

IBM MQ Lab Tour – Paris – 26 September 2017



IBM MQ

# IBM MQ Console and REST

Jamie Squibb  
[jamie\\_squibb@uk.ibm.com](mailto:jamie_squibb@uk.ibm.com)

# Notices and disclaimers

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

## **U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

## **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

# Notices and disclaimers (continued)

---

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

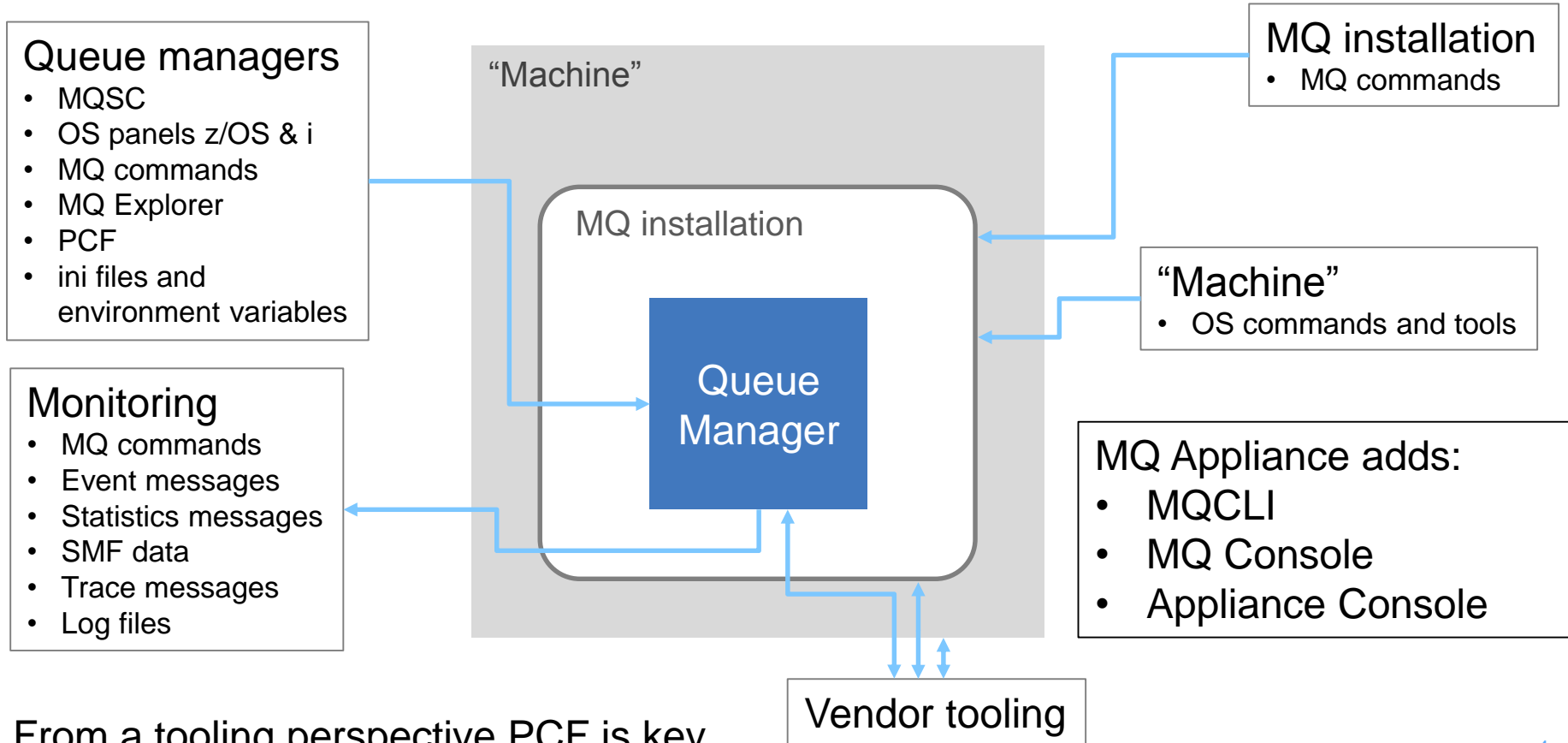
IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

# Agenda

---

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

# Administering software MQ



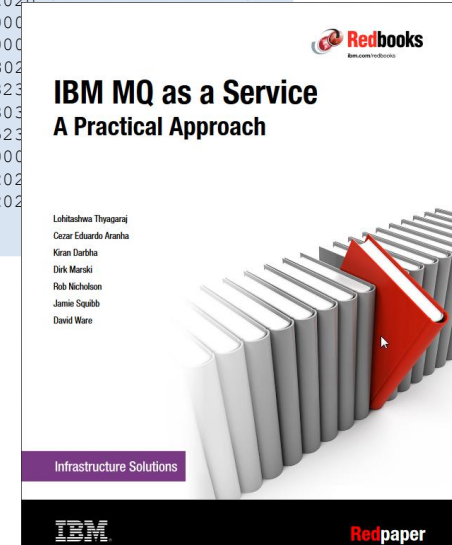
# Why we need more

- 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
- 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

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

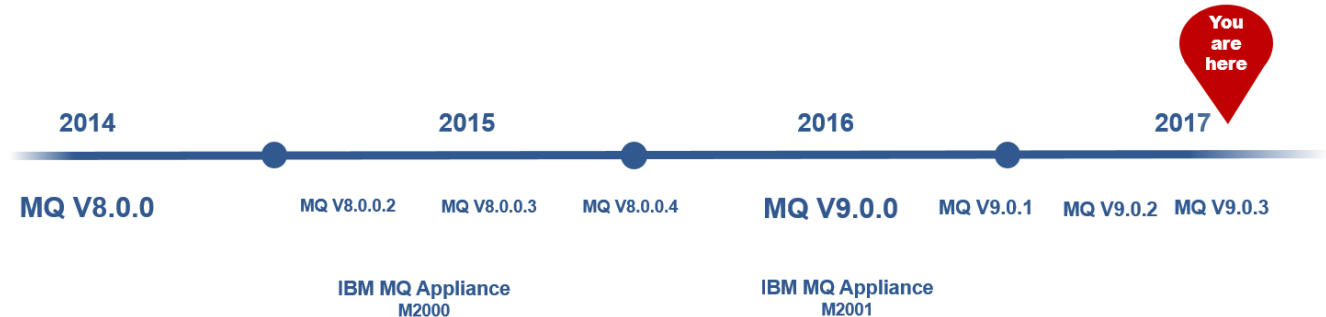
length - 724 of 724 bytes

00000000: 080A 4103 0000 0000 5744 5220 0200 0000 '..A....WDR ....\
00000010: 8800 0000 6700 0000 514D 4752 315F 3230 '^...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 2020 0000 0000 0000 '                \
00000080: 58CA 0000 0000 0000 0000 0000 0000 0000 '                \
00000090: 644E 4656 2116 4656 3230 3135 2D31 302D '                \
000000A0: 3239 2020 0000 0000 3039 2E34 312E 3239 '                \
000000B0: 0100 0000 4D51 4D4D 0000 0000 3038 3038 '                \
000000C0: 3030 3034 0000 0000 434C 5553 5445 5239 '                \
000000D0: 2E51 4D47 5231 2020 2020 2020 0B00 0000 '                \
000000E0: 0800 0000 0200 0000 2020 2020 2020 2020 '                \
000000F0: 2020 2020 2020 2020 2020 2020 2020 2020 '                \
```



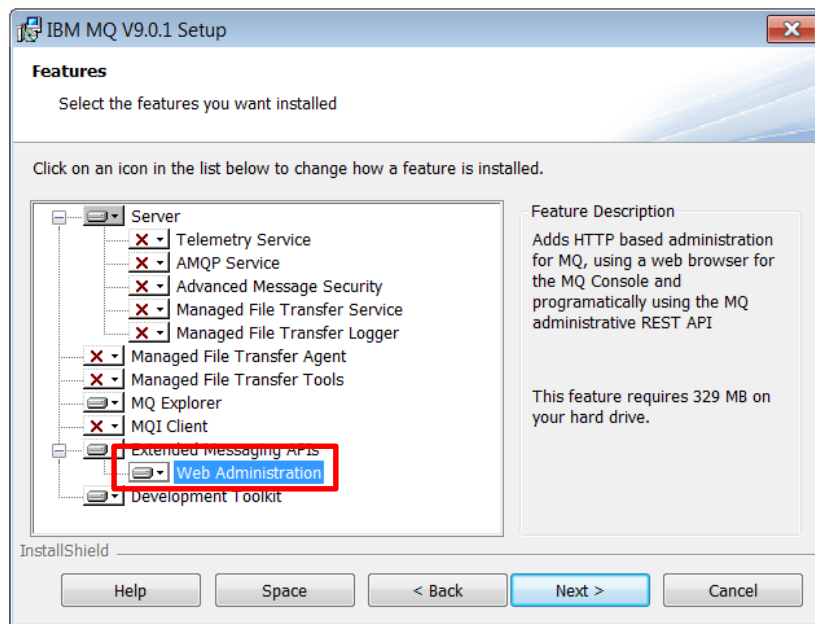
# Overview

- 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



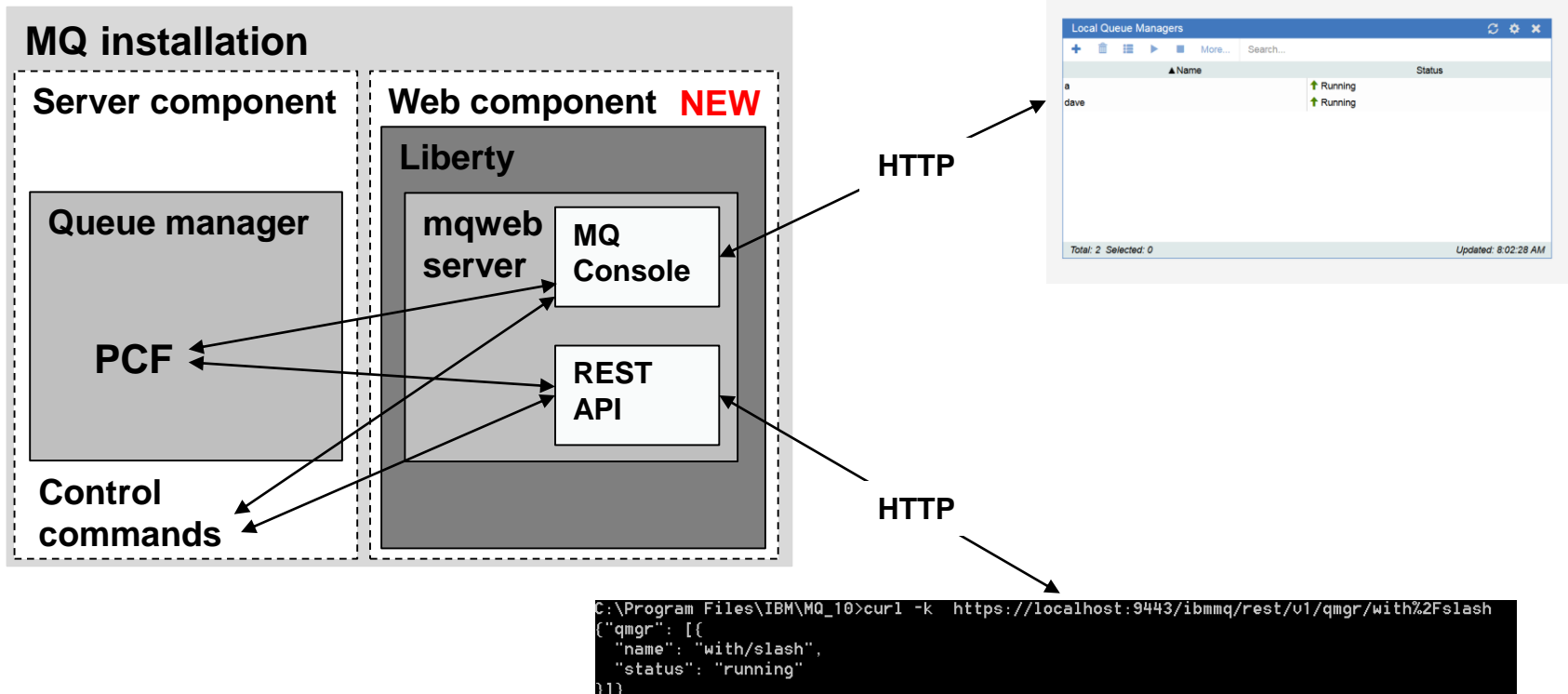
# Web component

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





# Perhaps a picture would help?



# 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

```
CWWKE00011: The server mqweb has been launched.
CWWKG0028A: Processing included configuration resource: C:\Program
Files\IBM\Latest902\web\mqletc\mqweb.xml
A CWWKG0028A: Processing included configuration resource: C:\Program Files (x86)\IBM\WebSphere
MQ\web\installations\Latest902\servers\mqweb\mqwebuser.xml
CWWKE00021: The kernel started after 2.493 seconds
CWWKF00071: Feature update started.
CWWKO02191: 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.
CWWKZ00181: Starting application com.ibm.mq.rest.
CWWKZ00181: Starting application com.ibm.mq.console.
SRVE01691: Loading Web Module: com.ibm.mq.rest.v1.
SRVE02501: Web Module com.ibm.mq.rest.v1 has been bound to default_host.
CWWKT00161: Web application available (default_host): https://localhost:9443/ibmmq/rest/v1/
CWWKZ00011: Application com.ibm.mq.rest started in 0.518 seconds.
SRVE01691: Loading Web Module: mqconsole.
SRVE02501: Web Module mqconsole has been bound to default_host.
CWWKT00161: Web application available (default_host): https://localhost:9443/ibmmq/console/
SRVE01691: Loading Web Module: com.ibm.mq.consoleinternal.
SRVE02501: Web Module com.ibm.mq.consoleinternal has been bound to default_host.
CWWKT00161: Web application available (default_host): https://localhost:9443/ibmmq/console/internal/
CWWKZ00011: Application com.ibm.mq.console started in 0.525 seconds.
CWWKF00121: 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].
CWWKF00081: Feature update completed in 2.095 seconds.
CWWKF00111: The server mqweb is ready to run a smarter planet.
REST023: MQ REST API level: p902-dfct-L170216.1
```

# 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>
```

# Managing mqweb server

- Distributed: 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 up all necessary variables then starts the mqweb server

```
EDIT          USER.PROCLIB(MQWEBMVL) - 01.03
Command ==>
000058 //*****
000059 //*
000060 //          PROC
000061 //*
000062 // SET INSTDIR='/u/mleming/v902betamqm/web'
000063 // SET USERDIR='/u/mleming/mq902web'
000064 //*
000065 //STEP1 EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
000066 // PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv mqweb'
000067 //WLPUDIR DD PATH='&USERDIR.'
000068 //STEPLIB DD DSN=ANTZ.MQ.V900.DFCT.OUT.SCSQANLE,DISP=SHR
000069 //          DD DSN=ANTZ.MQ.V900.DFCT.OUT.SCSQAUTH,DISP=SHR
000070 //STDOUT DD SYSOUT=*
000071 //STDERR DD SYSOUT=*
000072 //STDIN DD DUMMY
000073 //STDENV DD *
000074 JAVA_HOME=/java/java80_bit64_sr3_fp20/J8.0_64
000075 PATH=/u/mleming/v902betamqm/web/bin:/bin:/usr/sbin
000076 LIBPATH=/u/mleming/v902betamqm/java/lib
000077 //*
```

# What is REST?

---

- **RE**presentational **S**tate **T**ransfer
  - An architectural style
  - Term first coined by Roy Fielding in his PhD thesis
  - 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) that can use them
  - Originally used for serving web-pages
  - Works really well for APIs too
- Generally light-weight and relatively simple to use, much simpler than SOAP web-services
  - Has become incredibly common in recent years
- However, there are a lot 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

# 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 that 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

A nested unnamed  
object, in an array

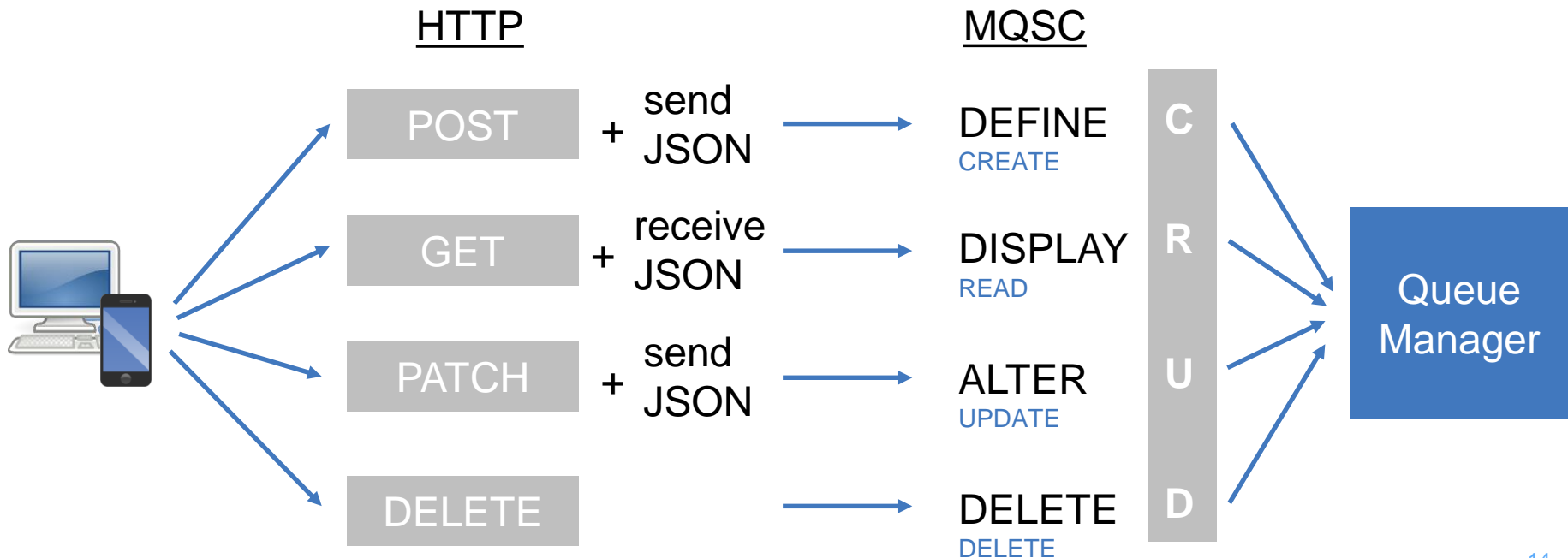
```
{  
  "qmgr": [  
    {  
      "name": "AQM1",  
      "state": "running"  
    }  
  ]  
}
```

Square bracket denotes  
JSON array

Name - value pair, where  
the value is of type string

# MQ REST API

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



# 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



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

- Ability to list queue managers associated with installation
- Example below uses curl to list all queue managers
  - The -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"
  }
]}
```

# 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"
  },
  ]
}
```

# 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"
}]}
```

# Defining 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

# Displaying queues

- 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"
  }
]}
```

# Altering 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
```

# Deleting queues

- 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"}]}
```

# Displaying queue status

- 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"}]}
```



# API discovery

- Want to find out what is available in the MQ REST API & 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](#)

GET	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Retrieves details of all queues
POST	/ibmmq/rest/v1/qmgr/{qmgrName}/queue	Creates a queue
DELETE	/ibmmq/rest/v1/qmgr/{qmgrName}/queue/{qName}	Deletes a queue
GET	/ibmmq/rest/v1/qmgr/{qmgrName}/queue/{qName}	Retrieves details of a specific queue
PATCH	/ibmmq/rest/v1/qmgr/{qmgrName}/queue/{qName}	Modifies a queue

# API discovery

GET /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

Model Example Value

```
"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	<input type="text" value="(required)"/>	Name of the Queue Manager containing the queue of interest	path	string

# 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 - you 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>
```

# REST API authentication

- Token based

- User logs in once with user id and password and 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

- DELETE to the login URL logs out

← Cookie stored for use on next request

- 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"  
    }  
  ]  
}
```

# 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
  - Not currently supported on the MQ Appliance
- CORS support enabled for web-browser based environments
  - Provides a whitelist of origins (URLs) which can invoke the REST API

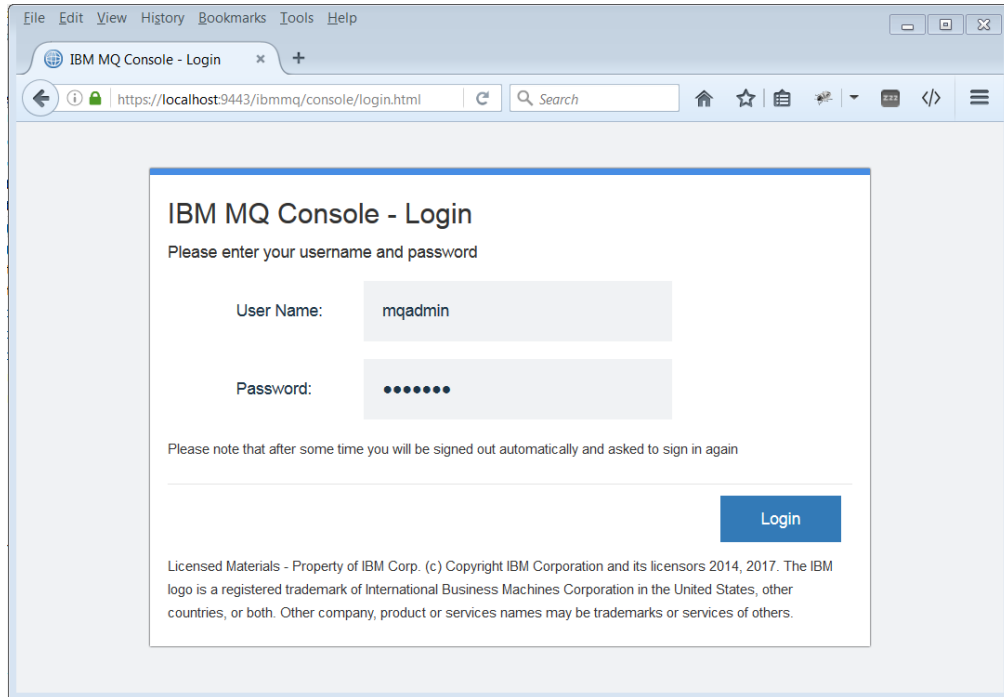
# MQ Console

---

- 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 a different implementation than for the appliance at version 8.0.0.x
  - Functional parity with MQ Console on the appliance at version 8.0.0.x
- 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

# MQ Console – log in

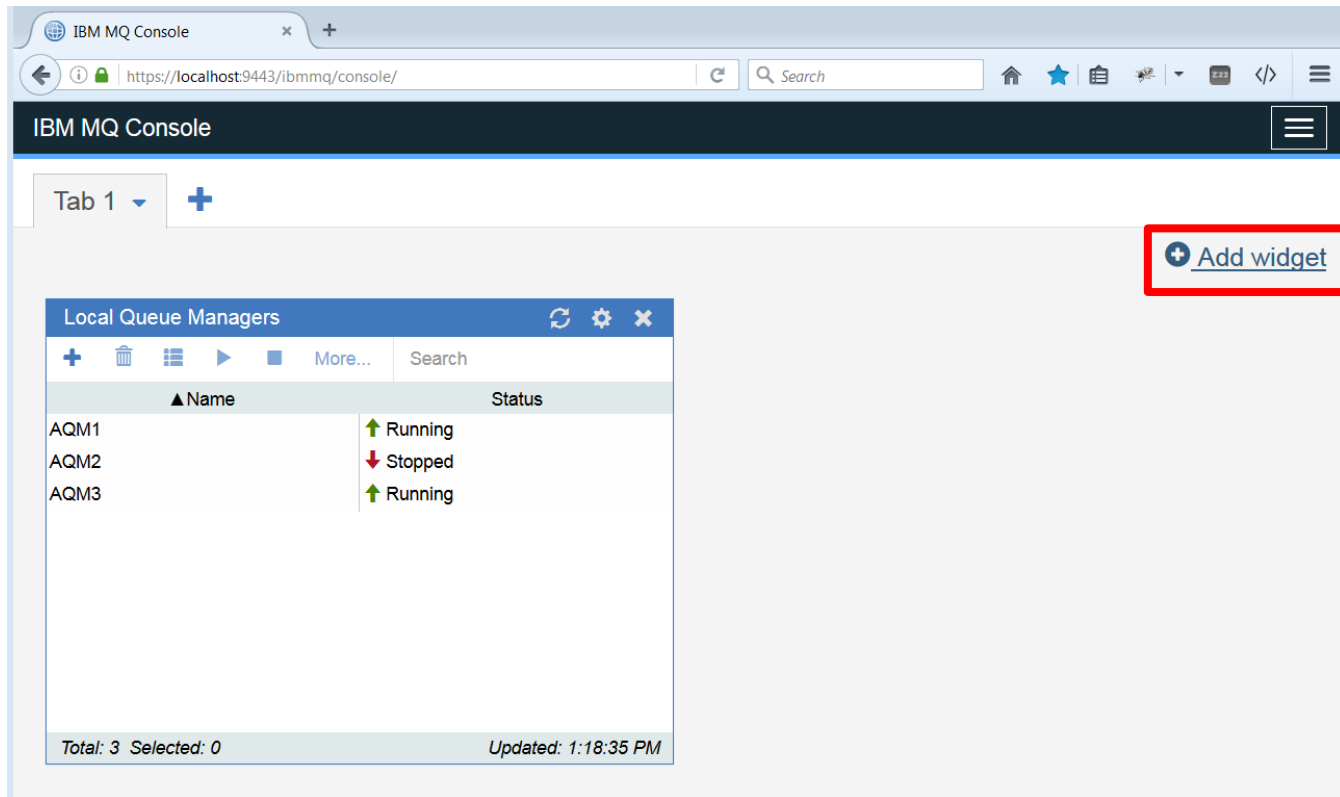
- Point your web-browser at the MQ Console and log in
  - With a user id and password
  - With a client certificate
- Log in credentials validated using user registry configured in the mqweb server
  - Like the REST API
- Access determined by role
  - Same role names as REST API
  - But in a different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



# 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.



The screenshot shows the IBM MQ Console interface in a web browser. The browser address bar displays `https://localhost:9443/ibmmq/console/`. The page title is "IBM MQ Console". Below the title bar, there is a tab labeled "Tab 1" with a plus sign to add more tabs. In the top right corner, there is a button labeled "Add widget" with a plus icon, which is highlighted with a red rectangle. The main content area features a widget titled "Local Queue Managers" with a refresh, settings, and close icon. The widget contains a table with the following data:

Name	Status
AQM1	Running
AQM2	Stopped
AQM3	Running

At the bottom of the widget, it shows "Total: 3 Selected: 0" and "Updated: 1:18:35 PM".



# 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.

The screenshot shows the IBM MQ Console interface with a modal dialog titled "Add a new widget". The dialog is open over a dashboard showing a list of queue managers (AQM1, AQM2, AQM3) under the heading "Local Queue Managers". The dialog contains the following content:

- 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 (dropdown menu)
- Queues** Configure destinations for messages (highlighted with a red box)
- 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
- Close** (button)

The background dashboard shows a "Tab 1" and a "Total: 3 Selected: 0" status at the bottom.

# 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.

The screenshot shows the IBM MQ Console interface in a web browser. The browser address bar displays `https://localhost:9443/ibmmq/console/`. The page title is "IBM MQ Console". Below the title bar, there is a tab labeled "Tab 1" and a "+ Add widget" button.

Two widgets are displayed:

- Local Queue Managers:** A table with columns "Name" and "Status". It lists three queue managers: AQM1 (Running), AQM2 (Stopped), and AQM3 (Running). The status is indicated by green up arrows for "Running" and a red down arrow for "Stopped".
- Queues on AQM1:** A table with columns "Name", "Queue type", and "Queue depth". It lists nine queues (Q1-Q9) with their respective types and depths. The depth for all queues is 0. This widget is highlighted with a red border.

At the bottom of each widget, there is a summary bar: "Total: 3 Selected: 0 Updated: 1:22:07 PM" for the first widget, and "Total: 9 Selected: 0 Updated: 1:22:03 PM" for the second widget.

Name	Status
AQM1	Running
AQM2	Stopped
AQM3	Running

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	

# 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

The screenshot displays the IBM MQ Console interface. At the top, there is a dark header bar with the text "IBM MQ Console" on the left and a hamburger menu icon on the right. Below the header, a navigation bar contains several tabs: "Over view", "AQM1 details", "AQM2 details", and "AQM3 details", followed by a plus sign. The "AQM1 details" tab is highlighted with a red box. Below the navigation bar, the main content area shows a "Local Queue Managers" window. This window has a blue header with the title "Local Queue Managers" and icons for refresh, settings, and close. Below the header is a toolbar with icons for add, delete, list, play, and a "More..." button, along with a search field. The main area of the window is a table with two columns: "Name" and "Status".

Name	Status
AQM1	Running
AQM2	Stopped
AQM3	Running

On the right side of the interface, a sidebar menu is visible, containing the following items: "Help", "Keyboard shortcuts", "Import dashboard", "Export dashboard", "Reset dashboard", "Settings", "About", and "Logout 'mqadmin'". The "Import dashboard" and "Export dashboard" items are highlighted with a red box.

# MQ Console – manage

- Monitor your MQ queue managers using charts generated from statistics information published to system topics
- Display and alter objects using the properties editor
- Browse and send messages
- Provides a sub-set of MQ Explorer function

Properties for 'AQM3'

General

Extended

Cluster

Repository

Communication

Events

SSL

Statistics

Online monitoring

Statistics monitoring

Default transmission queue:

Channel auto definition: Disabled

Channel auto definition exit:

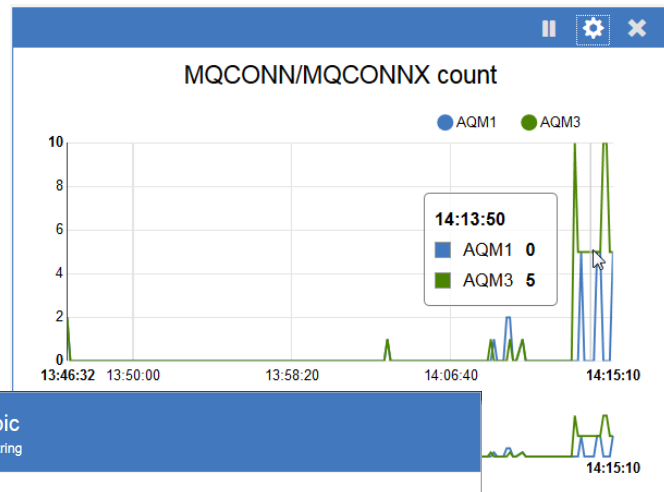
IP address version: IPv4

Activity recording: Message

Trace-route recording: Message

CHLAUTH records: Disabled

REVDNS lookup: Enabled



Subscribe to Topic

Subscribe to a given topic string

Topic string:

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

IBM MQ Lab Tour – Paris – 26 September 2017



**IBM MQ**

Questions?