

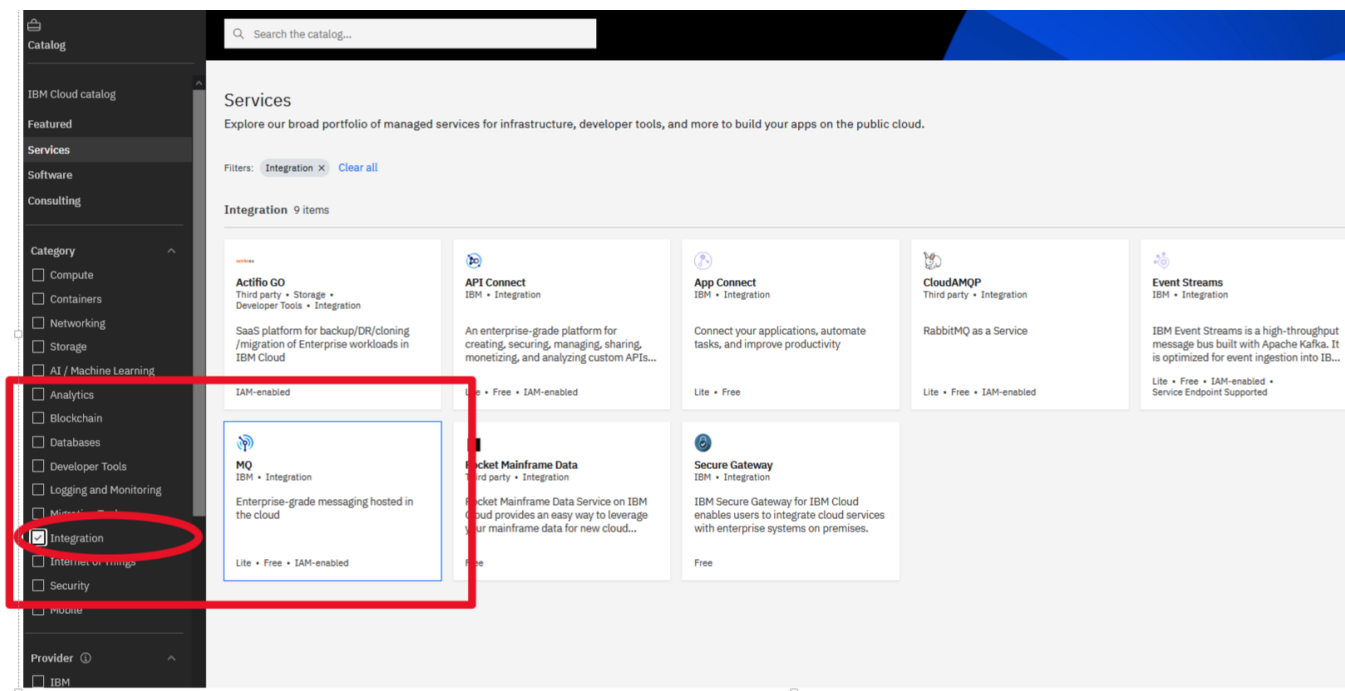
lab-07 Работа Node.js с очередями IBM MQ

Создание в облаке IBM MQ

В облачном каталоге, MQ можно найти по прямому линку: <https://cloud.ibm.com/catalog/services/mq>

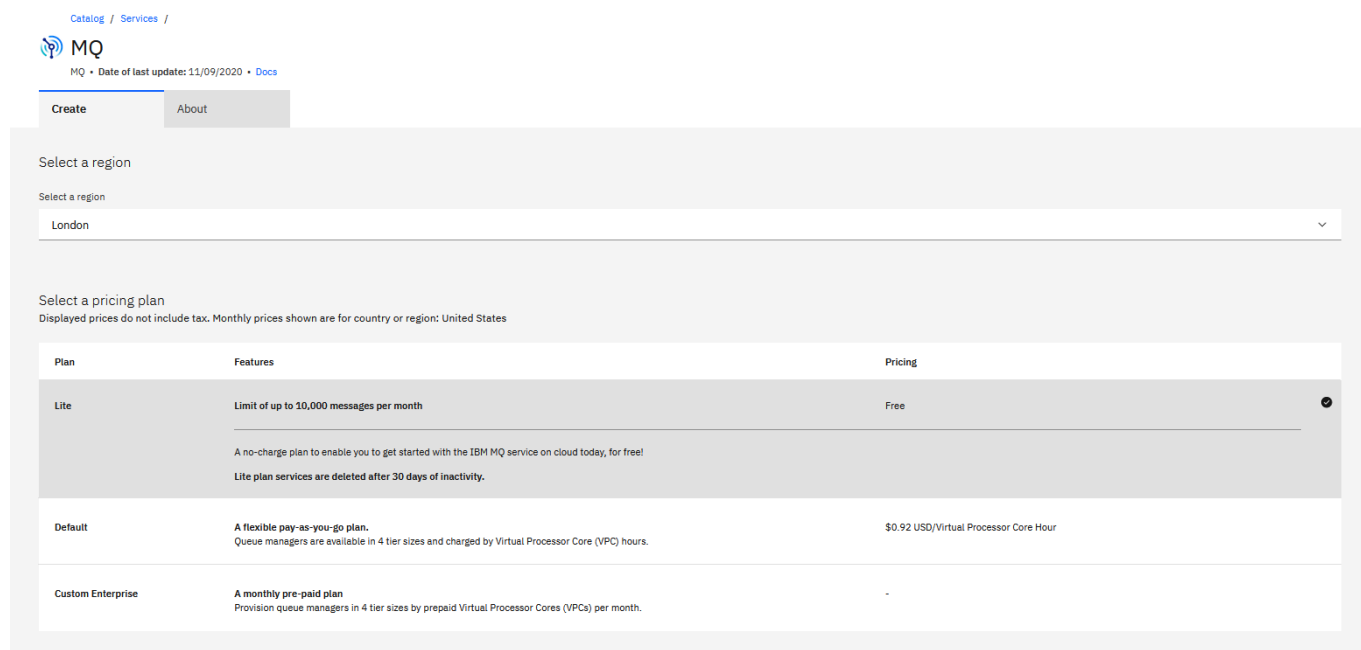
[Линк на документацию IBM MQ](#)

- найти в каталоге



pic-01

- запустить создание, выбрав light plan



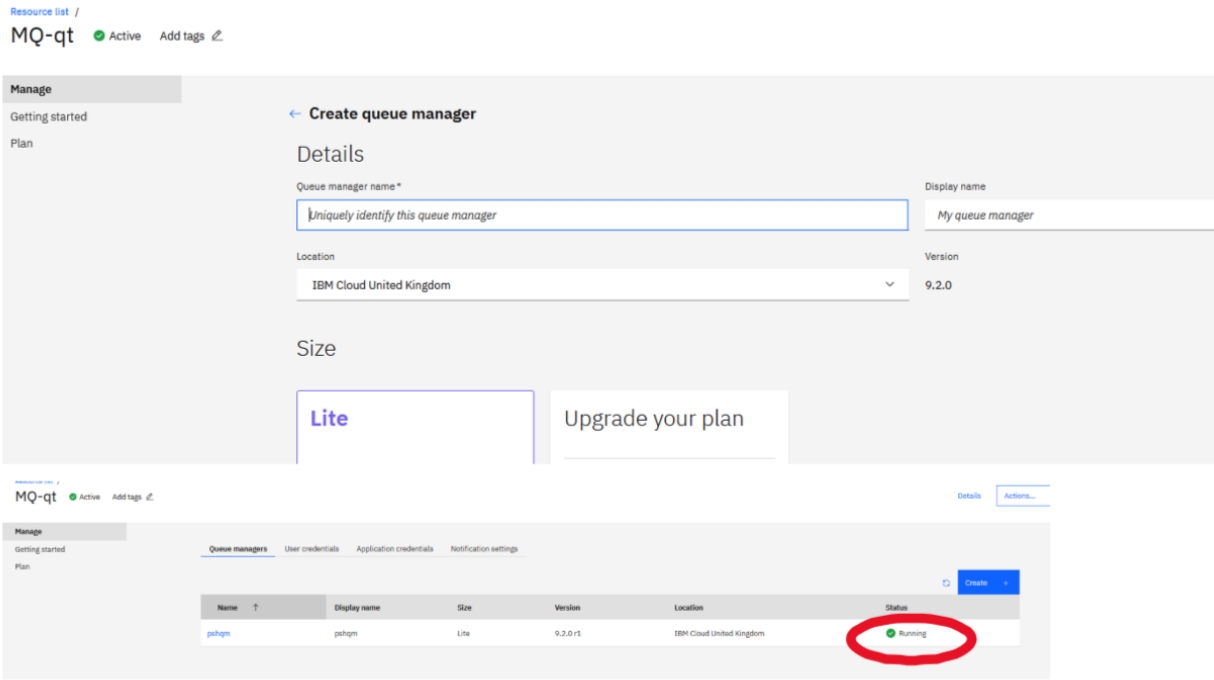
pic-02

- Дождаться пока задеплоится сервис и перейдет в статус "Active"

CONTINUOUS DELIVERY	VERSION	LOCATION	CONTINUOUS DELIVERY	STATUS	
Db2-bv	default	London	Db2	Active	—
IBM Log Analysis with LogDNA-u2	default	London	IBM Log Analysis with LogDNA	Active	—
MQ-qt	default	London	MQ	Active	—
Watson Assistant-zy	default	London	Watson Assistant	Active	—

pic-03

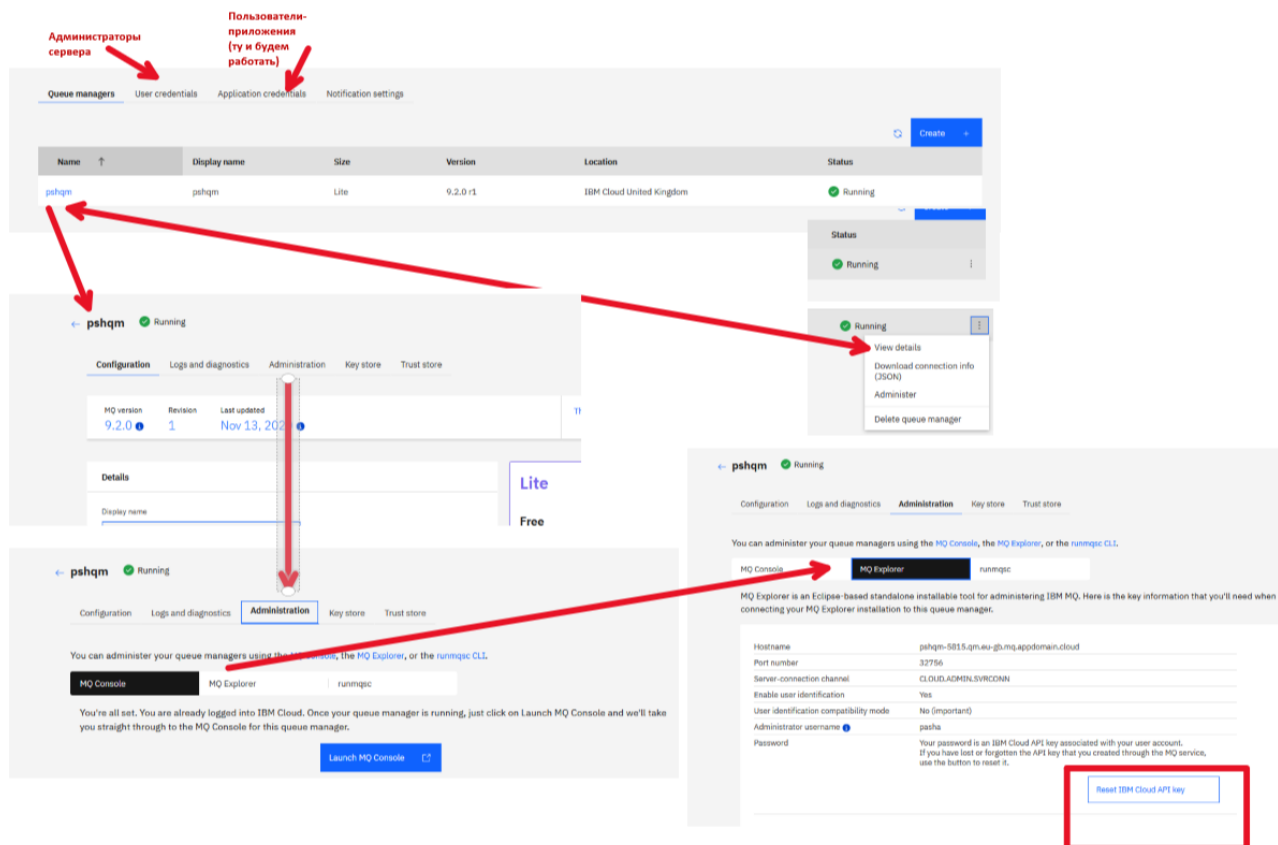
- Создать QM-manager и дождаться, когда он стартует



pic-04

Все, сервис создан!

- Сгенерировать административный API KEY с скачать параметры подключения



pic-05

Скачать Connection-info

```
{
  "platform": "ibmmq-on-cloud",
  "queueManagerName": "****",
  "hostname": "*****.qm.eu-gb.mq.appdomain.cloud",
  "listenerPort": 32***,
  "logPort": 30***,
  "applicationChannelName": "CLOUD.APP.SVRCONN",
  "adminChannelName": "CLOUD.ADMIN.SVRCONN",
  "deploymentLocation": "bmx-eu-gb",
  "webConsoleURL": "https://web-*****.qm.eu-gb.mq.appdomain.cloud/ibmmq/console",
  "messagingREST": "https://web*****.qm.eu-gb.mq.appdomain.cloud/ibmmq/rest/v1/messaging/qmgr/****/queue/(queue_name)/message",
  "adminREST": "https://web-****.qm.eu-gb.mq.appdomain.cloud/ibmmq/rest/v1/admin/qmgr/*****"
}
```

А, сгенерированный административный API KEY выглядит так

```
{
  "name": "MQ user key",
  "mqUsername": "*****",
  "description": "Automatically created api key for MQ usage",
  "createdAt": "*****",
  "apiKey": "*****"
}
```

Это используется для доступа к администрированию программными способами, на пример через REST API

- Перейти в раздел Application credentials и создать для своего приложения реквизиты подключения. Их будем использовать при подключении приложения к серверу MQ

The screenshot shows the 'Application credentials' section of the IBM Cloud console. At the top, there are tabs for 'Queue managers', 'User credentials', 'Application credentials', and 'Notification settings'. Below the tabs is a table with columns 'Name', 'Application username', and 'Description'. The table contains two entries: 'node2' and 'nodeapp'. A red box highlights the 'Add' button in the top right corner. A red arrow points from this button to the 'Add new application credentials' form. The form has fields for 'Display name' (filled with 'Svc1'), 'Description' (filled with 'Svc1 - типа мой сервис по загрузке курсов'), and 'Application username' (filled with 'svc1'). A red box highlights the 'Add and generate API key' button at the bottom right of the form. Below the form, a text label says 'А вот тут и генерируются параметры подключения'. At the bottom, there is another table showing the updated list of credentials, including 'svc1'. A dropdown menu is visible next to the 'svc1' entry, with options: 'Edit MQ username', 'Add new API key', and 'Remove application'.

Name	Application username	Description
node2	node2	node app 2 users
nodeapp	nodeapp	nodeapp

← Add new application credentials

Details

Display name *
Svc1

Description
Svc1 - типа мой сервис по загрузке курсов

Application username *
svc1

Cancel Add and generate API key

А вот тут и генерируются параметры подключения

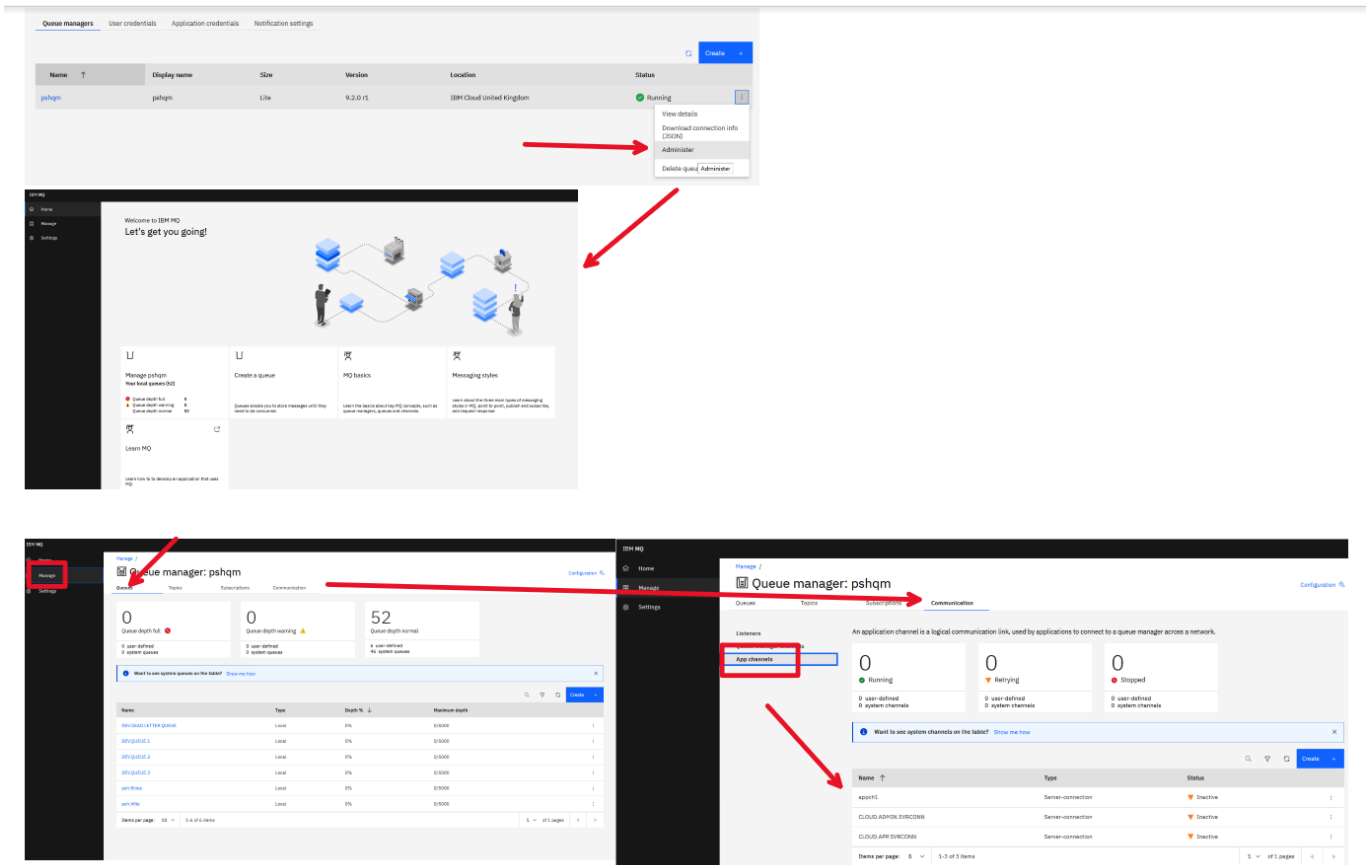
Name	Application username	Description
node2	node2	node app 2 users
nodeapp	nodeapp	nodeapp
svc1	svc1	Svc1 - типа мой сервис по загрузке курсов

Edit MQ username
Add new API key
Remove application

pic-06

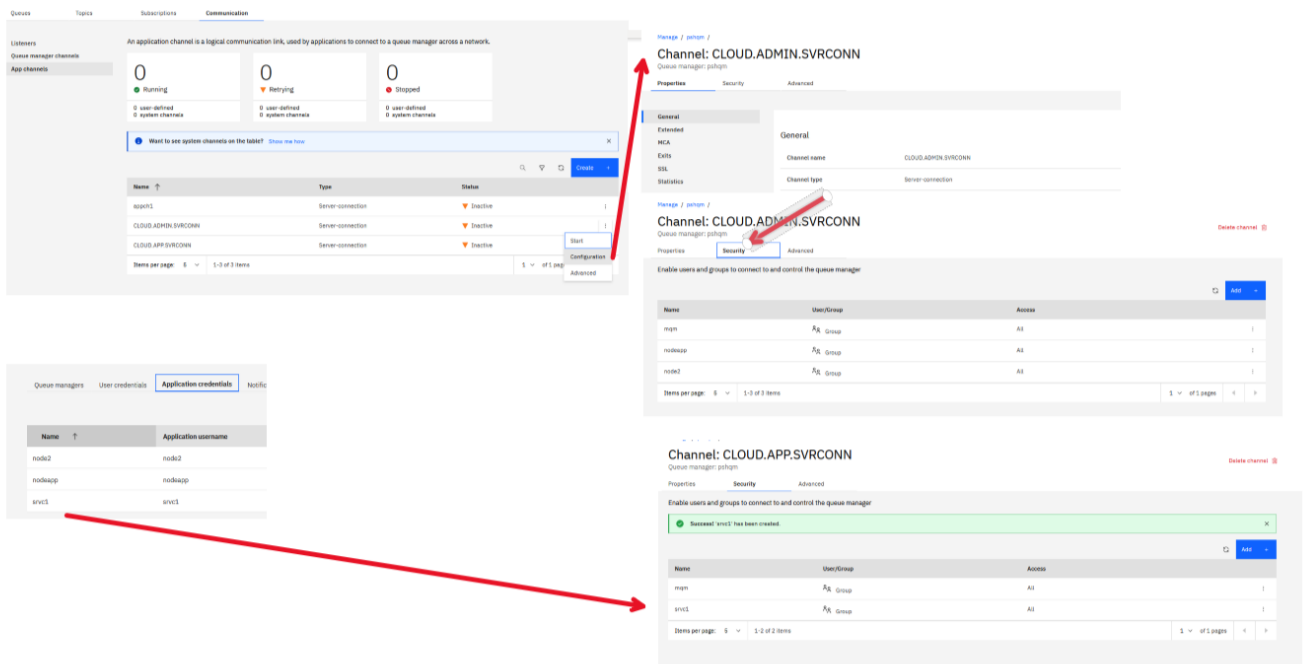
- Самый важный момент! зайти в настройку коммуникационных каналов и доавить параметры вашего приложения в канал

Найти каналы можно на риске pic-7



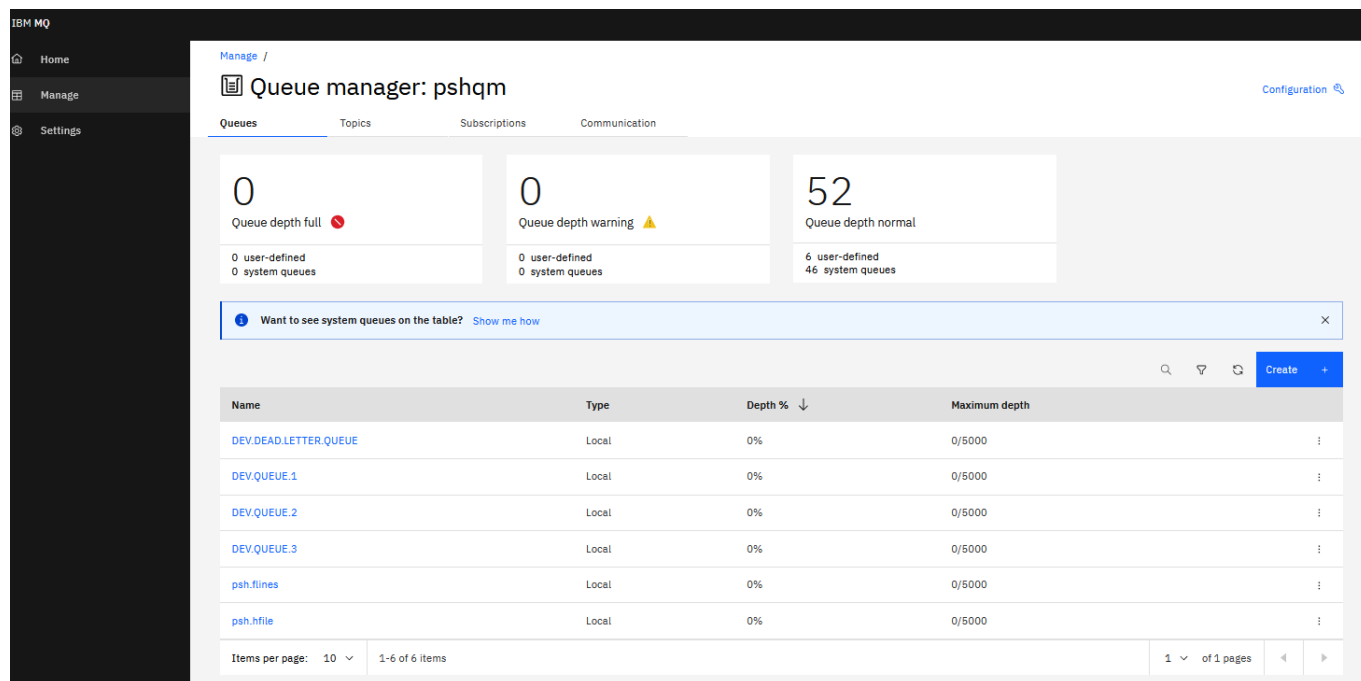
pic-07

- Добавить пользователя вашего приложения, для доступа к каналу



pic-08

- Настройка закончена. Можем создать необходимые очереди и положить несколько тестовых сообщений в очередь



pic-09

Заметки по поводу Node.JS библиотеки для работы с IBM MQ

С IBM MQ on-premise можно подключаться по

- "родному" протоколу
- по MQTT протоколу
- по MQ-Light протоколу
- по Rest API

В облачной версии нашел только 2:

- "родной" протокол
- Rest API

В <https://github.com/ibm-messaging/mq-mqi-nodejs> представлена библиотека. Там много и других библиотек. В npm репозитории нужно искать: [mq-mqi-nodejs](#) Библиотека обновлялась 4 месяца назад под новую версию MQ.

Для MQTT протокола можно использовать библиотеку: [MQTT.js](#) [MQ-Light](#)

Но, эти библиотеки нужно изучать еще

Описание модуля по работе с IBM MQ по "родному" протоколу

Для работы с MQ написан модуль `./services/imq-srv.js`, который имеет 3 класса:

- **MQBase** - базовый класс, который обеспечивает вычитку конфигурации и подключение/отключение к MQ-manager, и несколько сервисных функций.
- **MQSender** - расширение **MQBase** функциями для отправки сообщения в очередь
- **MQReciever** - расширение **MQBase** функциями для синхронного чтения сообщений из очереди.

Для работы класса используются такие конфигурационные параметры **localdev-config.json** :

```
{
  "mq_qm": "Имя MQ manager",
  "mq_host": "Хост для подключения",
  "mq_port": "Порт для подключения",
  "mq_channel": "Канал для подключения",
  "mq_username": "Application username для подключения к каналу",
  "mq_apikey": "apikey для подключения к приложению"
}
```

(./config/mappings.json)

```
{
  "version": 1,
  "mq_qm": {
    "searchPatterns": [
      "env:mq_qm",
      "file:/localdev-config.json:$.mq_qm"
    ]
  },
  "mq_host": {
    "searchPatterns": [
      "env:mq_host",
      "file:/localdev-config.json:$.mq_host"
    ]
  },
  "mq_port": {
    "searchPatterns": [
      "env:mq_port",
      "file:/localdev-config.json:$.mq_port"
    ]
  },
  "mq_channel": {
    "searchPatterns": [
      "env:mq_channel",
      "file:/localdev-config.json:$.mq_channel"
    ]
  },
  "mq_username": {
    "searchPatterns": [
      "env:mq_username",
      "file:/localdev-config.json:$.mq_username"
    ]
  },
  "mq_apikey": {
    "searchPatterns": [
      "env:mq_apikey",
      "file:/localdev-config.json:$.mq_apikey"
    ]
  }
}
```

```
}  
}
```

В каталоге **./test** находятся 2 тестовых файла:

- **test-imq-srvc-send.js** отправка сообщений в очередь
- **test-imq-srvc-recieve.js** синхронное чтение сообщений из очереди.

Последовательность отправки сообщений:

- подключить модуль и создать класс

```
// подключаем модуль  
var xsrv = require('../services/imq-srvc');  
var srv = new xsrv.MQSender();
```

- *mq_connect* подключиться к MQ-менеджеру
- *mq_openq* подключиться к очереди (к объекту MQ-менеджера)
- *mq_putMessage* отправить сообщение в очередь (тут можно в цикле)
- *mq_closeq* отключиться от очереди (от объекта MQ-менеджера)
- *q_disconnect* отключиться от MQ-менеджера

Последовательность при чтении сообщений:

- подключить модуль и создать класс

```
// подключаем модуль  
var xsrv = require('../services/imq-srvc');  
var srv = new xsrv.MQReciever();
```

- *mq_connect* подключиться к MQ-менеджеру
- *mq_openq* подключиться к очереди (к объекту MQ-менеджера)
- *mq_getMessage* Синхронное чтение сообщения и очереди
- *mq_closeq* отключиться от очереди (от объекта MQ-менеджера)
- *q_disconnect* отключиться от MQ-менеджера

Результаты прогона тестовы

Отправка сообщения

9 / 12

Waiting for the debugger to disconnect...

Manage / pshqm /

Local Queue: DEV.QUEUE.1 messages

2 messages (0.04%)

Maximum queue depth: 5000

Timestamp ↓	Application ID	User ID	Application data
2020-11-15 22:48:27	rogram Files\nodejs\node.exe	nodeapp	{{"r030":36,"txt":"Австралийский доллар","rate":20.1826,"cc":"AUD","exchangedate":"01.10.2020"},{"r030":124,"txt":"Канадский доллар","rate":21.1525,"cc":"CAD","exchangedate":"01.10.2020"},{"r030":156,"txt":"Юань Женьминьби","rate":4.1578,"cc":"CNY","exchangedate":"01.10.2020"}, {"r030":191,"txt":"Кюна","rate":4.3883,"cc":"HRK","exchangedate":"01.10.2020"}, {"r030":203,"txt":"Чешская крона","rate":1.2194,"cc":"CZK","exchangedate":"01.10.2020"}, {"r030":208,"txt":"Данская крона","rate":4.4541,"cc":"DKK","exchangedate":"01.10.2020"}, {"r030":344,"txt":"Гонконгский доллар","rate":3.6529,"cc":"HKD","exchangedate":"01.10.2020"}, {"r030":348,"txt":"Фонконтский доллар","rate":0.09078,"cc":"HUF","exchangedate":"01.10.2020"}, {"r030":356,"txt":"Индийская рупия","rate":0.38371,"cc":"INR","exchangedate":"01.10.2020"}, {"r030":360,"txt":"Пыния","rate":0.0019025,"cc":"IDR","exchangedate":"01.10.2020"}, {"r030":376,"txt":"Новый испанский шекель","rate":8.2363,"cc":"ILS","exchangedate":"01.10.2020"}, {"r030":392,"txt":"Ена","rate":0.26786,"c
2020-11-15 22:48:26	rogram Files\nodejs\node.exe	nodeapp	{"sender":"test","reciever":"test","body":"Я пришел к тебе с приветом, рассказать, что солнце встало!"}

Items per page: 10 1-2 of 2 items 1 of 1 pages

pic-10

Чтение из очереди

```

Тестируем модуль imq-srvs
mq_connect=start
mq_connect: Подключаюсь!
mq_connect: OK=Подключились
{"_hConn":2113929221,"_name":"pshqm"}
  ✓ function mq_connect - подключиться к серверу (1495ms)
mq_openq: OK=подключились
i_hObj={"_hObj":101,"_mqQueueManager":
{"_hConn":2113929221,"_name":"pshqm"},"_name":"DEV.QUEUE.1"}
  ✓ Открываю очередь для записи сообщений (143ms)
mq_getMessage: Читаю сообщение из очереди
{"mqmdhdr":
{"msgid":"414d5120707368716d202020202020d8c6ae5f02fccf23","corelid":"414d5120707
368716d202020202020d8c6ae5f03fccf23","format":"MQSTR","putdate":"20201115","enco
ding":1208,"CodedCharSetId":1208},"msg":
{"sender":"test","reciever":"test","body":"Я пришел к тебе с приветом, рассказать,
что солнце встало!"}}
  ✓ Прочитать из очереди сообщения (129ms)
mq_closeq: OK=Соединение закрыто!
i_hObj= Отключились
  ✓ Отключиться от объекта ОЧЕРЕДЬ (213ms)
mq_disconnect: OK=отключились!
  ✓ Отключиться от сервера MQ (99ms)

```

5 passing (2s)

Waiting for the debugger to disconnect.

Manage / pshqm /

Local Queue: DEV.QUEUE.1 messages

Actions

1 messages (0.02%)

Maximum queue depth: 5000

🔍 🗑️ ↺ Create +

Timestamp ↓	Application ID	User ID	Application data
2020-11-15 22:48:27	rogram Files\nodejs \node.exe	nodeapp	[{"r030":36,"txt":"Австралийський долар","rate":20.1826,"cc":"AUD","exchangedate":"01.10.2020"}, {"r030":124,"txt":"Канадський долар","rate":21.1525,"cc":"CAD","exchangedate":"01.10.2020"}, {"r030":156,"txt":"Юань Женьміньбі","rate":4.1578,"cc":"CNY","exchangedate":"01.10.2020"}, {"r030":191,"txt":"Куна","rate":4.3883,"cc":"HRK","exchangedate":"01.10.2020"}, {"r030":203,"txt":"Чеська крона","rate":1.2194,"cc":"CZK","exchangedate":"01.10.2020"}, {"r030":208,"txt":"Данська крона","rate":4.4541,"cc":"DKK","exchangedate":"01.10.2020"}, {"r030":344,"txt":"Гонконгівський долар","rate":3.6529,"cc":"HKD","exchangedate":"01.10.2020"}, {"r030":348,"txt":"Форинт","rate":0.09078,"cc":"HUF","exchangedate":"01.10.2020"}, {"r030":356,"txt":"Індійська рупія","rate":0.38371,"cc":"INR","exchangedate":"01.10.2020"}, {"r030":360,"txt":"Рупія","rate":0.0019025,"cc":"IDR","exchangedate":"01.10.2020"}, {"r030":376,"txt":"Новий ізраїльський шекель","rate":8.2363,"cc":"ILS","exchangedate":"01.10.2020"}, {"r030":392,"txt":"Бена","rate":0.26786,"c

Items per page: 10 1-1 of 1 items 1 of 1 pages

pic-11