Integration Technical Conference 2019

M12: Making it easy to develop and support applications with MQ

Matthew Whitehead
mwhitehead@uk.ibm.com

IBM **Cloud**

IBM

Agenda

Why focus on developers?

Learn-MQ

The MQ Developer Badge

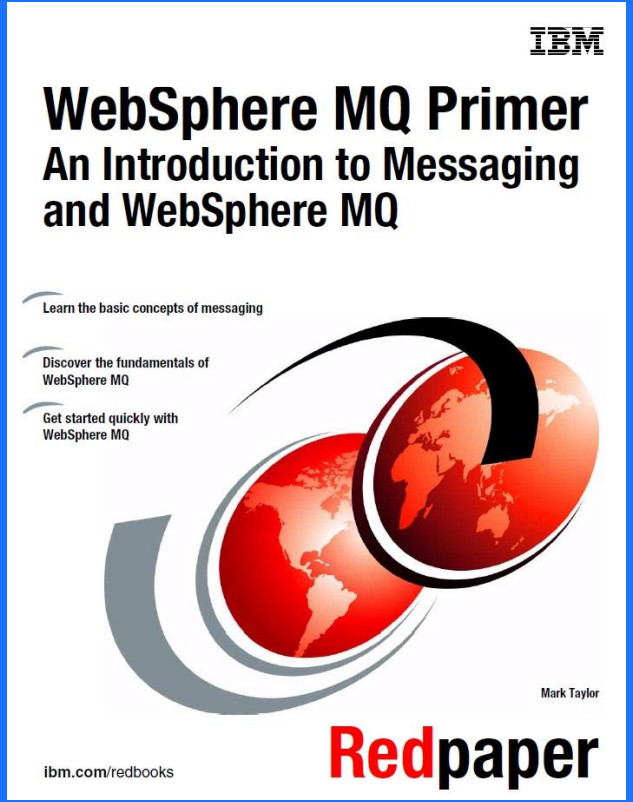
Overview of specific things we built/shipped/created

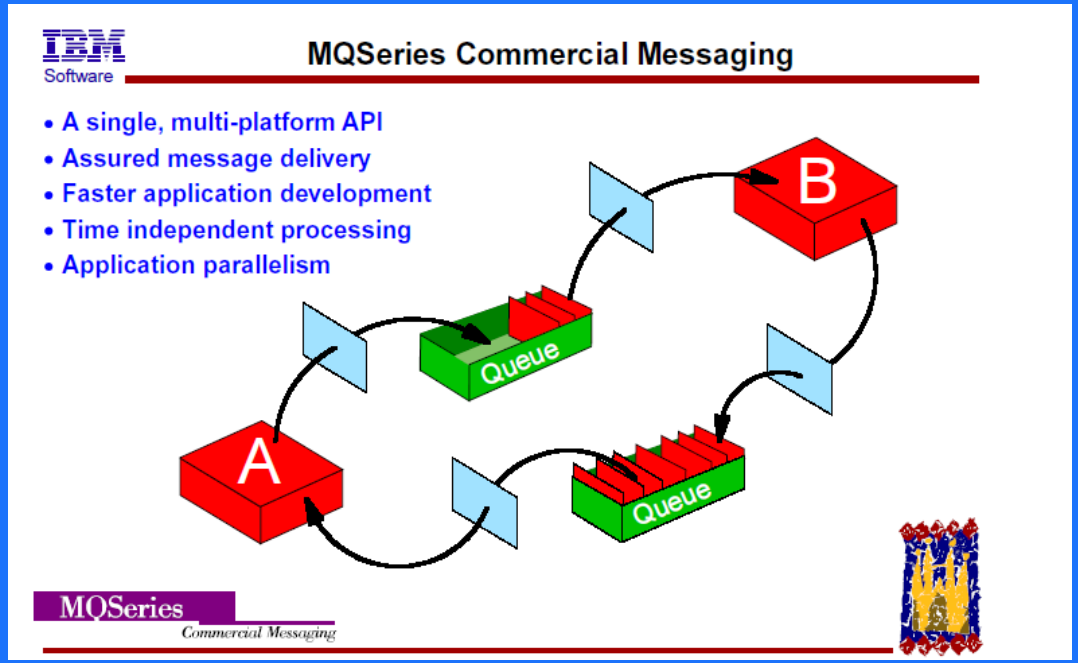
Developer engagement

- MQ can be hard to get going with for new developers
- How do we make it easier for new application teams to pick up MQ and integrate with their development practices?
- Not always been a product focus – spent more time on administration enablement

Developer Engagement

This is what developer engagement used to look like




IBM
Software
MQSeries Commercial Messaging
• A single, multi-platform API
• Assured message delivery
• Faster application development
• Time independent processing
• Application parallelism
A
B
Queue
Queue
MQSeries
Commercial Messaging

What’s it like now?

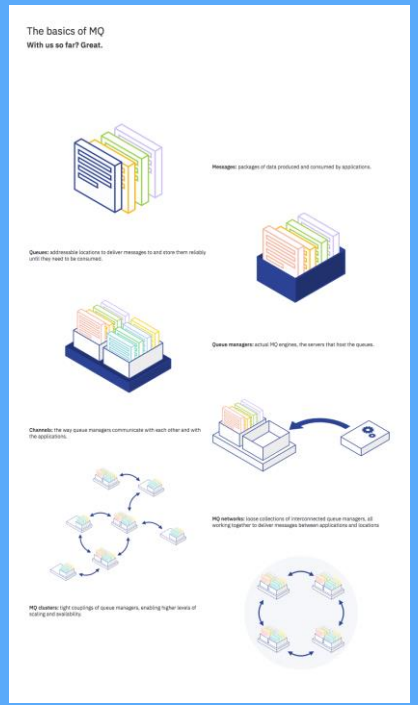
LearnMQ

Finding it hard to get developers started with MQ?

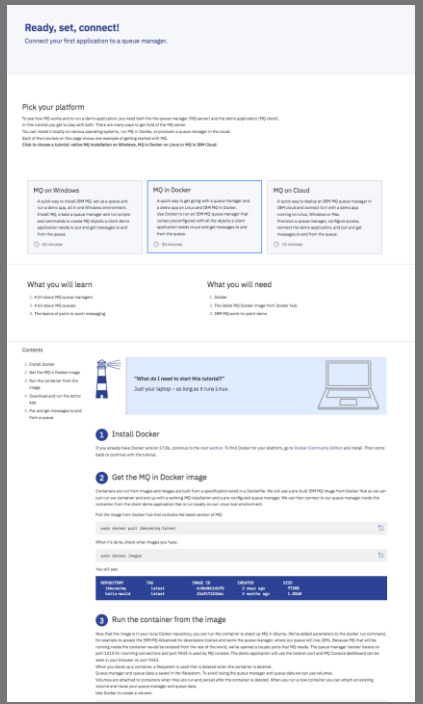
Point them to:

developer.ibm.com/messaging/learn-mq

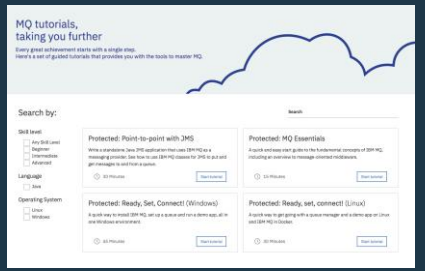
Totally new to MQ? Learn the basics

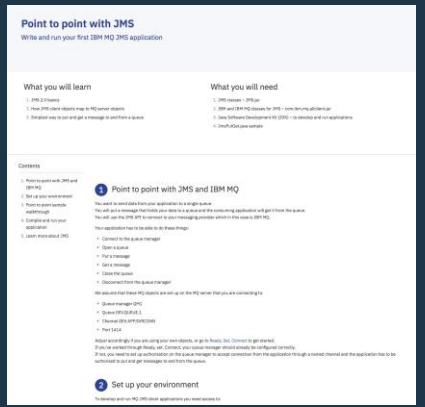


Step-by-step guides to getting up and running with MQ



Tutorials on building your applications and avoiding bad practices





Developer Engagement: Mission Statement

To enable a user instructed to use MQ for the first time, to go from zero understanding to ***running*** a sample application in a sandbox environment with a fundamental understanding of MQ concepts in 2 hours

To enable an application developer, instructed to use MQ for the first time, to go from zero understanding to ***writing*** their first MQ application in the language and environment of their choice within an afternoon

Welcome to messaging, made easy.

IBM MQ is the world's most powerful messaging solution.
LearnMQ will teach you how to master it.

ibm.biz/learn-mq

MQ tutorials, taking you further

Every great achievement starts with a single step.
Here's a set of guided tutorials that provides you with the tools to master MQ.

MQ Essentials (5-10 mins)

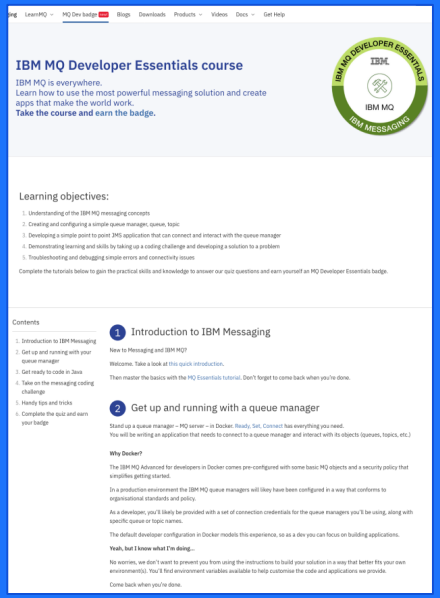
Get a queue manager (10-15 mins)

Hello world (10-15 mins)

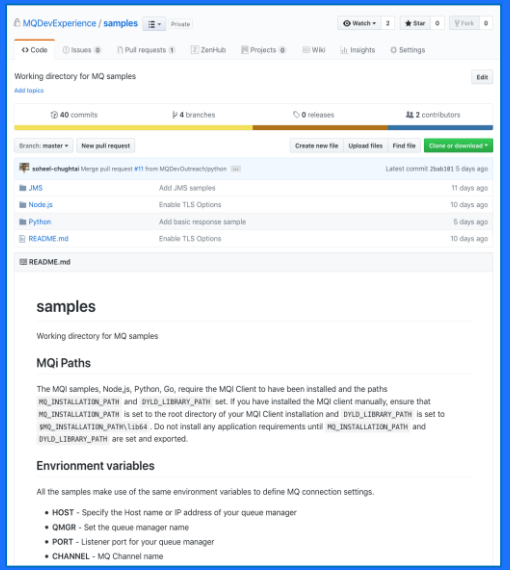
25 years in 25 minutes!

LearnMQ

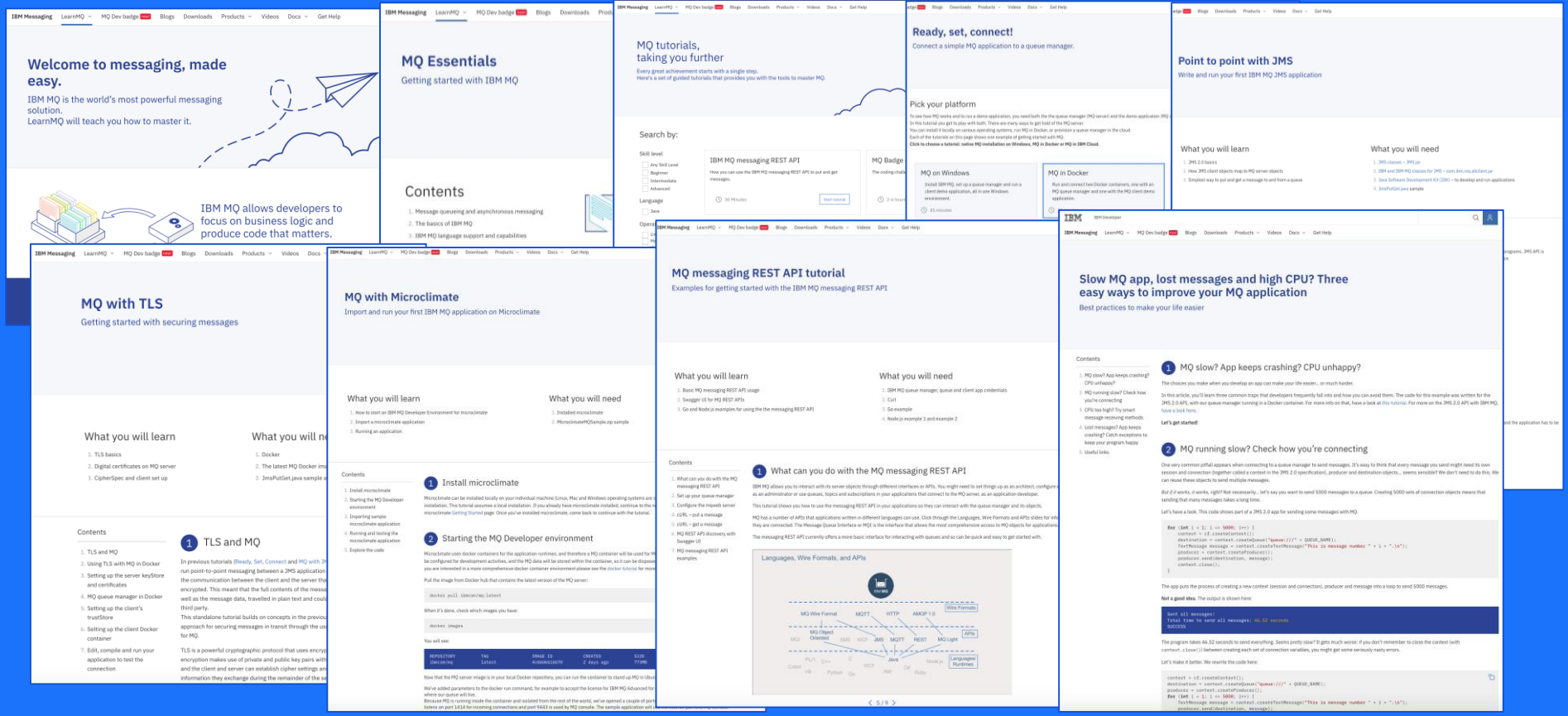
Developer Essentials Badge and challenge



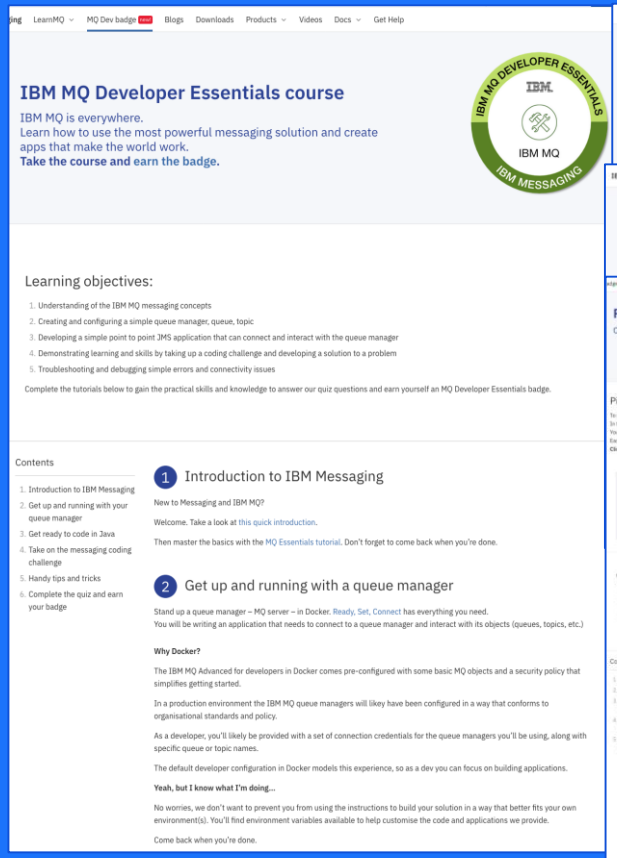
Samples and patterns

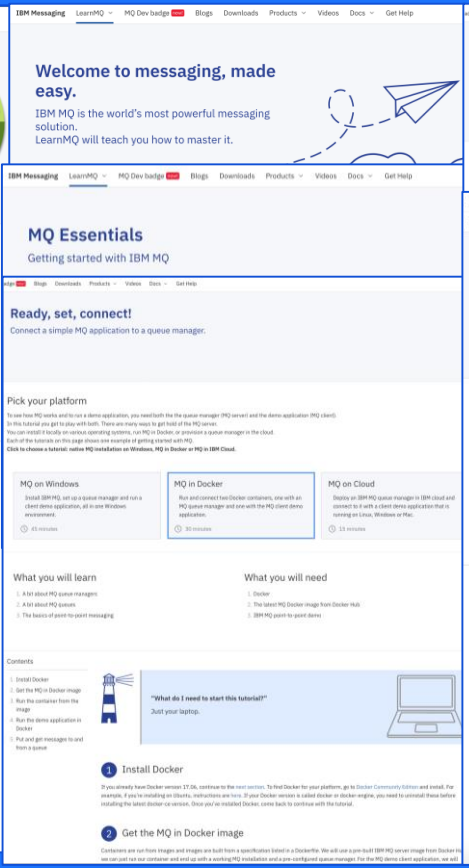


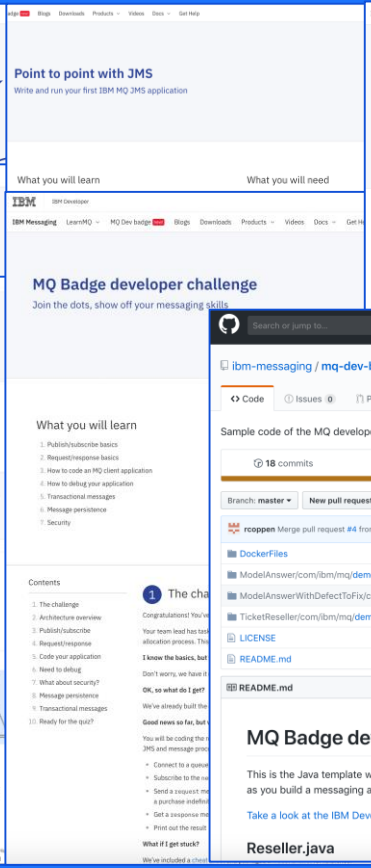
LearnMQ tutorials

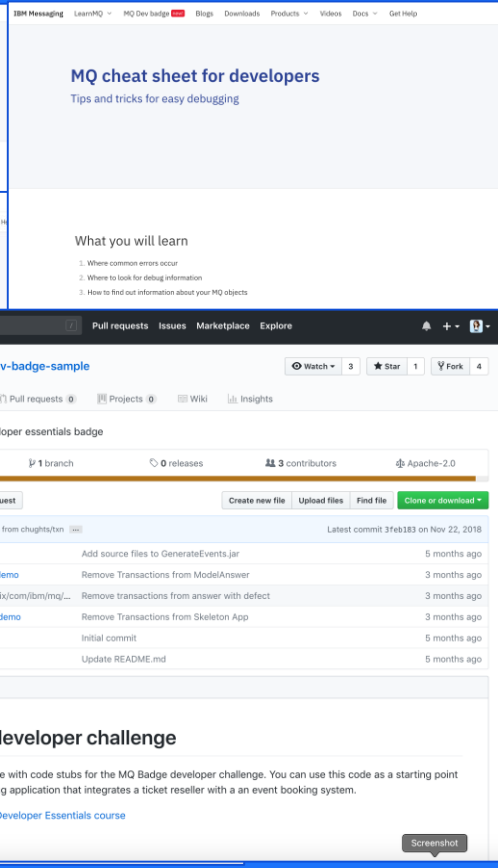


MQ Developer Essentials Badge



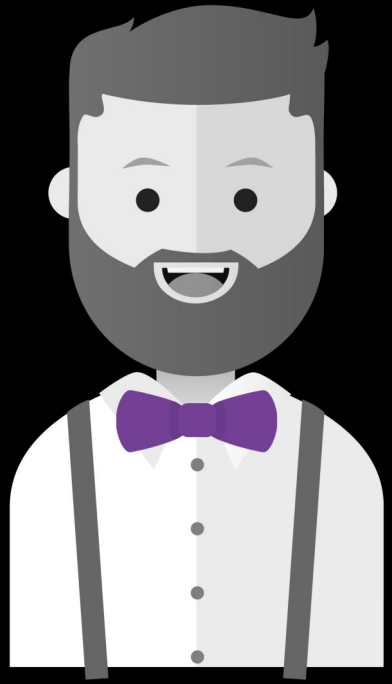






This year, MQ is celebrating its 25th birthday and so is Andre, a new application developer.

He needs to deliver a reliable MQ application for his business.

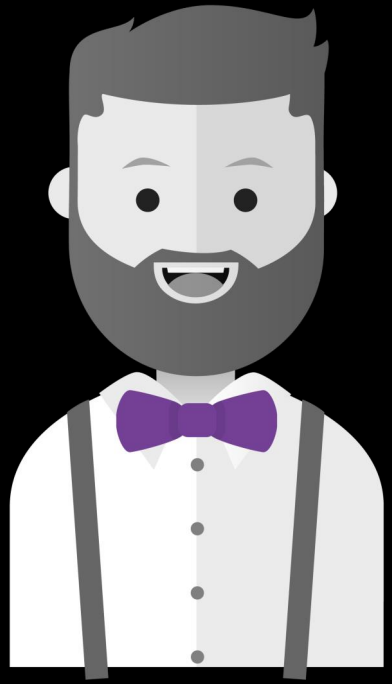


developer: *so what ?*

Just show me an API and I'll code

Working code speaks for itself.

Ready to learn and excited to grow marketable skills.



Andre
Application Developer

I need to deliver a new application

My job is to get code working

Skills

Java | C | Ruby | REST | Linux

developer: insight

responsibilities

- app design & logic
- getting new app connected to the MQ Network
- DevOps
- incident resolution
- getting to done!

needs

- find information on MQ
- core concepts of MQ
- build applications with MQ
- some community support
- marketable skills

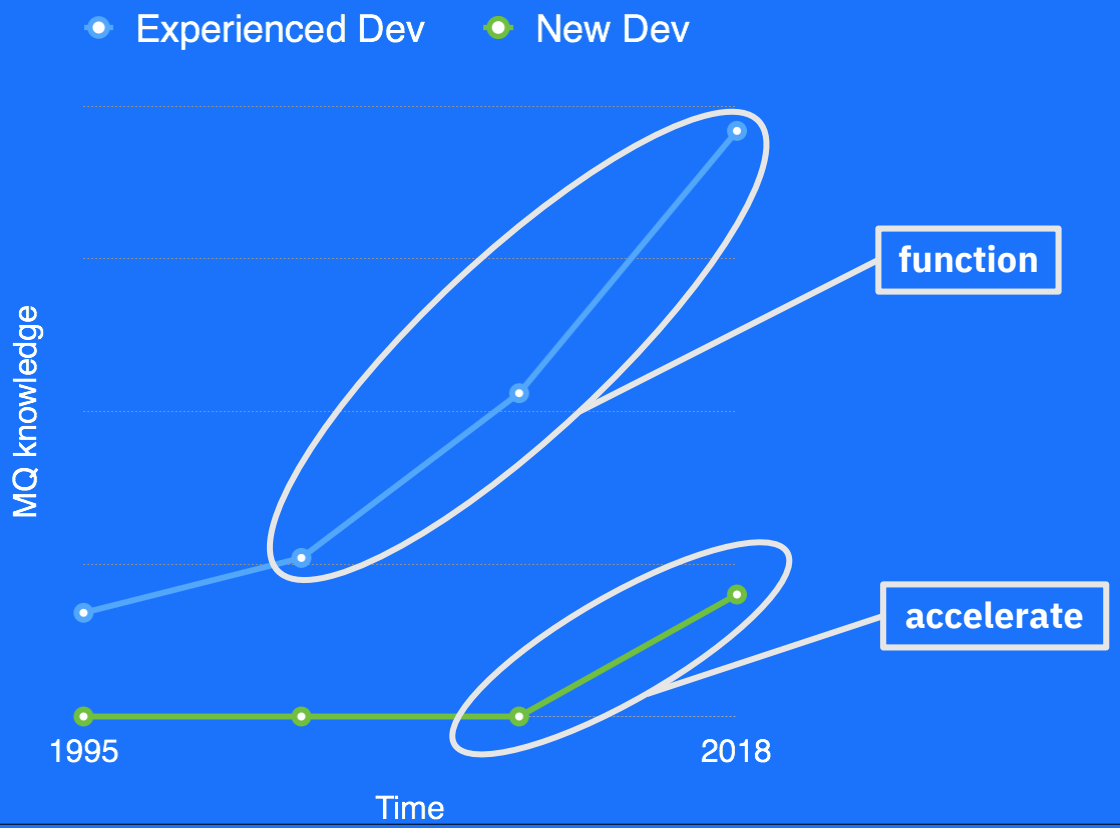
likes

- samples, assets and code to get started (GitHub)
- working in his preferred dev environment
- doing stuff the *right way*
- minimal setup
- enjoying work
- Help when he needs it

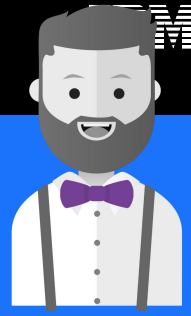
Just one thing,

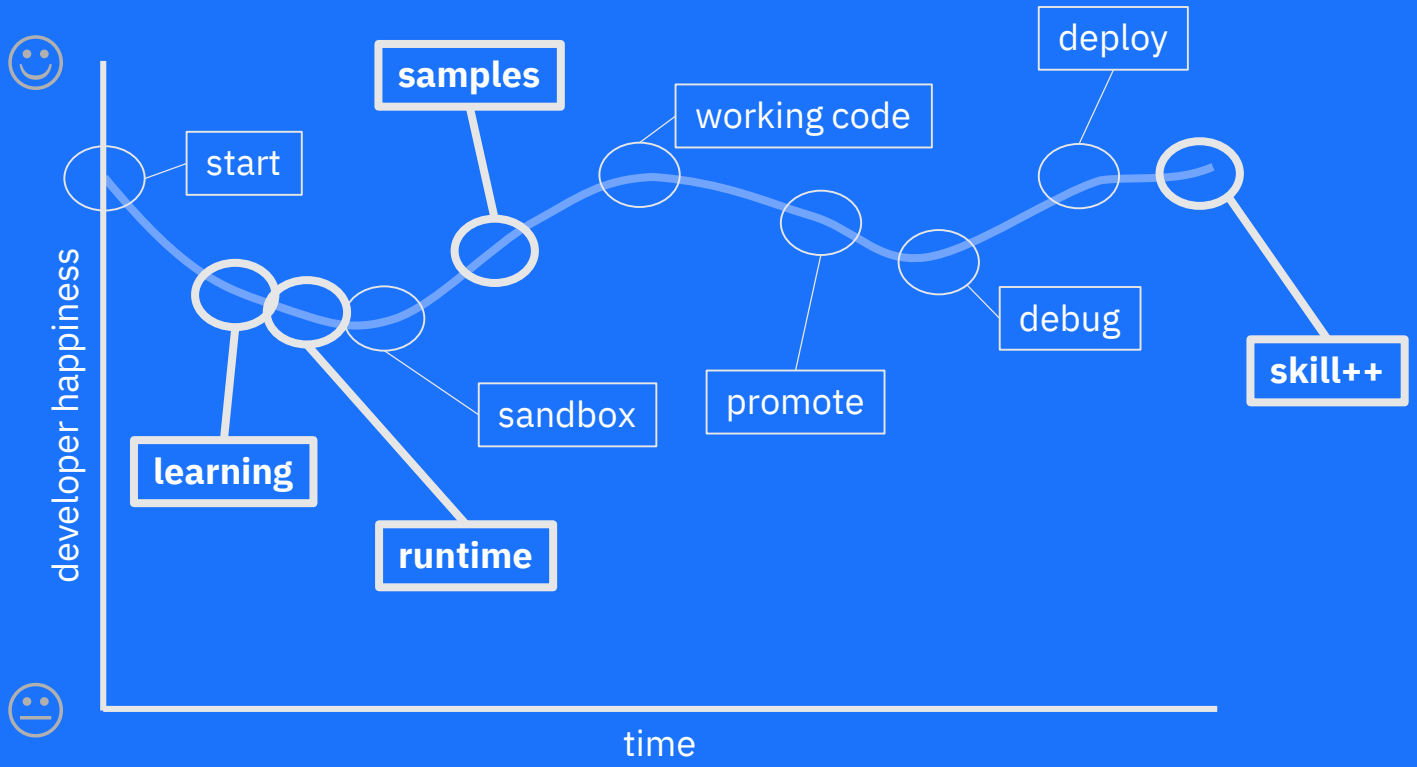
He’s not used IBM MQ before

developer: knowledge cycle

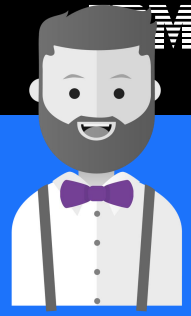


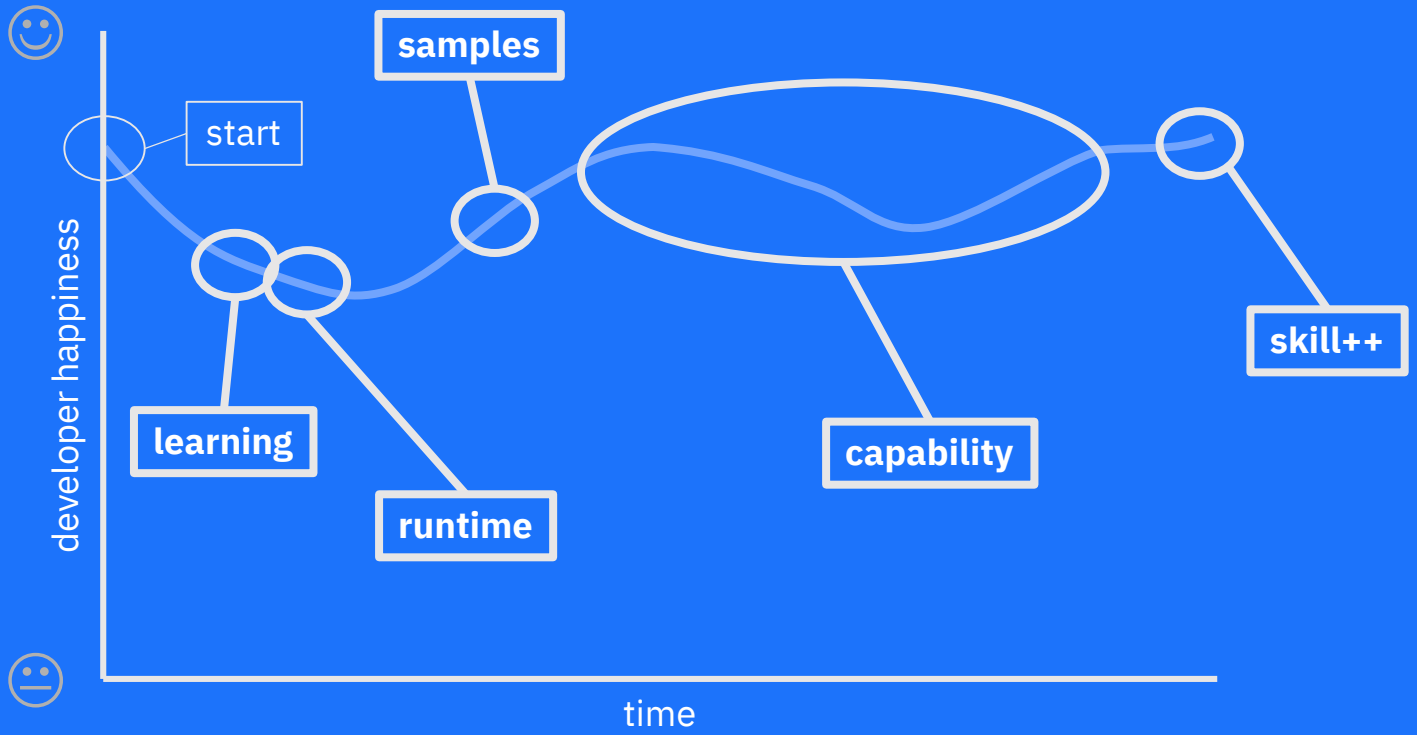
dev experience: positive direction





dev experience: positive direction





developer: challenge

Time to working code

Time to first Hello World

developer: deployment landscape

More than just APIs: scale, reliability, security and extensibility capabilities are critical.

Need working production ready code to succeed.

IBM MQ provides enterprise grade messaging and is a very marketable skill.

developer: resources

Knowledge Centre

Google

MQSeries.net

Internal Doc

stackoverflow

*>50% coders have fewer than 5 years' professional experience**

*https://insights.stackoverflow.com/survey/2018

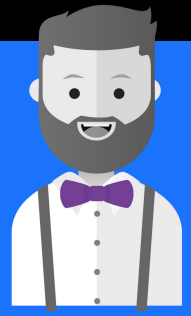
developerWorks

Demo

Recent activity...

(what have we actually been doing)

Supporting developers: recent activity



APIs & Protocols

REST

Go

Node.js

Maven

MQLight

Spring

Docker

LTS & CD

Easier ways to get started – Build, IDEs, and tools

- Developers also need easy access to interfaces – no matter their experience
- Do not want to have to install full products

- Many IDEs and build tools integrate with public repositories
- MQ Java interfaces now available from Central Repository (Maven)
  - No need to explicitly install or download
  - Just reference MQ jars in application configuration

Gradle: build.gradle

```
dependencies {
  compile("com.ibm.mq:com.ibm.mq.allclient:9.1.0.0")
}
```

Maven: pom.xml

```
<dependency>
    <groupId>com.ibm.mq</groupId>
    <artifactId>com.ibm.mq.allclient</artifactId>
    <version>9.1.0.0</version>
</dependency>
```

Easier ways to get started – Java

- Many Java developers use Spring
  - Can reduce amount of code needed
  - MQ JMS used with Spring for many years

build.gradle

```
dependencies {
 compile("com.ibm.mq:mq-jms-spring-boot-starter:+")
}
```

- Spring Boot & Auto-configure give further code reduction
  - You can get a program running quickly
  - With default capabilities

- MQ now has Spring Boot starter
  - Versions for both Boot 1 and Boot 2
  - Source on github; jar on Maven Central

Easier ways to get started – Runtime Availability

- Docker container with full MQ server function
  - Creates sample objects for developers
  - Queues, topics, userids etc
- See github.com/ibm-messaging/container with several build options

- Redistributable Client packages (Linux, Windows) now easier to access
  - To make it easy for developers to package standalone applications
  - Perhaps in a container
  - No login required to download
  - Now includes SDK (header files/copybooks) to enable application development
    - Simplifies creation of a build environment via Docker containers

- See https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/messaging/mqdev/redist/

Easier ways to get started – Mac

- Mac Client for 9.1.1
- Can be found here: ibm.biz/mq-mac-dev

Image courtesy of Blake Patterson:
https://www.flickr.com/photos/blakespot/
License: https://creativecommons.org/licenses/by/2.0/

- Allow Mac users to develop MQ applications
  - Native editor
  - Avoid need for VM just for MQ
- Access C MQI backed language bindings
  - Node.js, Golang, python etc.
- Remote runmqsc
- Local dev using docker container for server side testing
- Not supported for production: dev only.

Client libraries

- Redistributable Client packages now easier to access
  - Make it easy for developers to package standalone applications
  - For example, in a container
  - Windows and Linux

- No SDK install needed to access headers to compile MQ C programs
  - and Cobol and C++
  - A build process might be container-based, creation of container can download/unzip latest SDK

https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/messaging/mqdev/redist/

Multiple APIs and Protocols

- IBM MQ supports multiple APIs and multiple client protocols. Both proprietary and open.

  APIs: **MQI, JMS, MQ Light, REST** …

  Protocols: **MQ, AMQP, MQTT, HTTP**

- These support a wide range of application styles, from the simplest of messaging needs through to the richest

- MQ is the transport; messages produced from any API or protocol can be received by any other API or protocol

APIs

MQI
Exposes the full set of MQ capabilities
Uses the MQ protocol

JMS 2.0
Supports the full JMS API for use in many JSE or JEE environments
Uses the MQ protocol

MQ Light
A simple pub/sub messaging API
Uses the AMQP 1.0 protocol

Protocols

MQ
MQ's highly reliable and performant messaging protocol

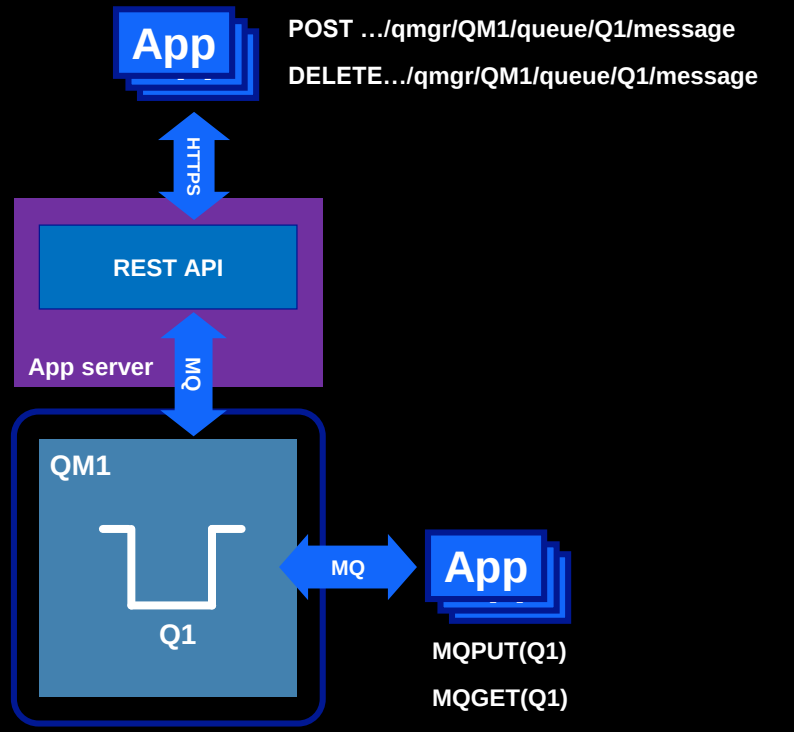
MQTT
Supports the open standard MQTT API and protocol
Open source Eclipse Paho clients

AMQP
Support for AMQP 1.0 enables support for open source clients such as Qpid Proton

REST
A very simple but secure messaging API over REST
…

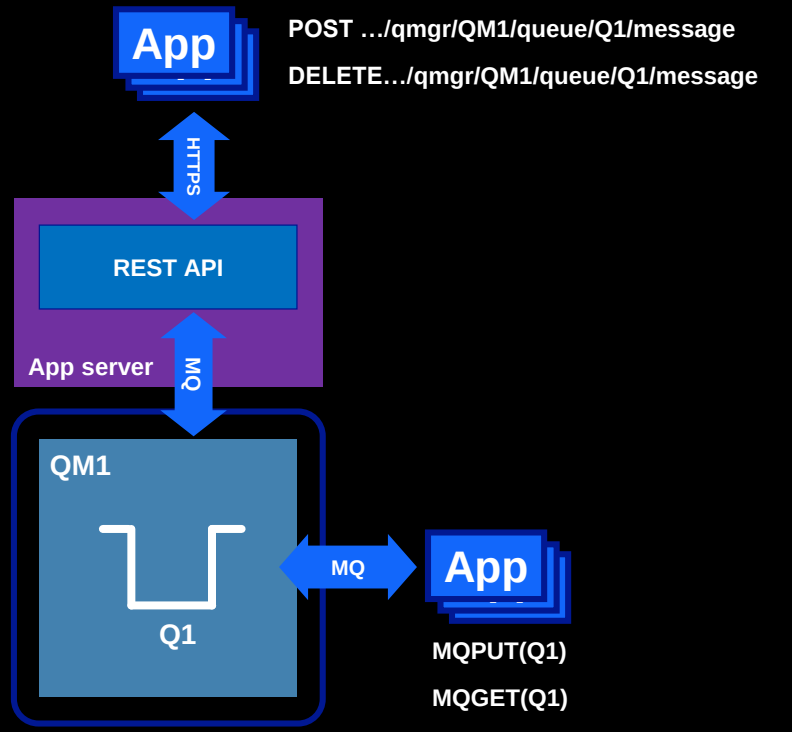
REST Messaging

- Many users just want the ***simplest*** way to get messages in and out of an MQ system
- A RESTful API gives you just that. Easily enabling messaging from just about ***any environment*** with no need for an MQ client or platform/language limitations
- MQ support for z/OS Connect now provides a similar mechanism on z/OS.



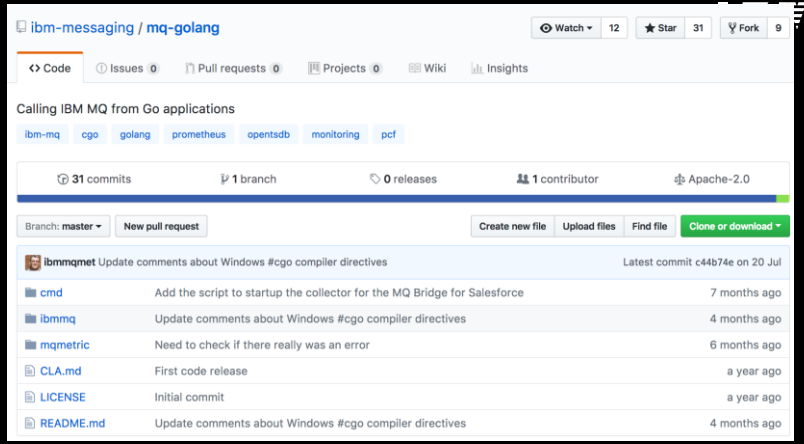
REST Messaging

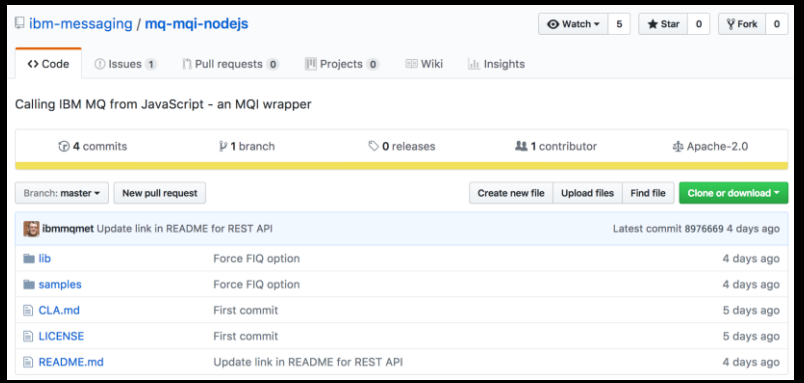
- The new HTTP server support in MQ 9.0.x provides the platform for a properly integrated REST API solution
- CD releases start to see a messaging REST API evolve, first with simple synchronous point-to-point support.



New MQI languages

- The MQI API exposes the richest MQ messaging capabilities
  - Traditionally mostly used by C and COBOL
  - Other bindings have been available
- New interfaces available for **GO** and **Node.js** designed around full MQI function
- Provided as as-is open source, let us know if there's a need to take these further
- Built on the C MQI library





New MQI languages - examples

Intent to make it easy to write for MQ from newer languages and environments

In a natural style for the language

eg Strings, not padded-fixed-lengths

Helps you to develop apps wherever you need

For **Node.js JavaScript** see **github.com/ibm-messaging/mq-mqi-nodejs**

Or **require('ibmmq')** from NPM

For **Go** bindings see **github.com/ibm-messaging/mq-golang**

Both include sample programs to help get started

```
mqmd = new mq.MQMD();
pmo = new mq.MQPMO();
msg = "Hello from Node";

mq.Put(hObj,mqmd,pmo,msg,function(err) {
  if (err) {
    console.log(err);
  }
});
```

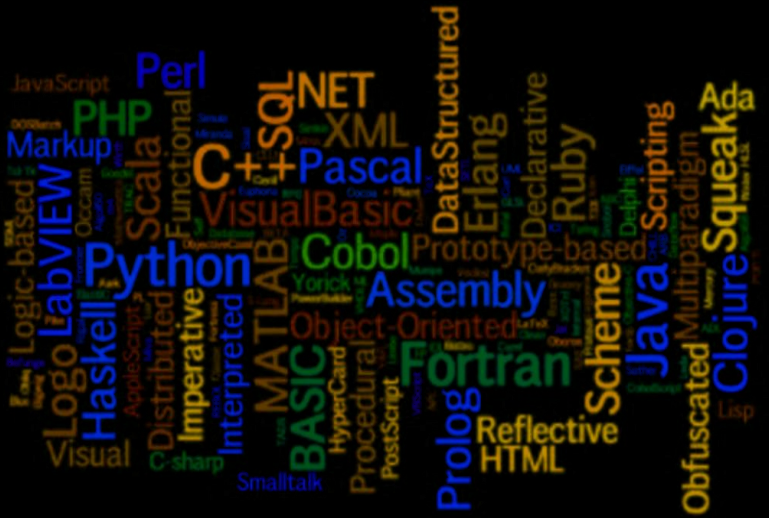
```
mqmd := ibmmq.NewMQMD()
pmo  := ibmmq.NewMQPMO()
mqmd.Format = "MQSTR"
msg  := "Hello from Go"

buffer := []byte(msg)
err = qObject.Put(mqmd, pmo, buffer)
if err != nil {
  fmt.Println(err)
}
```

3rd party contributions

Varying maintenance, capabilities, license

Python: pythonhosted.org/pymqi/
Perl: search.cpan.org/dist/MQSeries/
PHP: pecl.php.net/package/mqseries
Ruby: github.com/reidmorrison/rubywmq

Built on top of C MQI library



MQ Light APIs

MQ Light interfaces in a variety of languages

Client source in GitHub

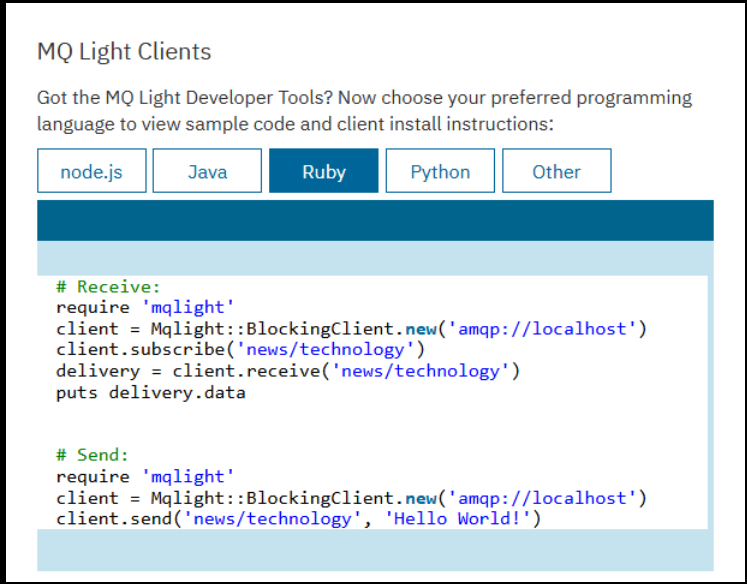
Integrated with natural public repository

- JavaScript: NPM
- Java: Maven
- Ruby: gem install mqlight
- Python: pip install mqlight

.Net blog series at

developer.ibm.com/messaging/2017/11/13/mq-light-messaging-microsoft-net-part-1/

developer.ibm.com/messaging/mq-light



Availability of runtimes

Docker container with full MQ server function

- Creates sample objects for developers
- Queues, topics, userids etc

See https://hub.docker.com/r/ibmcom/mq/

Redistributable Client packages now easier to access

- To make it easy for developers to package standalone applications
- Perhaps in a container

public.dhe.ibm.com/ibmdl/export/pub/software/websphere/messaging/mqdev/redist

Thank you

Trademark Statement

- IBM and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions. Other marks may be trademarks or registered trademarks of their respective owners.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company, product and service names may be trademarks, registered marks or service marks of their respective owners.
- References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

Important Disclaimer

- THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
- WHILST EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
- IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.
- IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.
- NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:
  - CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
  - ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.