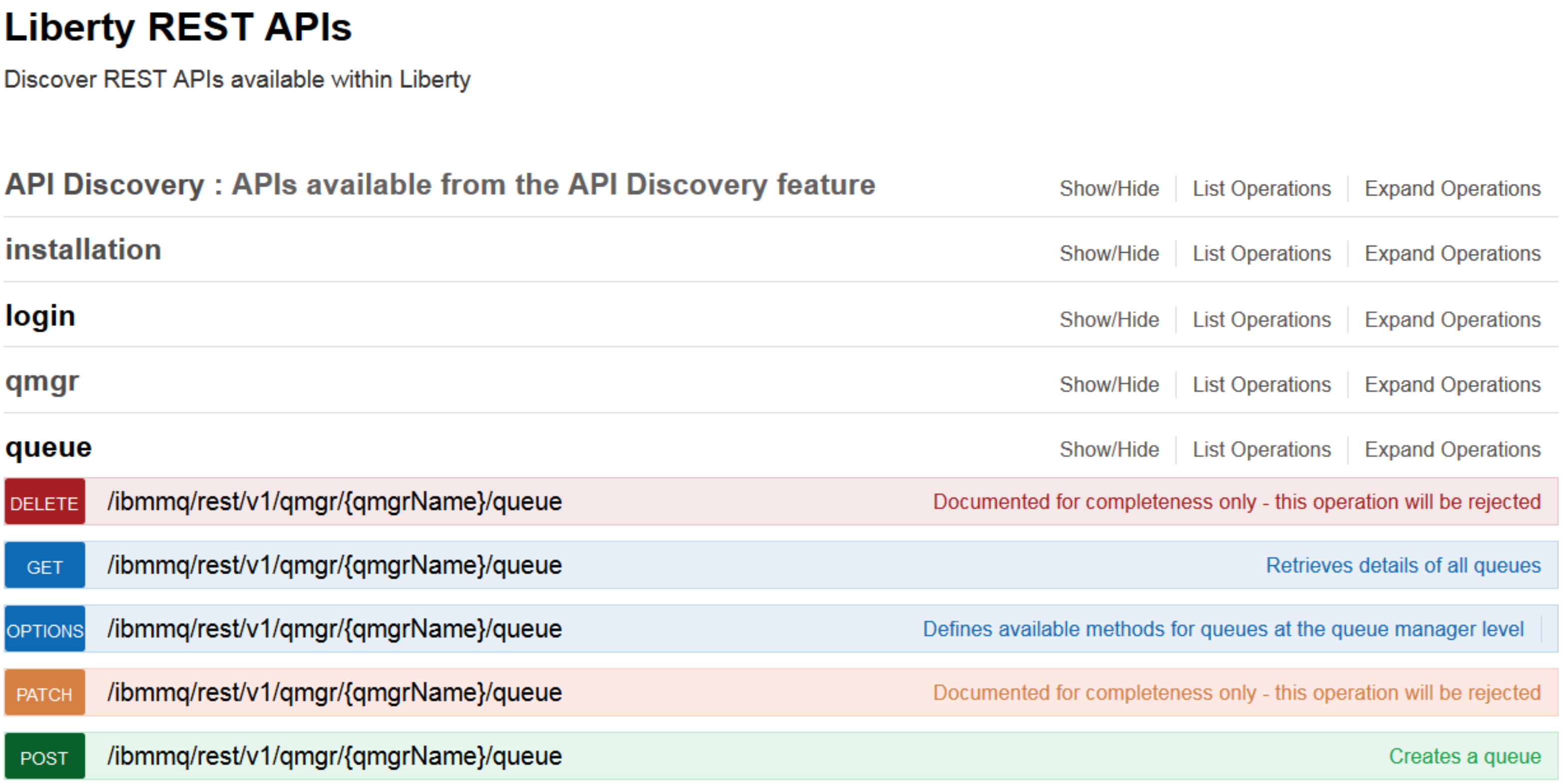


```
C:\Program Files\IBM\Latest902\bin>curl -k "https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue/Q.LOCAL?status=*"
{"queue": [{
  "name": "Q.LOCAL",
  "status": {
    "currentDepth": 0,
    "lastGet": "",
    "lastPut": "",
    "mediaRecoveryLogExtent": "",
    "monitoringRate": "off",
    "oldestMessageAge": -1,
    "onQueueTime": {
      "longSamplePeriod": -1,
      "shortSamplePeriod": -1
    },
    "openInputCount": 0,
    "openOutputCount": 0,
    "uncommittedMessages": 0
  },
  "type": "local"
}]}
```

V9.0.1

API discovery

- Want to find out what is available in the MQ REST API, and don’t want to read the KC?
- Then try out API discovery!
- Function in WLP that describes the MQ REST API using Swagger
- Makes it easier to see what is there, and try it out



Liberty REST APIs

Discover REST APIs available within Liberty

API Discovery : APIs available from the API Discovery feature — Show/Hide | List Operations | Expand Operations

installation — Show/Hide | List Operations | Expand Operations

login — Show/Hide | List Operations | Expand Operations

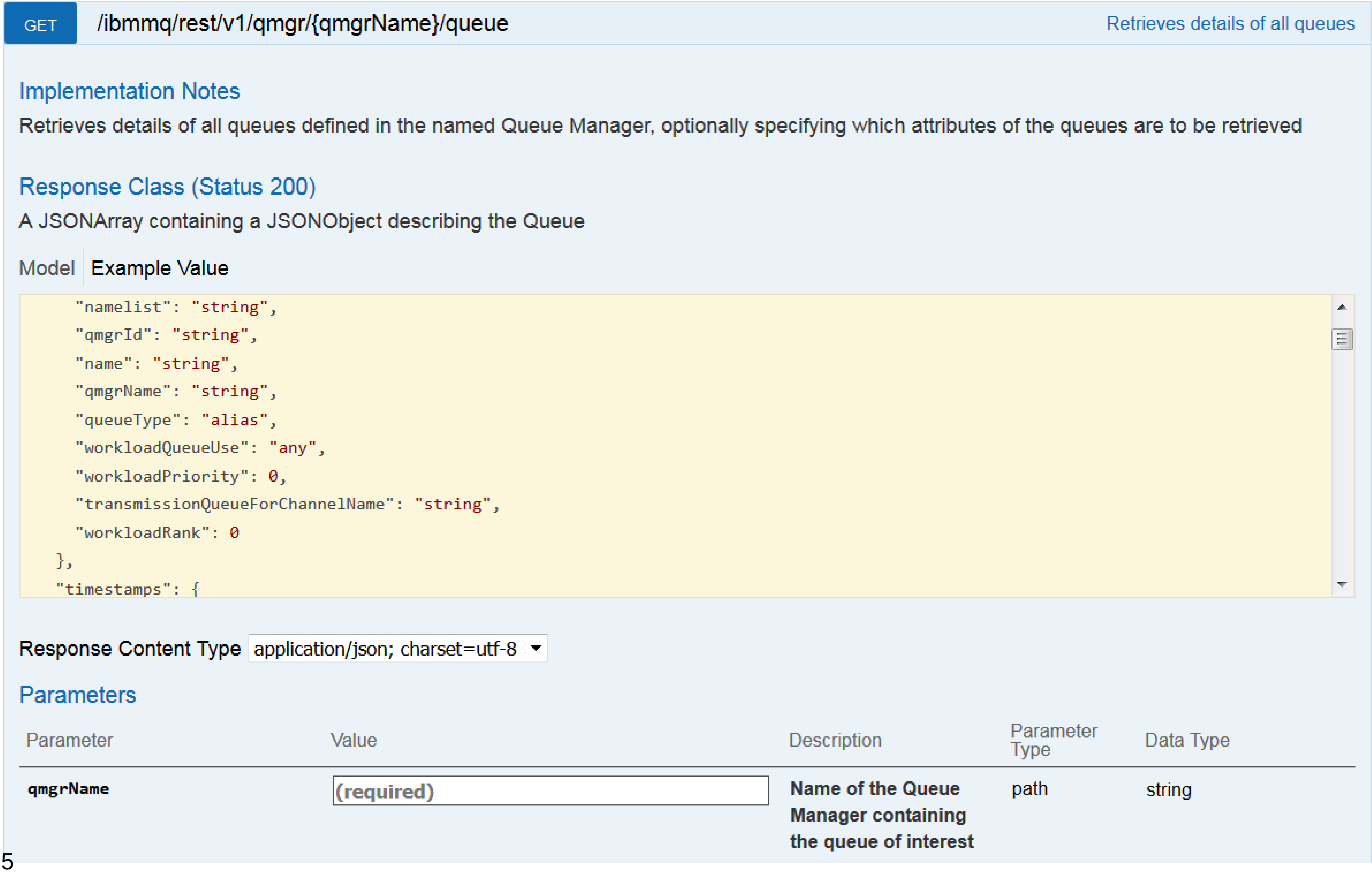
qmgr — Show/Hide | List Operations | Expand Operations

queue — Show/Hide | List Operations | Expand Operations

DELETE	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Documented for completeness only - this operation will be rejected
GET	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Retrieves details of all queues
OPTIONS	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Defines available methods for queues at the queue manager level
PATCH	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Documented for completeness only - this operation will be rejected
POST	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Creates a queue

V9.0.1

API discovery



/ibmmq/rest/v1/qmgr/{qmgrName}/queue

Retrieves details of all queues

Implementation Notes

Retrieves details of all queues defined in the named Queue Manager, optionally specifying which attributes of the queues are to be retrieved

Response Class (Status 200)

A JSONArray containing a JSONObject describing the Queue

```
      "namelist": "string",
      "qmgrId": "string",
      "name": "string",
      "qmgrName": "string",
      "queueType": "alias",
      "workloadQueueUse": "any",
      "workloadPriority": 0,
      "transmissionQueueForChannelName": "string",
      "workloadRank": 0
    },
    "timestamps": {
```

Response Content Type: application/json; charset=utf-8

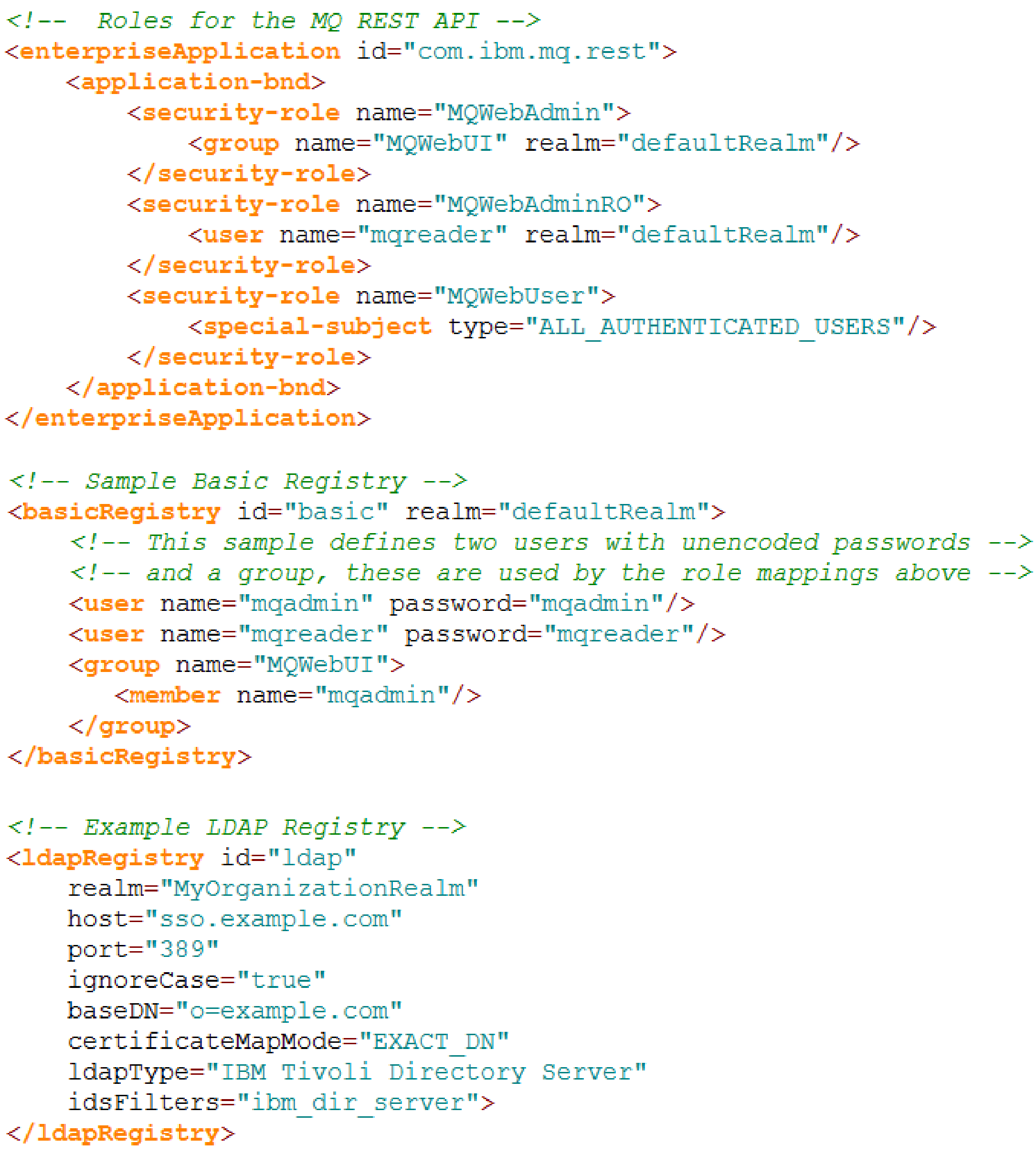
Parameters

Parameter	Value	Description	Parameter Type	Data Type
qmgrName	(required)	Name of the Queue Manager containing the queue of interest	path	string

V9.0.2

REST API security

- **Role based access control. Need to be a member of at least one role**
  - MQWebAdmin
  - MQWebAdminRO
  - MQWebUser
- **User and groups defined in a registry**
  - Basic
  - LDAP
  - SAF (on z/OS)
- **REST is locked down by default, need to do some configuring**
  - Samples provided to make this simpler



```
<!--  Roles for the MQ REST API -->
<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQWebUI" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="mqreader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <special-subject type="ALL_AUTHENTICATED_USERS"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<!-- Sample Basic Registry -->
<basicRegistry id="basic" realm="defaultRealm">
    <!-- This sample defines two users with unencoded passwords -->
    <!-- and a group, these are used by the role mappings above -->
    <user name="mqadmin" password="mqadmin"/>
    <user name="mqreader" password="mqreader"/>
    <group name="MQWebUI">
       <member name="mqadmin"/>
    </group>
</basicRegistry>

<!-- Example LDAP Registry -->
<ldapRegistry id="ldap"
    realm="MyOrganizationRealm"
    host="sso.example.com"
    port="389"
    ignoreCase="true"
    baseDN="o=example.com"
    certificateMapMode="EXACT_DN"
    ldapType="IBM Tivoli Directory Server"
    idsFilters="ibm_dir_server">
</ldapRegistry>
```

V9.0.2

REST API authentication

- **Token based**
  - User logs in once with user id and password and then gets a cookie which is used for subsequent requests

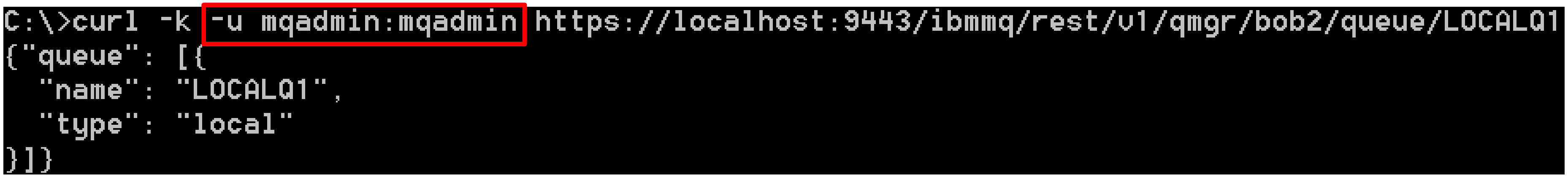
```
curl -k -X POST -H "Content-Type: application/json"
    -d "{\"username\":\"mqadmin\",\"password\":\"mqadmin\"}"
    https://localhost:9443/ibmmq/rest/v1/login -c c:\temp\cookiejar.txt
```

User id and password provided as JSON payload

Cookie stored for use on next request

  - DELETE to the login URL logs out

- **Or HTTP basic authentication**
  - User id and password provided as an encoded header, must be set for each request



```
C:\>curl -k -u mqadmin:mqadmin https://localhost:9443/ibmmq/rest/v1/qmgr/bob2/queue/LOCALQ1
{"queue": [{
  "name": "LOCALQ1",
  "type": "local"
}]}
```

V9.0.2

REST API authentication

- **Or use a client certificate**
  - Must be provided with each call to the REST API
  - Distinguished name from certificate is mapped to user in configured user registry

- **CORS support enabled for web-browser based environments**
  - Provides a whitelist of origins (URLs) which can invoke the REST API

V9.0.1

The MQ Console

V9.0.1

MQ Console

- **Browser based interface for administering and managing MQ**
  - No client side install needed
  - Originally available in MQ Appliance only

- **As of 9.0.1 a common capability across appliance and software MQ**
  - Re-engineered on AngularJS so different implementation than on 8.0.0.* appliance
  - Functional parity with MQ Console in 8.0.0.* appliance

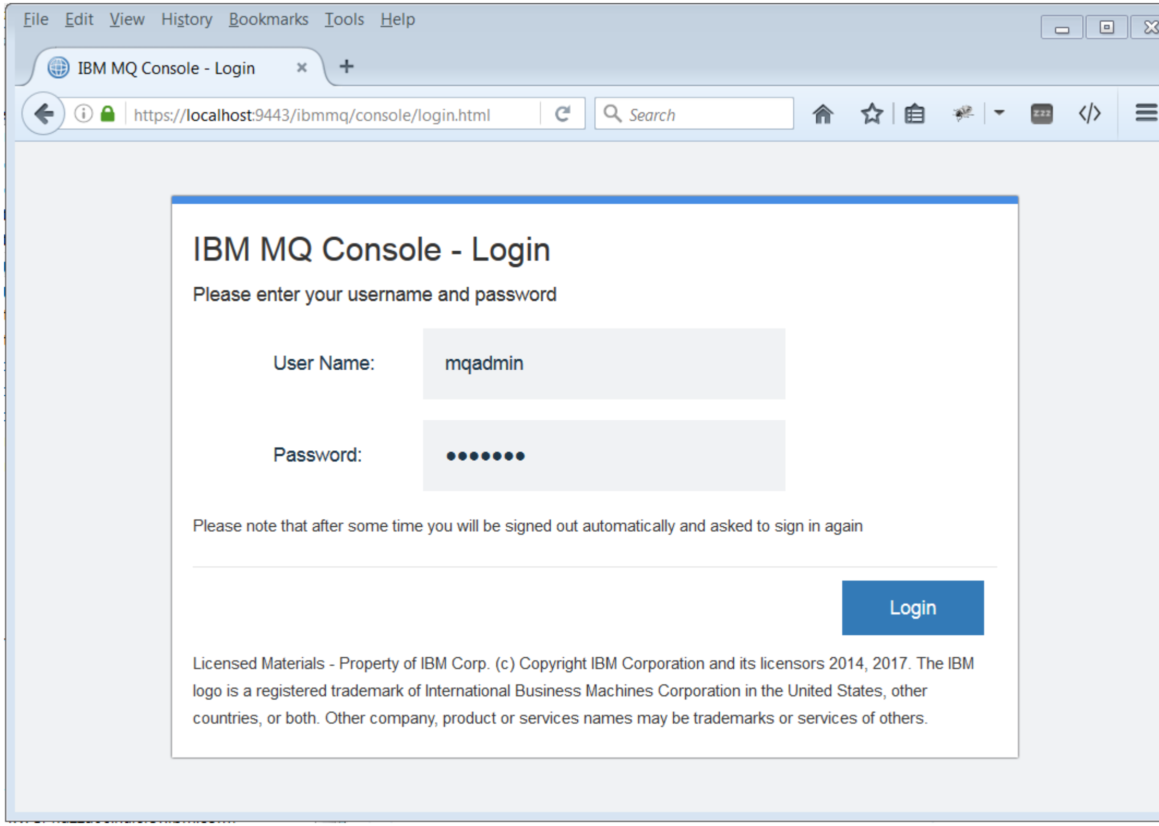
- **Some capabilities not available on z/OS**
  - Can’t create/delete/start/stop queue managers, etc

- **Can only interact with queue managers running in the same installation**
  - On z/OS all queue managers at the same CD level

V9.0.1

MQ Console – log in

- **Point your web-browser at the MQ Console and log in**
  - With a user id and password
  - With a client certificate
- **Log in credentials validated via user registry configured in the mqweb server**
  - Like the REST API
- **Access determined by role**
  - Same role names as REST API
  - But in a different name space so REST users don't need to have same access as MQ Console users


File Edit View History Bookmarks Tools Help
IBM MQ Console - Login
https://localhost:9443/ibmmq/console/login.html
Search

IBM MQ Console - Login

Please enter your username and password

User Name: mqadmin

Password: •••••••

Please note that after some time you will be signed out automatically and asked to sign in again

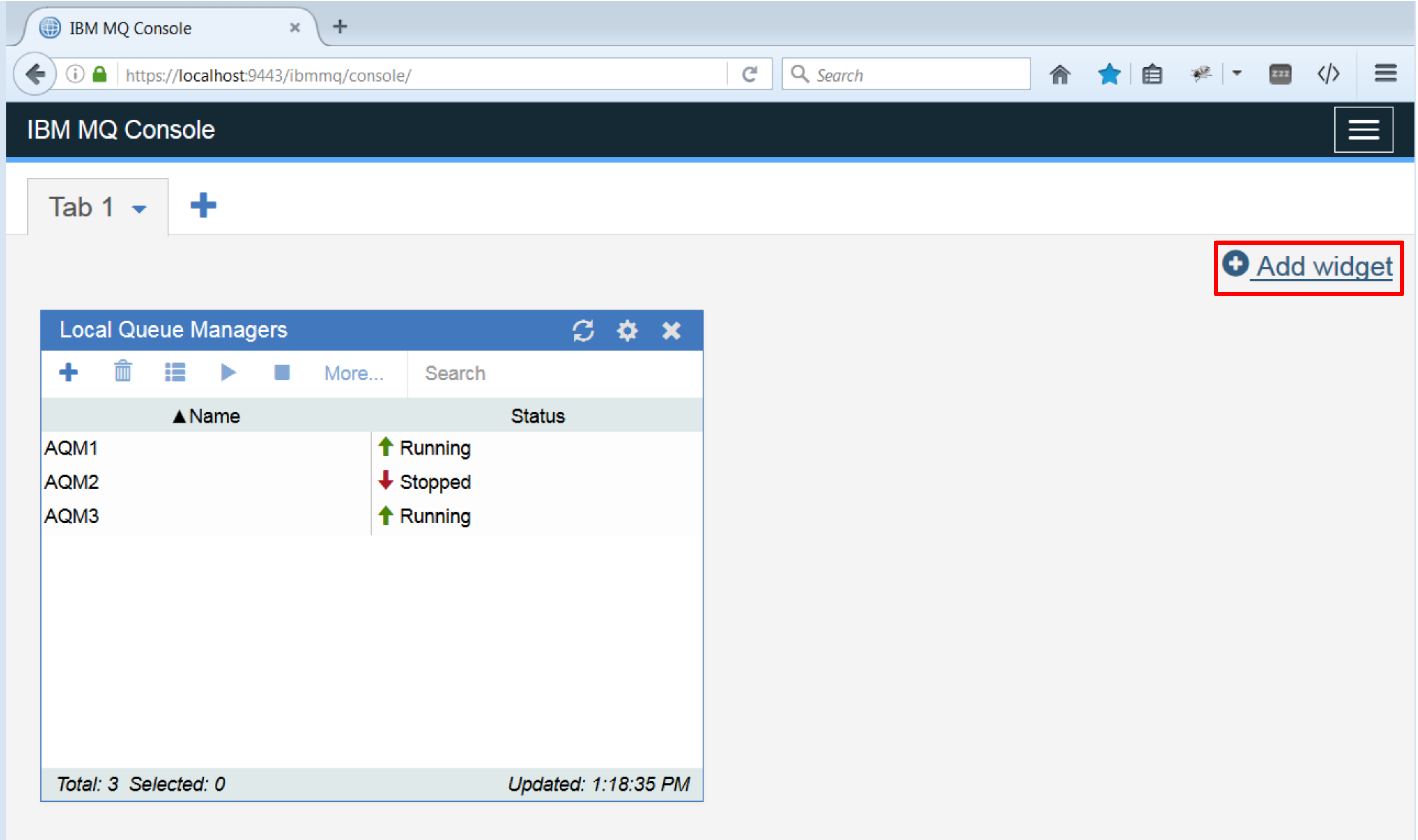
Login

Licensed Materials - Property of IBM Corp. (c) Copyright IBM Corporation and its licensors 2014, 2017. The IBM logo is a registered trademark of International Business Machines Corporation in the United States, other countries, or both. Other company, product or services names may be trademarks or services of others.

V9.0.1

MQ Console – add widgets

- Console dashboard consists of a number of widgets, each widget shows information for a particular set of MQ objects: queue managers, queues, etc



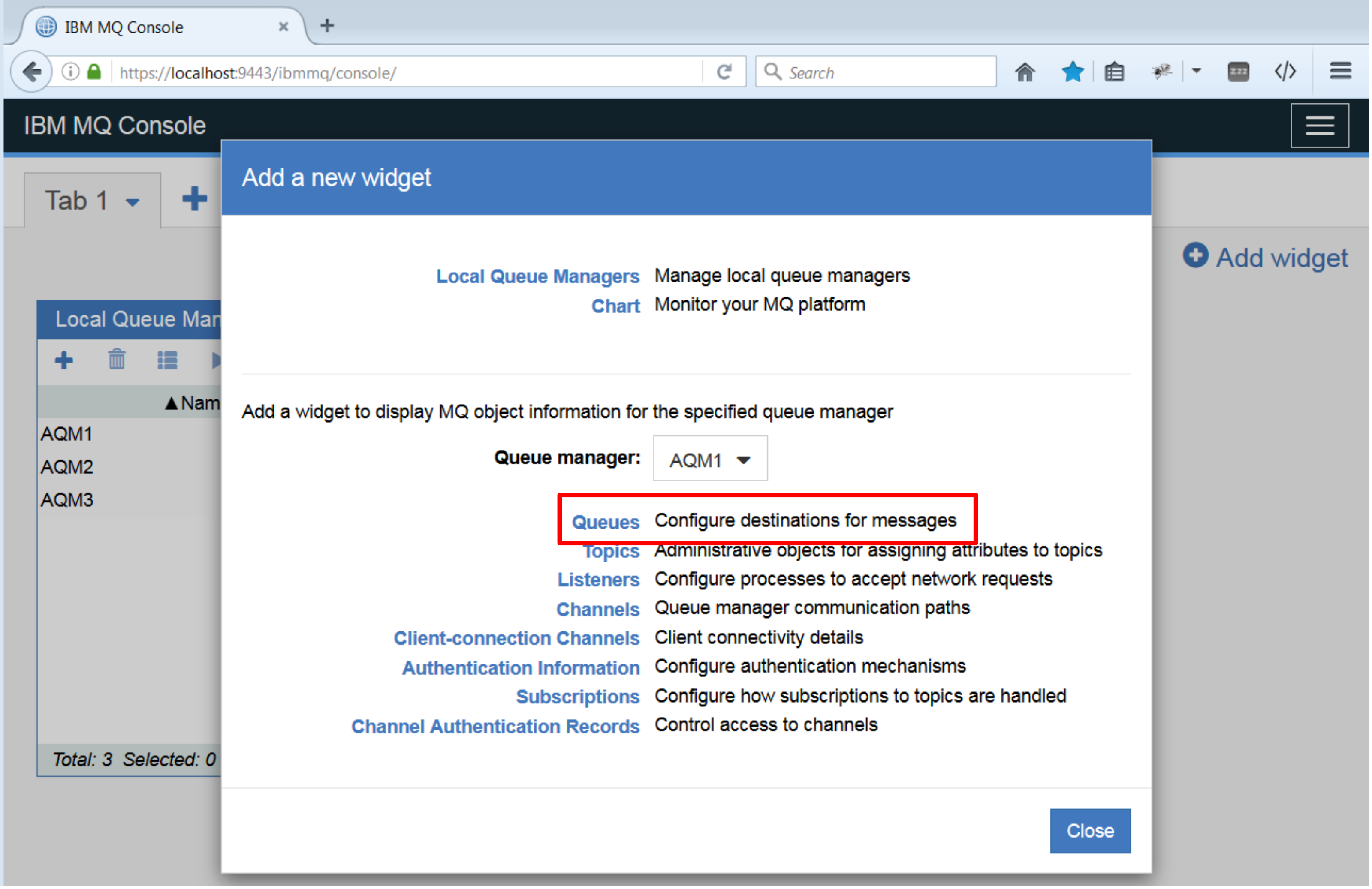
▲Name	Status
AQM1	Running
AQM2	Stopped
AQM3	Running

Total: 3 Selected: 0 *Updated: 1:18:35 PM*

V9.0.1

MQ Console – add widgets

- Console dashboard consists of a number of widgets, each widget shows information for a particular set of MQ objects: queue managers, queues, etc



Add a new widget

Local Queue Managers	Manage local queue managers
Chart	Monitor your MQ platform

Add a widget to display MQ object information for the specified queue manager

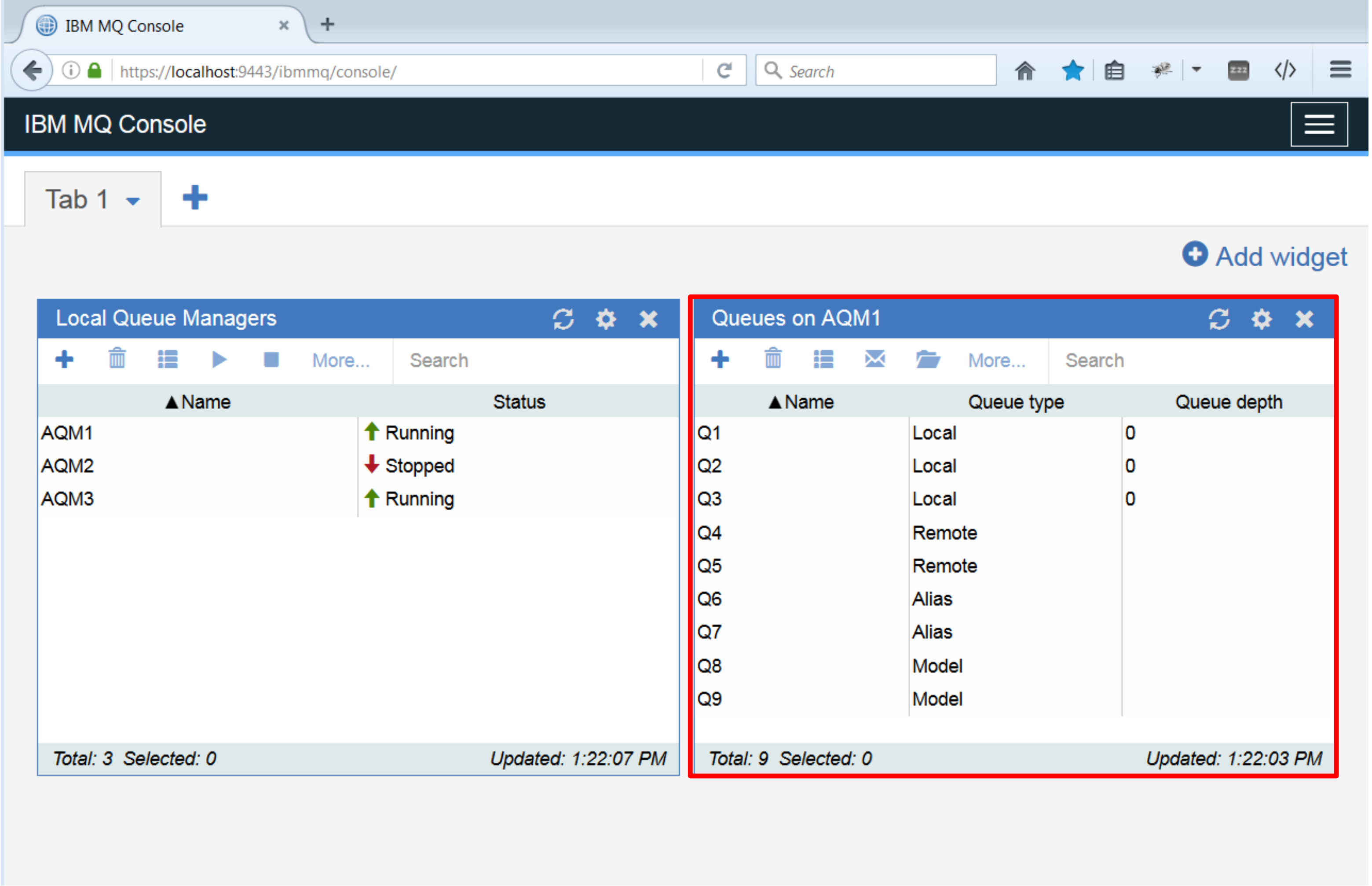
Queue manager: AQM1

Queues	Configure destinations for messages
Topics	Administrative objects for assigning attributes to topics
Listeners	Configure processes to accept network requests
Channels	Queue manager communication paths
Client-connection Channels	Client connectivity details
Authentication Information	Configure authentication mechanisms
Subscriptions	Configure how subscriptions to topics are handled
Channel Authentication Records	Control access to channels

V9.0.1

MQ Console – add widgets

- Console dashboard consists of a number of widgets, each widget shows information for a particular set of MQ objects: queue managers, queues, etc



Local Queue Managers

▲Name	Status
AQM1	↑ Running
AQM2	↓ Stopped
AQM3	↑ Running

Total: 3 Selected: 0 Updated: 1:22:07 PM

Queues on AQM1

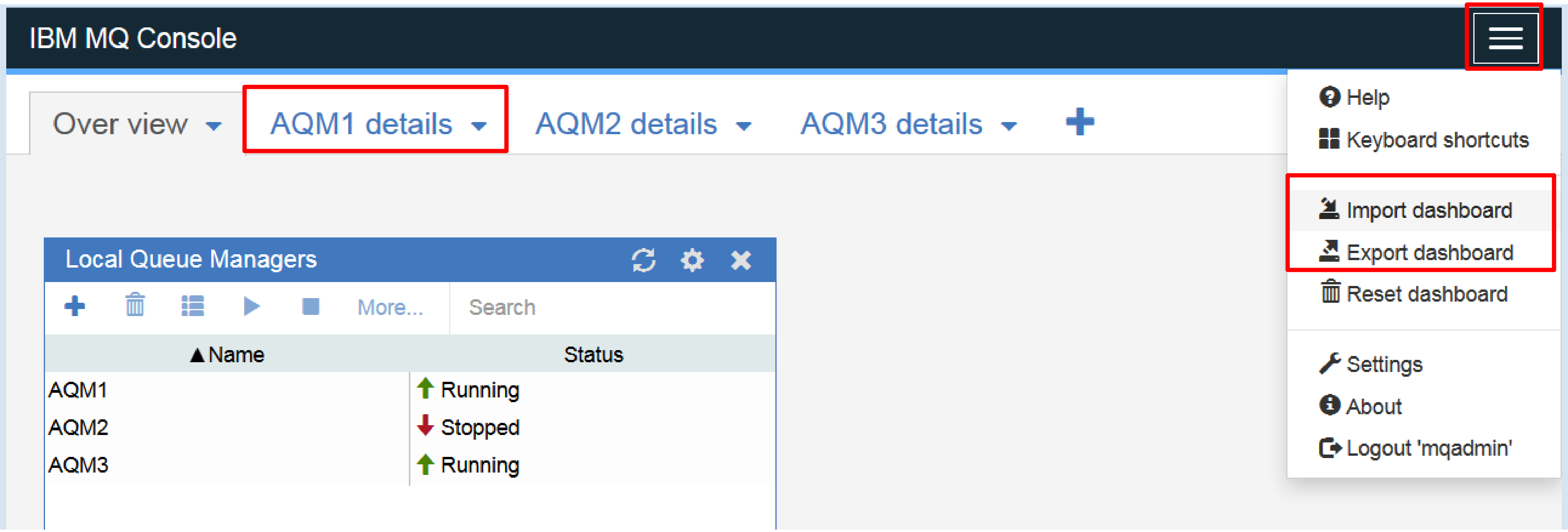
▲Name	Queue type	Queue depth
Q1	Local	0
Q2	Local	0
Q3	Local	0
Q4	Remote	
Q5	Remote	
Q6	Alias	
Q7	Alias	
Q8	Model	
Q9	Model	

Total: 9 Selected: 0 Updated: 1:22:03 PM

V9.0.1

MQ Console – layout

- Can use multiple tabs to help manage content
- Each user can lay out their dashboard according to their needs
- Can export dashboard to share layout with others

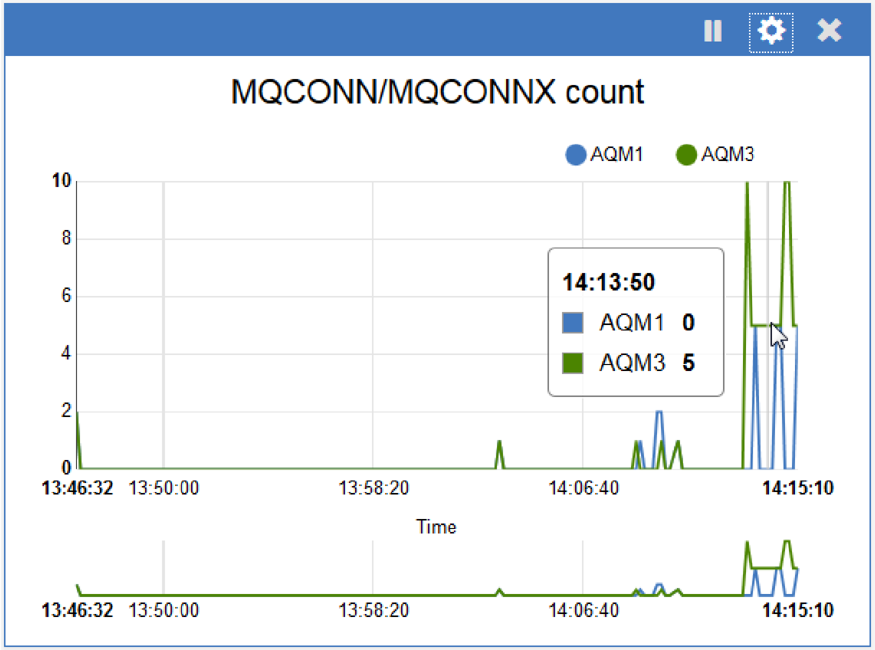


▲Name	Status
AQM1	Running
AQM2	Stopped
AQM3	Running

V9.0.1

MQ Console - manage

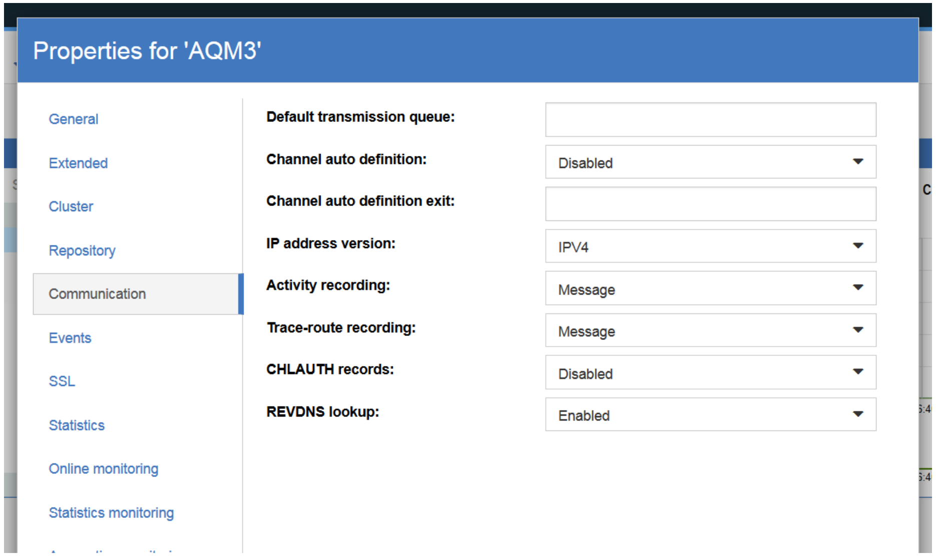
- **Monitor your MQ queue managers using charts generated from statistics information published to system topics**
  - System topics added in 9.0.0 on distributed



- **Display and alter objects using the properties editor**

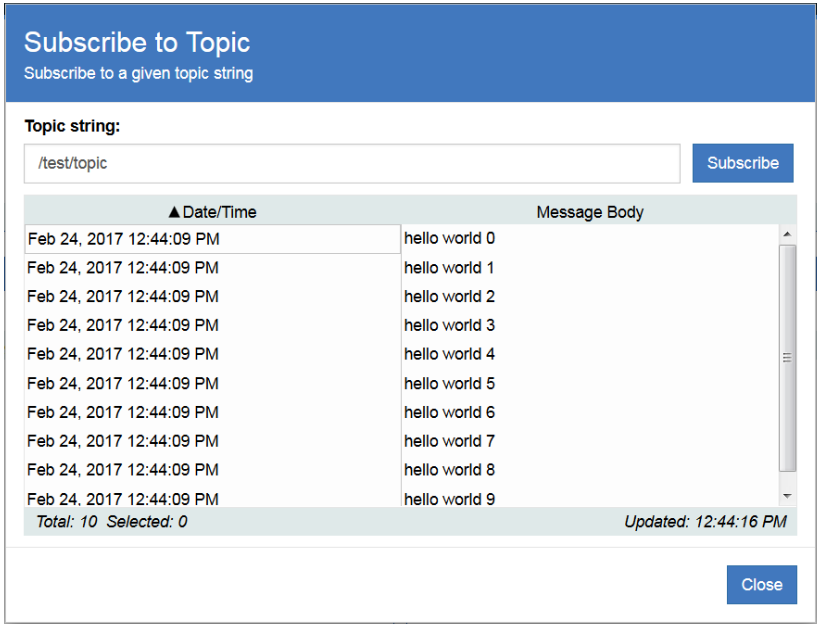
- **Browse and send messages**

- **Provides a sub-set of MQ Explorer function**



Properties for 'AQM3'

Default transmission queue:	
Channel auto definition:	Disabled
Channel auto definition exit:	
IP address version:	IPV4
Activity recording:	Message
Trace-route recording:	Message
CHLAUTH records:	Disabled
REVDNS lookup:	Enabled



Subscribe to Topic

Subscribe to a given topic string

Topic string: /test/topic

▲Date/Time	Message Body
Feb 24, 2017 12:44:09 PM	hello world 0
Feb 24, 2017 12:44:09 PM	hello world 1
Feb 24, 2017 12:44:09 PM	hello world 2
Feb 24, 2017 12:44:09 PM	hello world 3
Feb 24, 2017 12:44:09 PM	hello world 4
Feb 24, 2017 12:44:09 PM	hello world 5
Feb 24, 2017 12:44:09 PM	hello world 6
Feb 24, 2017 12:44:09 PM	hello world 7
Feb 24, 2017 12:44:09 PM	hello world 8
Feb 24, 2017 12:44:09 PM	hello world 9

Total: 10 Selected: 0 *Updated: 12:44:16 PM*

Summary

- **Existing capabilities**
- **What’s new?**
  - The mqweb server
  - The MQ REST API
  - The MQ Console

Questions?