

# POWER BI INTEGRATION

*Monitor your most critical reporting data in Web or Mobile app.*

---

## USERS GUIDE

---

*August 2019*

*Version 2.1.4*

Kyrylo Kostiukov

kostiukov@bimproject.net

## COPYRIGHT NOTICE

THIS DOCUMENTATION AND THE DESCRIBED SOFTWARE ARE COPYRIGHTED, 2018, BY KYRYLO KOSTIUKOV. ALL RIGHTS RESERVED. KYRYLO KOSTIUKOV RESERVES THE RIGHT TO MAKE IMPROVEMENTS TO THE PRODUCTS DESCRIBED IN THIS MANUAL AT ANY TIME WITHOUT NOTICE.

NO PART OF THIS USER GUIDE MAY BE REPRODUCED, COPIED, TRANSLATED OR TRANSMITTED, IN ANY FORM OR BY ANY MEANS WITHOUT PRIOR WRITTEN PERMISSION OF KYRYLO KOSTIUKOV. INFORMATION PROVIDED IN THIS USER GUIDE INTENDED TO BE ACCURATE AND RELIABLE HOWEVER, KYRYLO KOSTIUKOV ASSUMES NO RESPONSIBILITY FOR ITS USE; NOR FOR ANY INFRINGEMENTS OF RIGHTS OF THIRD PARTIES WHICH MAY RESULT FROM ITS USE.

## PREFACE

---

This document intended to be used to support end users in usage of the Power BI Integration extension for Magento developed by Kyrylo Kostiukov.

Introduction part provides information of features of the extension and general concepts that has to be understood in order to achieve best results from the Power BI Integration and to organize usage of it in the most efficient way.

Configuration part provides information on how to setup initial configuration and where to find additional information on how to enable integration between Power BI Service and Magento Store. It provides details on configuring regular data updates between Magento and Power BI and some tips on how it could be done efficiently.

Sales Cube Usage section describes how to configure and use Sales Cube that comes with Power BI Sales Cube Extension.

Embedding section describes how to use embedding features to embed Power BI content into Magneto Platform.

Next part of the User Guide provides some usage examples that may help users to get valuable results from the provided integration between Magento store and Power BI Online service.

Troubleshooting and Advanced Features describe special features that provide more insights and flexibility for fore intensive usage and for handling special cases that occur while extending the functionality over time.

Final notes give users an overview of what could be the future of the extension and how can users achieve extra functionality and grow their BI infrastructure based on the provided integration.

This document is not covering details related to installation of the extension on Magento platform or development information needed to extend the functionality of the provided extension. Such information might be found in the following documents:

- Power BI Integration Installation Guide;
- Power BI Integration Developer Guide.

Please, contact the extension provider if you want to receive the listed documents.

*This document is not intended to be used to assess the requirements that have to be met by the described extensions.* Such requirements are described in

- Power BI System Requirements Specification.

# CONTENT

<b>QUICK START GUIDES</b> .....	<b>6</b>
POWER BI INTEGRATION START GUIDE .....	6
SALES CUBE SETUP GUIDE.....	8
<b>OBTAINING EXTENSION LICENSE</b> .....	<b>12</b>
<b>INSTALLATION</b> .....	<b>15</b>
MAGENTO 1 EXTENSION INSTALLATION.....	15
MAGENTO 2 EXTENSION INSTALLATION.....	18
<b>BASIC CONFIGURATION</b> .....	<b>21</b>
CONFIGURING EXTENSION LICENSE KEY.....	21
POWER BI APP REGISTRATION .....	23
SCHEDULED UPDATES.....	28
<b>GENERAL CONCEPTS</b> .....	<b>29</b>
TERMINOLOGY .....	29
LOADING PROCESS .....	31
DATASETS AND TABLE PROFILES .....	35
<i>Managing Datasets</i> .....	35
<i>Managing Table Profiles</i> .....	38
MULTIPLE USERS.....	41
<i>Roles</i> .....	41
<i>Simplest use case: single user scenario</i> .....	42
<i>Simplest multiuser use case: multiple independent users</i> .....	42
<i>Use cases of content delivery</i> .....	42
<i>Use cases of content consuming</i> .....	43
<i>Multiple users with different permissions</i> .....	44
<b>TABLE PROFILES BASED ON MAGENTO REPORTS</b> .....	<b>46</b>
CREATION OF TABLE PROFILES BASED ON MAGENTO REPORTS.....	46
CONFIGURING REPORT BASED TABLE PROFILE .....	46
SUPPORTED REPORTS .....	47
MAGENTO STATISTICS REFRESHING .....	49
USAGE OF FILTERS IN MAGENTO REPORTS .....	50
<b>SALES CUBE USAGE</b> .....	<b>51</b>
WHAT IS SALES CUBE DATASET? .....	51
CONFIGURING SALES CUBE ETL SETTINGS .....	51
CONFIGURING SALES CUBE DATASET .....	52
SALES CUBE DATA LOADING .....	53
REPORT TEMPLATES FOR SALES CUBE DATASET .....	54
SALES CUBE DATA CLEAN-UP SCHEDULE.....	54
<b>EMBEDDING POWER BI CONTENT</b> .....	<b>56</b>

ACCESS POWER BI CONTENT WITHIN MAGENTO PLATFORM .....	56
ENABLE AND CONFIGURE EMBEDDED CONTENT .....	57
<i>Pages with Embedded Content</i> .....	57
<i>Embedding Modes</i> .....	60
<i>Configuring the Embedded Content</i> .....	61
<b>ODATA API</b> .....	<b>64</b>
ODATA API OVERVIEW .....	64
ACTIVATING THE ODATA API .....	64
ACTIVATING ODATA ACCESS TO A DATASET .....	65
SCHEDULING DATA ACTUALIZATION FOR ODATA API .....	66
<b>LIMITATIONS</b> .....	<b>68</b>
LIMITATIONS OF REPORT BASED TABLE PROFILES .....	68
LIMITATIONS OF SALES CUBE DATASET .....	69
<b>USAGE EXAMPLES</b> .....	<b>70</b>
CREATING POWER BI REPORTS IN POWER BI SERVICE .....	70
CREATING POWER BI DASHBOARDS .....	71
CREATING REPORTS IN POWER BI DESKTOP .....	72
ADD CALCULATED MEASURES .....	75
ACCESS CONTENT WITH POWER BI MOBILE APP .....	77
SETUP NOTIFICATIONS (ALERTS), REPORTS SCHEDULE .....	79
INTEGRATION WITH Q&A .....	79
INTEGRATION WITH CORTANA .....	80
<b>TROUBLESHOOTING</b> .....	<b>84</b>
DEVELOPMENT CONFIGURATION OPTIONS .....	84
DUMP OF DEBUGGING INFO .....	85
<b>ADVANCED FEATURES</b> .....	<b>86</b>
SCHEMA EDITOR .....	86
REPORT TEMPLATES .....	87
ETL JOBS .....	90
USAGE OF DEV TOOLS .....	92
<i>Assigning datasets and assignment removal</i> .....	92
<i>Sales Cube Dev Tools</i> .....	94
CHANGING THE STRUCTURE OF PERMANENT DATASET .....	95
CONFIGURING ACCESS CONTROL .....	96
<b>FINAL NOTES</b> .....	<b>97</b>
ROADMAP .....	97
EXTENDING THE FUNCTIONALITY .....	97

---

## QUICK START GUIDES

---

This section provides simple step by step instructions on how to setup the infrastructure without going into the details. This might be useful for the initial setup during the evaluation for the users not yet familiar with all concepts.

The instructions in this section and in this document do not include the installation step. This is covered in separate Installation Guide documentation.

---

### POWER BI INTEGRATION START GUIDE

---

#### STEP 1: INSTALLATION

Magento 1 extension files can be downloaded from the [www.bimproject.net](http://www.bimproject.net) after purchasing the license. Read “Installation” section of this documentation to get more information on how to install the extension on Magento 1.

Magento 2 extensions are distributed using private composer repository. Credentials required to access the repository are available on the website details page on [www.bimproject.net](http://www.bimproject.net).

To install the extension on Magento 2 store, perform the following steps (after purchasing of the extension license):

1. login to your account on [www.bimproject.net](http://www.bimproject.net)
2. Go to “My Websites” and configure the website.
3. Go to “My Licenses” and assign the license to the website.
4. On the website settings page you may find access data for the composer repository.
5. Configure your composer and add repository URI to composer.json file:

```
[{
  "type": "composer",
  "url": "http://www.bimproject.net/en/bim_repo/composer/index"
}]
```

6. Add authentication to auth.json:

```
{
  "http-basic": {
    "www.bimproject.net": {
      "username": "<user from website settings page>",
      "password": "<password from website settings page>"
    }
  }
}
```

## 7. Install extension to Magento 2:

```
composer require bim/power-bi-integration-metapackage-magento2:2.1.3
bin/magento setup:upgrade
bin/magento setup:di:compile
```

Read “Installation” section of this documentation to get more detailed information on how to install the extension on Magento Platform.

### STEP 2: SETUP CRONTAB

Run by crontab the following command every minute:

```
# For Magento 2
bin/magento bim:etl:scheduler

# For Magento 1
shell/bim_etl.php scheduler
```

### STEP 3: CONFIGURATION

1. Obtain extension License Key from Website Details page on [www.bimproject.net](http://www.bimproject.net) and enter this key in the configuration of Magento Platform.
2. Register App for Power BI in Azure Active Directory and obtain Client ID and Client Secret. Enter the Client ID and Client Secret to configuration of Magento Platform.

### STEP 4: CONNECT TO POWER BI SERVICE

1. Open Datasets Page (Reports -> Datasets). Click on “Connect to Power BI API” button. Provide user and password for Power BI to connect the user with Magento Platform.

### STEP 5: CREATE PERMANENT DATASET

1. Open Datasets page.
2. Click on “New”.
3. Select “Mixed” if asked to select the dataset type.
4. Fill in the dataset properties and save the dataset.

### STEP 6: CREATE TABLE PROFILE

1. Open Magento Report (for example, Sales Report) and fill in the filters. You can start by filtering out for short period of time and extend it later. Show the report with the filters specified. Full list of supported reports could be found in “Table Profiles based on Magento Reports: Supported reports” section of this documentation.
2. Select Power BI as export option and Press the “Export” button.

## STEP 7: CONFIGURE TABLE PROFILE

1. Navigate to Table Profile properties page using the link in the message displayed after the creation of the Table Profile or by navigating to *Reports -> Table Profiles*
2. Assign the Table Profile to the created dataset.

## STEP 8: PUBLISH DATASET

Publish the created dataset by clicking on “Publish” button on dataset properties page.

## STEP 9: LOAD DATA TO THE DATASET

Load data for the created dataset by clicking on “Load” button on dataset properties page.

At this stage the dataset with data will be available on Power BI Service and is ready to be used to create reports and dashboards.

## STEP 10: CONFIGURING SCHEDULED DATA UPDATES

1. Configure the Live or Regular scheduler. This could be configured in Magento configuration pages on BIM Extensions -> Data Integration (ETL) -> Scheduler tab.
2. Enable automated data actualization for the user on the same page (Reports -> Datasets). Make sure, Magento account is connected to Power BI.
3. Open dataset properties and set Load Mode to the configured schedule (Live or Regular).

Dataset data will be updated automatically.

---

## SALES CUBE SETUP GUIDE

---

### STEP 1: INSTALLATION

Magento 1 extension files can be downloaded from the [www.bimproject.net](http://www.bimproject.net) after purchasing the license. Read “Installation” section of this documentation to get more information on how to install the extension on Magento 1.

Magento 2 extensions are distributed using private composer repository. Credentials required to access the repository are available on the website details page on [www.bimproject.net](http://www.bimproject.net).

To install the extension on Magento 2 store, perform the following steps (after purchasing of the extension license):

1. login to your account on [www.bimproject.net](http://www.bimproject.net)
2. Go to “My Websites” and configure the website.
3. Go to “My Licenses” and assign the license to the website.
4. On the website settings page you may find access data for the composer repository.
5. Configure your composer and add repository URI to composer.json file:



```
[{
  "type": "composer",
  "url": "http://www.bimproject.net/en/bim_repo/composer/index"
}]
```

#### 6. Add authentication to auth.json:

```
{
  "http-basic": {
    "www.bimproject.net": {
      "username": "<user from website settings page>",
      "password": "<password from website settings page>"
    }
  }
}
```

#### 7. Install extension to Magento 2:

```
composer require bim/power-bi-integration-metapackage-magento2:2.1.3
bin/magento setup:upgrade
bin/magento setup:di:compile
```

Read “Installation” section of this documentation to get more detailed information on how to install the extension on Magento Platform.

### STEP 2: SETUP DB CONNECTIONS AND CRONTAB

1. Create new database in MySQL 5.6 or MySQL 5.7.
2. Add connection information to `app/etc/env.php` file of Magento 2 installation under connection name "bim\_dwh" (similar to "default"). On Magento 1 the connection information should be added to `app/etc/local.xml` file.
3. Add connection information for reading data of Magento. Please, copy "default" configuration and add it as "bim\_data\_export".
4. Run by crontab the following command every minute:

```
# For Magento 2
bin/magento bim:etl:scheduler
```

```
# For Magento 1
shell/bim_etl.php scheduler
```

### STEP 3: CONFIGURATION

1. Obtain extension License Key from Website Details page on [www.bimproject.net](http://www.bimproject.net) and enter this key in the configuration of Magento Platform.

2. Register App for Power BI in Azure Active Directory and obtain Client ID and Client Secret. Enter the Client ID and Client Secret to configuration of Magento Platform.
3. Configure the Live scheduler to run every 2 minutes (recommended interval for initial setup). This could be configured in Magento configuration pages on BIM Extensions -> Data Integration (ETL) -> Scheduler tab.

#### STEP 4: CONNECT TO POWER BI SERVICE

1. Open Datasets Page (Reports -> Datasets). Click on "Connect to Power BI API" button. Provide user and password for Power BI to connect the user with Magento Platform.
2. Enable automated data actualization for the user on the same page (Reports -> Datasets).

#### STEP 5: CREATE PERMANENT DATASET

1. Open Datasets page.
2. Click on "New".
3. Select "Sales Cube" if asked to select the dataset type.
4. Fill in the dataset properties and save the dataset.

Do not Load the Dataset yet if the store has lot of data! Initial load might take significant time (up to 1 day).

#### STEP 6: PUBLISH DATASET

Publish the created Sales Cube dataset by clicking on "Publish" button on dataset properties page.

Please, not that this will crease dataset in Power BI, but will not populate it with data yet.

#### STEP 7 (OPTIONAL): ESTIMATE AMOUNT OF DATA

For the stores with log of data, do the estimation of the amount of data in the dataset to make sure, it does not go over the limitations of Power BI:

1. Open *Reports -> BIM Development Tools*
2. Open Sales Cube Capacity estimation tool.
3. Check if there are no tables which exceed the capacity. Also, the page will show estimated time it would take to load all data to Power BI.
4. If there are tables that exceed the allowed capacity, adjust dataset properties to reduce the timeframe or limit amount of stores in the dataset.
5. Once you will make sure that capacity is not increased, proceed further.

#### STEP 8: LOAD DATA TO THE DATASET

Trigger the start of data loading. Depending on the amount of data, it may take different time depending on the amount of data and the Power BI account limitations.

To start loading of the data:

1. Open Dataset properties in Magento Platform.
2. Click to “Start Data Loading” button.
3. Refreshing the page will update information in the data loading status field.

To load the dataset manually click on “Load” button on dataset properties page inside Magento Platform (only recommended for the cases when dataset does not have lot of data).

#### STEP 9: CONFIGURING SCHEDULED DATA UPDATES

Once the initial load is done, user can configure the desired interval for data actualization by configuring Live or Regular schedules and assigning the dataset to this schedule. Additionally it is recommended to configure the clean-up schedule. All created datasets will update full set of data by the clean-up schedule.

1. Configure the desired schedules configuration on the Magento Configuration page, section BIM Extensions -> Data Integration (ETL) -> Scheduler.
2. Change dataset properties to assign dataset to regular or live scheduler if it is not yet done.
3. Make sure, auto load is enabled for the user (is done in previous steps).

#### STEP 10: PUBLISH REPORT TEMPLATES

After the data will be delivered to PBI (or will start to be delivered), user can upload reports and configure embedding functionality.

1. Navigate to Reports -> PBI Templates menu inside Magento Platform.
2. Upload each report related to Sales Cube and assign it to newly loaded dataset.

#### STEP 11: CONFIGURE EMBEDDED CONTENT

1. Open Magento Platform configuration page and navigate to *BIM Extensions -> Power BI Integration -> Embedding* section.
2. Under the Embedding section, enable all embedded content and specify the datasets and reports for embedding. You will need to know Dataset ID, Report ID, Workspace ID from Power BI Service. Easiest way is to find this information in the URL after navigating to particular Power BI content in Power BI Service.

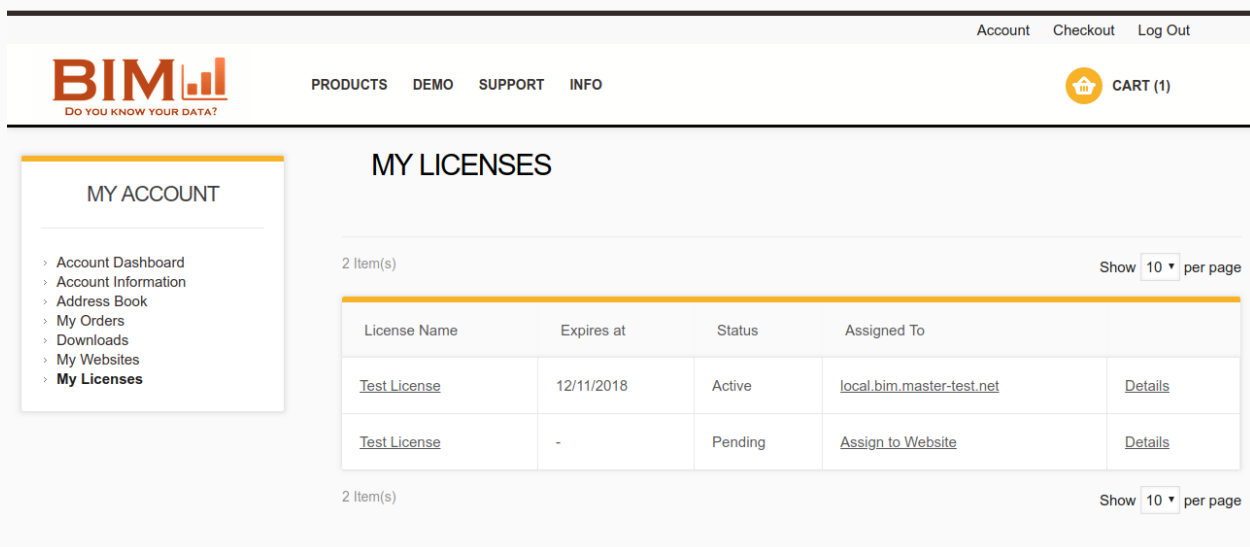
## OBTAINING EXTENSION LICENSE

### STEP 1: PURCHASE THE EXTENSION

Please, use the website [www.bimproject.net](http://www.bimproject.net) to find the license for the Extension matching your needs. Products and their descriptions could be found through the Products page on the website. Please, add the License to your cart and follow the instructions on the website to complete the order.

During making the order you might be asked to confirm that you have read and agree with the license conditions and other related documents. Once the order will be placed, license will be added to the list of licenses, but will stay in status “Pending” before it will be confirmed.

List of licenses can be found in Account page on [www.bimproject.net](http://www.bimproject.net) under menu “My Licenses”:



Account Checkout Log Out

**BIM**  
DO YOU KNOW YOUR DATA?

PRODUCTS DEMO SUPPORT INFO

CART (1)

### MY ACCOUNT

- > Account Dashboard
- > Account Information
- > Address Book
- > My Orders
- > Downloads
- > My Websites
- > **My Licenses**

### MY LICENSES

2 Item(s) Show 10 per page

License Name	Expires at	Status	Assigned To	
<a href="#">Test License</a>	12/11/2018	Active	<a href="#">local.bim.master-test.net</a>	<a href="#">Details</a>
<a href="#">Test License</a>	-	Pending	<a href="#">Assign to Website</a>	<a href="#">Details</a>

2 Item(s) Show 10 per page

Picture: BIM Website – My Licenses page

After the order has been made it should be internally reviewed. Once the order will pass the review, the invoice will be issued and the license will become “Active”:


### STEP 2: CONFIGURE WEBSITE

Each Extension License needs to be assigned to particular Website. List of Websites can be managed on the Account page on [www.bimproject.net](http://www.bimproject.net), menu item “My Websites”:

Account   Checkout   Log Out

**BIM**  
DO YOU KNOW YOUR DATA?

PRODUCTS   DEMO   SUPPORT   INFO

 CART (1)

---

**MY ACCOUNT**

---

- > Account Dashboard
- > Account Information
- > Address Book
- > My Orders
- > Downloads
- > **My Websites**
- > My Licenses

## MY WEBSITES

[add new website](#)

1 Item(s) Show 10 per page

Website Name	Platform	License Key	
<a href="#">local.bim.master-test.net</a>	M1	<a href="#">Available</a>	<a href="#">Details</a>   <a href="#">Delete</a>

1 Item(s) Show 10 per page

*Picture: BIM Website – My Websites page*

Press button “Add new Website” to start adding new Website configuration:

**MY ACCOUNT**

---

- > Account Dashboard
- > Account Information
- > Address Book
- > My Orders
- > Downloads
- > My Websites
- > My Licenses

## NEW WEBSITE ENTRY

**Website Configuration**

Website Name \*

Platform \*

Platform Version \*

Please, enter the edition (enterprise/community) and version of your platform software. E.g. "CE 1.9.3".

Website Domain Names \*

List of up to 5 domain names. Each domain should be entered in separate line (e.g. www.bimproject.net).

\* Required Fields

[save website](#)

*Picture: BIM Website – New client website creation form*

### STEP 3: ASSIGN LICENSE TO WEBSITE

Once Website is configured, it will be possible to assign the purchased license to this Website.

Navigate to “My Licenses” on Accounts page and click “Assign to Website”. This will open the page with Website Assignment Form:

## LICENSE DETAILS

### License Information

General	License Validity
<b>License Name:</b> <a href="#">Test License</a>	<b>Status:</b> Active
<b>Platform:</b> M1	<b>License Type:</b> Perpetual
<b>Order ID:</b> <a href="#">212006428</a>	<b>Active From:</b> 16/12/2018
	<b>Active To:</b> 21/05/2035

### Website

Associated Website Configuration

License Assignment Form

Website Name \*

Sample Website ▼

Buy submitting this form you confirm that this license will be used for the specified website. You will not be able to change this assignment in the future. Please contact us in case of any questions.

submit

### Documents

License Agreement

[Click to open license agreement](#)

*Picture: BIM Website –License details page*

Select the website and click on “Submit” to finish the assignment. After the first time, license gets purchased for particular Website, there will be License Key generated automatically. Since this moment the license starts its validity period.

# INSTALLATION

---

## MAGENTO 1 EXTENSION INSTALLATION

---

The Magento 1 installation process consists of the following steps:

1. Download the extension package.
2. Install the package to Magento Instance.
3. Configuring database connections for Sales Cube (in case if Sales Cube functionality comes with the extension).
4. Setup crontab configuration for the scheduler.

The downloaded Magento 1 extension package could be installed in different ways:

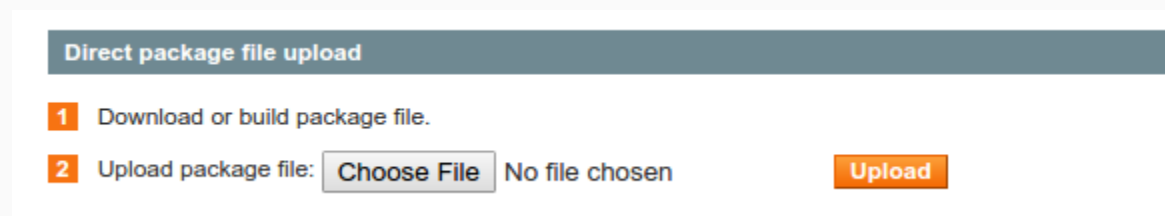
- Option 1: using Magento Connect Manager
- Option 2: manually using FTP/SSH access

### DOWNLOAD THE MAGENTO 1 EXTENSION PACKAGE

After purchasing the license (read “Obtaining Extension License” section of this documentation) the extension package should be available on the Downloads page. To open the Downloads, navigate to “Downloads” menu on the Account page on [www.bimproject.net](http://www.bimproject.net). Just select the latest version and the appropriate package to download.

### OPTION 1: INSTALLATION WITH MAGENTO CONNECT MANAGER

In order to install extension package you need Magento Admin account with access to Magento Connect Manager. Log into your stores admin console and navigate to *System* → *Magento Connect* → *Magento Connect Manager*. At this point you will be prompted to re-enter your administrator credentials. Once you are logged into the manager you will see form for uploading the package archive file under the section "Direct package file upload". Simply choose the downloaded package file from your file system and press "Upload" to start the installation process.



Direct package file upload

- 1 Download or build package file.
- 2 Upload package file:  No file chosen

Picture: Magento 1 package installation dialogue

Sometimes if some error occurs during the installation process, your system may stay locked in maintenance mode. In order to turn off the maintenance mode you can remove maintenance file from core directory of your Magento store.

#### OPTION 2: MANUAL INSTALLATION (RECOMMENDED)

Extract downloaded extension package archive file on some temporary folder on your local machine or on the remote server with Magento Platform instance. Using your ftp client or ssh (scp) copy all of the extracted files onto the store root of the Magento instance.

After copying the files go to the admin page *System* → *Cache Management* and click on "Flush Cache Storage".

#### SALES CUBE DATABASE CONFIGURATION, M1

Create database for DWH storage in MySQL 5.6 or MySQL 5.7.

Add "**bim\_dwh**" connection information to local.xml.

"**bim\_dwh**" should point to the newly created MySQL database for DWH storage.



```

<?xml version="1.0"?>
<config>
  <global>
    <install...>
    <crypt...>
    <disable_local_modules>>false</disable_local_modules>
    <resources>
      <db>
        <table_prefix><![CDATA[]]></table_prefix>
      </db>
      <default_setup>
        <connection>
          <host><![CDATA[localhost]]></host>
          <username><![CDATA[magento_user]]></username>
          <password><![CDATA[*****]]></password>
          <dbname><![CDATA[magento]]></dbname>
          <initStatements><![CDATA[SET NAMES utf8]]></initStatements>
          <model><![CDATA[mysql4]]></model>
          <type><![CDATA[pdo_mysql]]></type>
          <pdoType><![CDATA[]]></pdoType>
          <active>1</active>
        </connection>
      </default_setup>
      <bim_dwh>
        <connection>
          <host><![CDATA[localhost]]></host>
          <username><![CDATA[magento_user]]></username>
          <password><![CDATA[*****]]></password>
          <dbname><![CDATA[magento_dwh]]></dbname>
          <model><![CDATA[mysql4]]></model>
          <type><![CDATA[pdo_mysql]]></type>
          <active>1</active>
        </connection>
      </bim_dwh>
    </resources>
  </global>
</config>

```

*Picture: Configuring database connections for Magento 1*

### SETUP CRONTAB SCHEDULER, M1

Run by crontab the following command every minute:

```
shell/bim_etl.php scheduler
```

Below you will find the example of the corntab configuration (type “crontab -e” in command line to start editing the crontab configuration).

```
* * * * * php /path/to/magento/shell/bim_etl.php scheduler >>/path/to/magento/crontab-bim.log 2>&1
```

*Picture: Configuring crontab scheduler job for Magento 1*

## MAGENTO 2 EXTENSION INSTALLATION

---

Magento 2 has reliable and robust installation process based on composer. Composer is a dependency management tool well supported by the community of PHP developers.

Since composer not only makes the installation and upgrade process easier, but also helps to ensure the compatibility of the components, it is the only supported installation option for Magento 2 Extensions.

The Magento 2 installation process consists of the following steps:

1. Setup access to composer repository on [www.bimproject.net](http://www.bimproject.net) .
2. Install the composer package with the extension to Magento 2 Instance.
3. Configure database connections for Sales Cube (in case if Sales Cube functionality comes with the extension).
4. Setup crontab configuration for the scheduler.

### SETUP ACCESS TO COMPOSER REPOSITORY, M2

#### GET REPOSITORY ACCESS INFORMATION

On the website settings page you may find access data for the composer repository. To open website settings, go to Account page on [www.bimproject.net](http://www.bimproject.net), menu item “My Websites” and select a website from the list. Click on “Show repository access data” to display the repository URI, repository user name and password.

#### ADD REPOSITORY TO COMPOSER.JSON

Add repository access information to *composer.json* repositories. Open *composer.json* file in base folder with Magento Instance installation and add the following section to “repositories”:

```
{
  "type": "composer",
  "url": "http://www.bimproject.net/en/bim_repo/composer/index"
}
```

#### ADD AUTHENTICATION INFORMATION TO AUTH.JSON

Create file *auth.json* inside base directory of Magento Instance installation if it was not created before and add the following content to it:

```
{
  "http-basic": {
    "www.bimproject.net": {
      "username": "<user from website settings page>",
      "password": "<password from website settings page>"
    }
  }
}
```

Alternatively, a system wide *auth.json* could be used in composer home directory (~/.composer).

#### INSTALLING COMPOSER PACKAGE WITH AN EXTENSION, M2

Depending on the purchased Power BI Integration Extension package, there will be a composer metapackage available for the installation.

The name of the metapackage could be found in the “Licensed Software Details” list on website details page (“My Websites” on [www.bimproject.net](http://www.bimproject.net) -> “Details”).

bim/pbi-sales-cube-metapackage-magento2:~2.1.2	demo	19.01.2019
--	------	------------

*Picture: Meta-package name on client website details page on BIM Website*

Following is example of commands that will trigger the installation of the extension:

```
composer require bim/pbi-sales-cube-metapackage-magento2:~2.1.2
bin/magento setup:upgrade
bin/magento setup:di:compile
```

“bim/pbi-sales-cube-metapackage-magento2:~2.1.2” would need to be replaced with the available metapackage name and their version.

#### SALES CUBE DATABASE CONFIGURATION, M2

Create database for DWH storage in MySQL 5.6 or MySQL 5.7.

Add “bim\_dwh” and “bim\_data\_export” connection information to *app/etc/env.php* file of Magento installation.

“**bim\_dwh**” should point to the newly created MySQL database for DWH storage.

“**bim\_data\_export**” should point to Magento database. This connection might also be pointing to a read-only replica of the main Magento database for decreasing the load on the productive Magento database. It could be just a copy of Magento database settings for the initial setup.

```
<?php
return array(
    'db' => array(
        'connection' => array(
            'default' => array(
                'host' => 'localhost',
                'dbname' => 'magento2',
                'username' => 'magento2_user',
                'password' => '*****',
                'model' => 'mysql4',
                'engine' => 'innodb',
                'initStatements' => 'SET NAMES utf8;',
                'active' => '1',
            ),
            'bim_data_export' => array(
                'host' => 'localhost',
                'dbname' => 'magento2',
                'username' => 'magento2_user',
                'password' => '*****',
                'model' => 'mysql4',
                'engine' => 'innodb',
                'initStatements' => 'SET NAMES utf8;',
                'active' => '1',
            ),
            'bim_dwh' => array(
                'host' => 'localhost',
                'dbname' => 'magento2_dwh',
                'username' => 'dwh_user',
                'password' => '*****',
                'model' => 'mysql4',
                'engine' => 'innodb',
                'initStatements' => 'SET NAMES utf8;',
                'active' => '1',
            ),
        ),
    ),
);
```

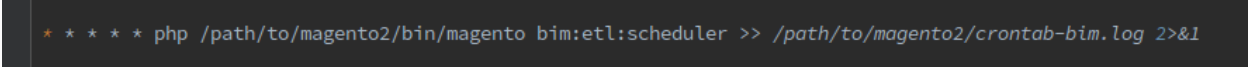
Picture: configuring database connections on Magento 2

## SETUP CRONTAB SCHEDULER, M2

Run by crontab the following command every minute:

```
bin/magento bin:etl:scheduler
```

Below you will find the example of the corntab configuration (type “crontab -e” in command line to start editing the crontab configuration).



```
* * * * * php /path/to/magento2/bin/magento bin:etl:scheduler >> /path/to/magento2/crontab-bim.log 2>&1
```

*Picture: configuring crontab scheduler job on Magento 2*

---

## BASIC CONFIGURATION

---

This section covers System Configuration parameters related to Power BI Integration Extensions.

After being installed a Power BI Integration Extension needs to be configured.

Configuration section for Power BI Integration could be found by navigating to *Stores -> Configuration* menu in Magento 2 or *System -> Configuration* in Magento 1.

---

## CONFIGURING EXTENSION LICENSE KEY

---

### DOWNLOAD/COPY LICENSE KEY

At this stage the License should already be purchased and assigned to a Website using Account pages on [www.bimproject.net](http://www.bimproject.net).

Once the License was assigned to a Website, the License Key got generated and is available for download. License Key can be downloaded on the Website’s Details page. To open the Website Details, navigate to “My Websites” menu (on the Account page on [www.bimproject.net](http://www.bimproject.net)) and click on “Details” link in the list of Websites.

License Key can be obtained on the Website Details page:

# WEBSITE DETAILS

## Website Information

<b>Website Configuration</b> <a href="#">EDIT</a>	<b>Domain Names</b> <a href="#">EDIT</a>
<b>Name:</b> Sample Website	bimproject.net stage.bimproject.net
<b>Platform:</b> M1, EE 1.4	dev.bimproject.net www.bimproject.net

## License Key

<b>Online License Key</b> <a href="#">EDIT</a>
You may copy online license key from the text field below: <a href="#">Show online license key</a>

<b>Offline License Key</b> <a href="#">EDIT</a>
You may copy offline license key from the text field below: <a href="#">Show offline license key</a> or click here to download the license key as a file: <a href="#">Download License Key</a>

## Repository Access

[Show repository access data](#)

*Picture: client website details page on BIM Website*

Online License Key is a short version of the key. Website [www.bimproject.net](http://www.bimproject.net) will be used to obtain the complete License Key by the Extension. If this communication between Magento Platform and [www.bimproject.net](http://www.bimproject.net) is not desired, Offline License Key might be used. Offline License Key can be downloaded on Website Details page and uploaded to Magento Platform.

If the License comes with documentation files or any other files to download, they will be available on "Downloads" page. Extension code can be obtained using composer repository. Access data to the repository will also be available on the Website's Details page.

## CONFIGURE LICENSE KEY

Once the License Key is obtained, it can be entered in Magneto Configuration. The configuration can be opened by navigating to Stores -> Configuration in Magento 2 (System -> Configuration in Magento 1). On the Configuration page navigate to BIM Extensions -> General -> License:

Picture: License key configuration for BIM extensions

## POWER BI APP REGISTRATION

Power BI Extension is integrated with Azure Active Directory (Azure AD) to provide secure sign in and authorization between Magento Platform and Power BI API. The following is needed in order to start using Power BI Integration Extension: Azure Active Directory tenant, an organizational user, and a Power BI service account. The extension uses a Client ID to identify itself to Azure AD. Also a Client Secret Key needs to be created.

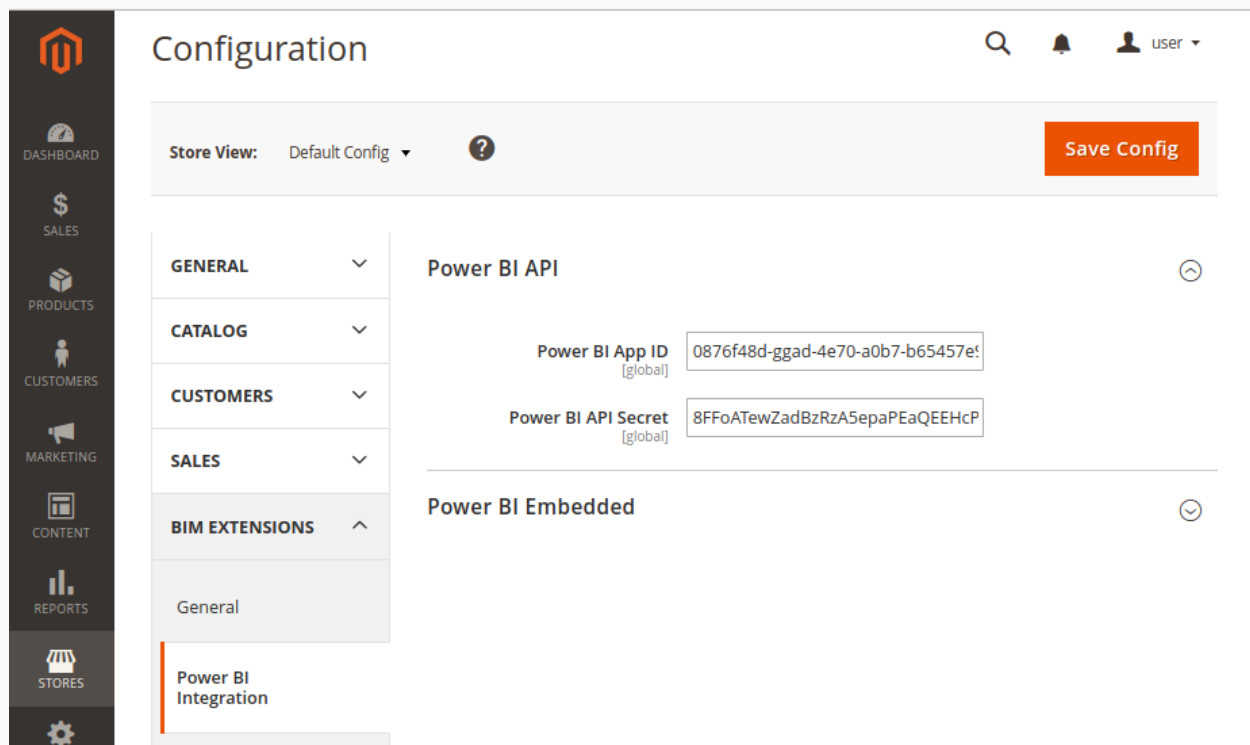
Microsoft has developed a tool to simplify the process of Power BI App registration: <https://dev.powerbi.com/apps>. Please, enter the following options during the usage of the tool:

- Use “Server-side web application (for web apps or web APIs)” as Application Type.
- Enter domain of your store as Home Page URL.
- Use the URL of your Magento Admin path supplied with “\*” suffix as Redirect URL  
Example: <https://your-website-domain.com/>\*
- Select all permissions.

If the tool does not work, you may look in the Power BI documentation from Microsoft for the information on how to configure the Power BI App and obtain a Client ID and a Client Secret Key:

<https://docs.microsoft.com/en-us/power-bi/developer/register-app>

The obtained Client ID and Client Secret Key needs to be configured for the Power BI Integration Extension on Magento Configuration page (Stores -> Configuration in Magento 2, System -> Configuration in Magento 1), section BIM Extensions -> Power BI Integration -> Power BI API:



Picture: Power BI Client ID and Client Secret configuration

Setting up Power BI API might take some time, especially if an organization were not using Active Directory before. Usually people face some problems due to incorrectly configured Redirect URI. Please, use the following value as Redirect URI:

`https://your-website-domain.com/*`  
(There is an asterisk at the end of a URI!)

For the rest of information, please, refer to the Microsoft documentation as main source of the information.

As an additional support, for a quick reference, we list the steps you may need to perform in order to obtain a Client ID and a client secret key for Power BI API.



## STEP 1: CREATE AN AZURE ACCOUNT

You should be able to create an Azure account for free. You pay only for Services you are using or for an optional support subscription.

Here is the URL where you may start with registration of an account:

<https://azure.microsoft.com/en-in/free/>

## STEP 2: CREATE AN AZURE ACTIVE DIRECTORY TENANT

Please, follow the instructions from Power BI documentation in order to create an Azure AD tenant:

<https://powerbi.microsoft.com/en-us/documentation/powerbi-developer-create-an-azure-active-directory-tenant/>

Please, note that you do not need to activate Power BI Pro account as it is stated in Microsoft documentation since you do not have to use Power BI embedded in order to setup the API.

## STEP 3: CREATE SOME USERS IN YOUR AZURE ACTIVE DIRECTORY TENANT

You may want to create some users if you did not do that before. Each user who will use Power BI Service or Power BI Mobile App may need to have separate Power BI user.

And if you want to share content and use Power BI Workspaces, you may add Power BI Pro licenses. Pro license is needed for the user who will create the Power BI Workspaces and populate it with content and for the users who will get access to the Workspaces.

## STEP 4: REGISTER POWER BI APPLICATION

You may use new Power BI App Registration Tool for that or using Azure portal.

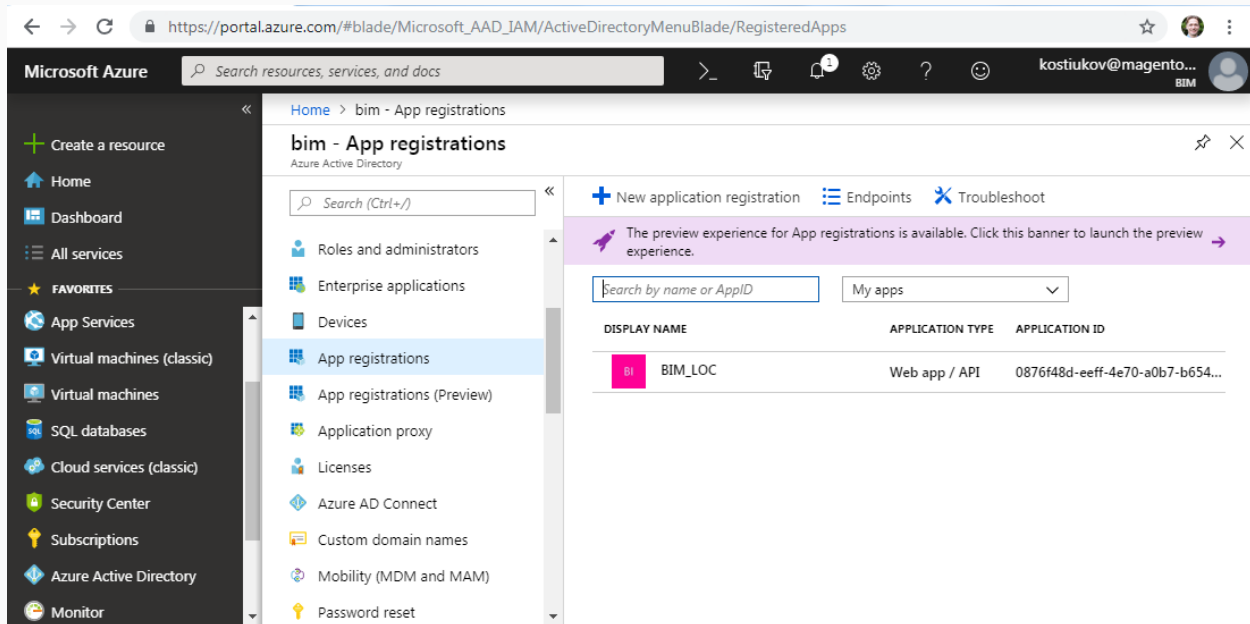
<https://powerbi.microsoft.com/en-us/documentation/powerbi-developer-register-app/>

Follow instructions related to *Server-side Web app* or *Web Applications* instead of registering a Native app.

Use Redirect URL as your domain name (of the backend system) with asterisks in the end.

**Redirect URI:** <https://your-website-domain.com/>\*

In result, the application should be available in the list of applications in Azure AD:



Picture: example of a App registration record in Azure AD

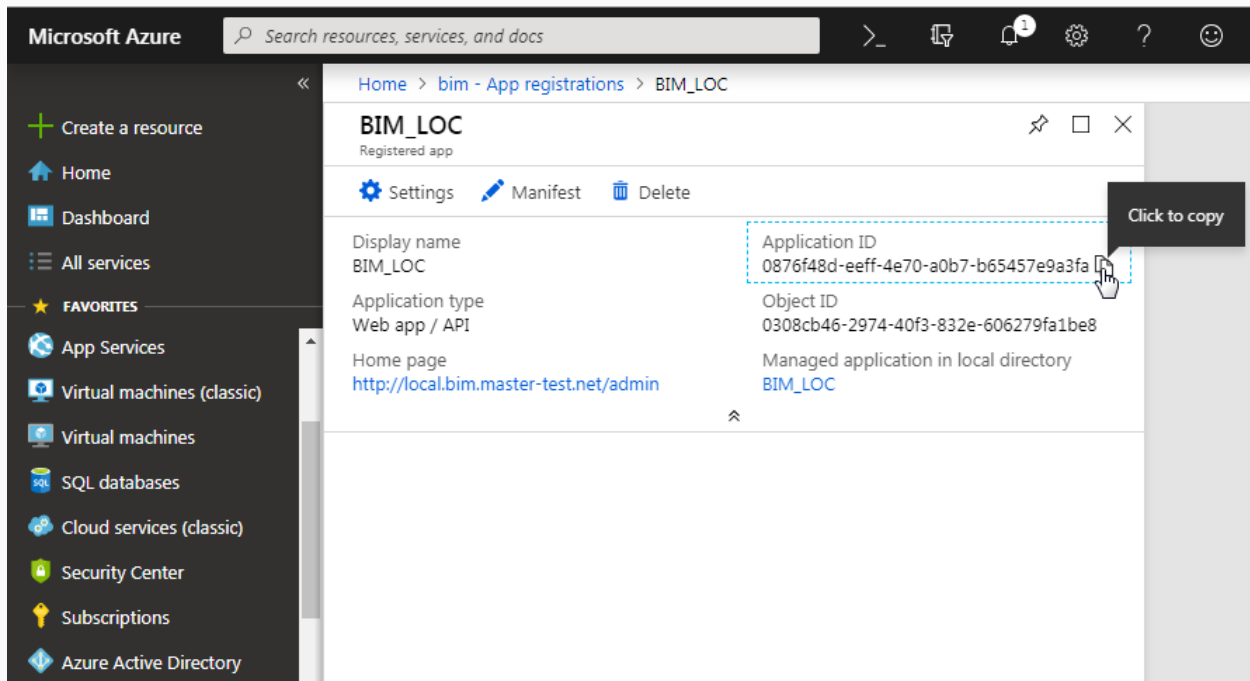
#### STEP 5: GET CLIENT ID AND SECRET KEY

Please, refer to the “How to get the client id” section of the documentation to get the Client ID:

<https://powerbi.microsoft.com/en-us/documentation/powerbi-developer-register-app/#how-to-get-the-client-id>

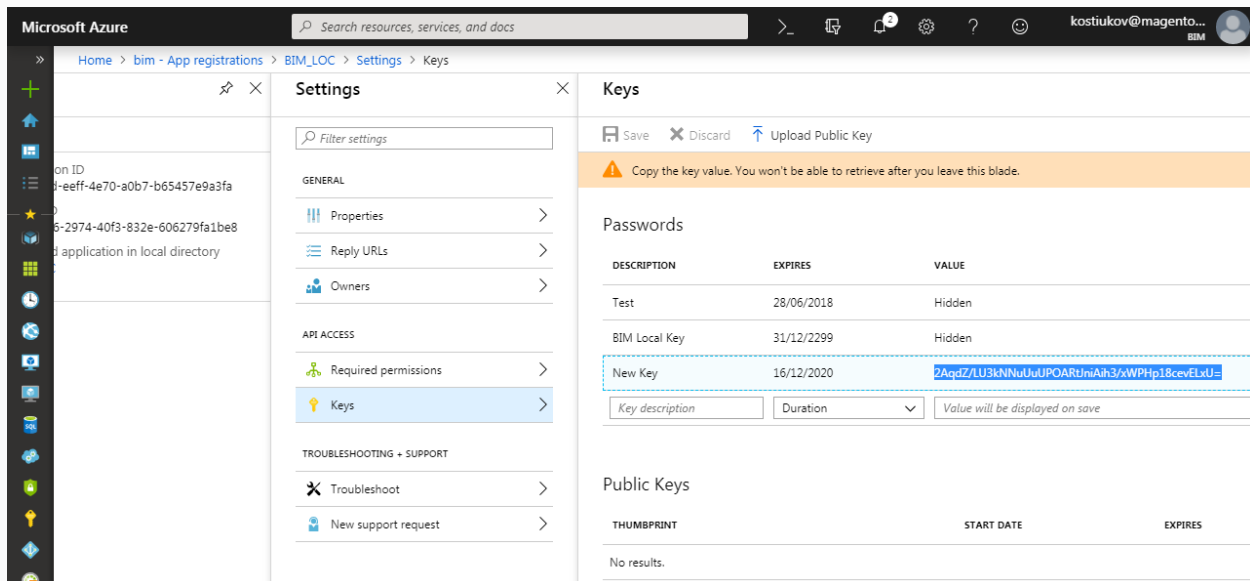
To get the client secret key, you may need just few additional steps to be performed on the Azure portal: after you have selected the application, click on *Keys* menu under *API ACCESS* section and add new key there.

Client ID inside Azure AD:



Picture: obtaining Client ID in Azure AD

### Client Secret inside Azure AD:

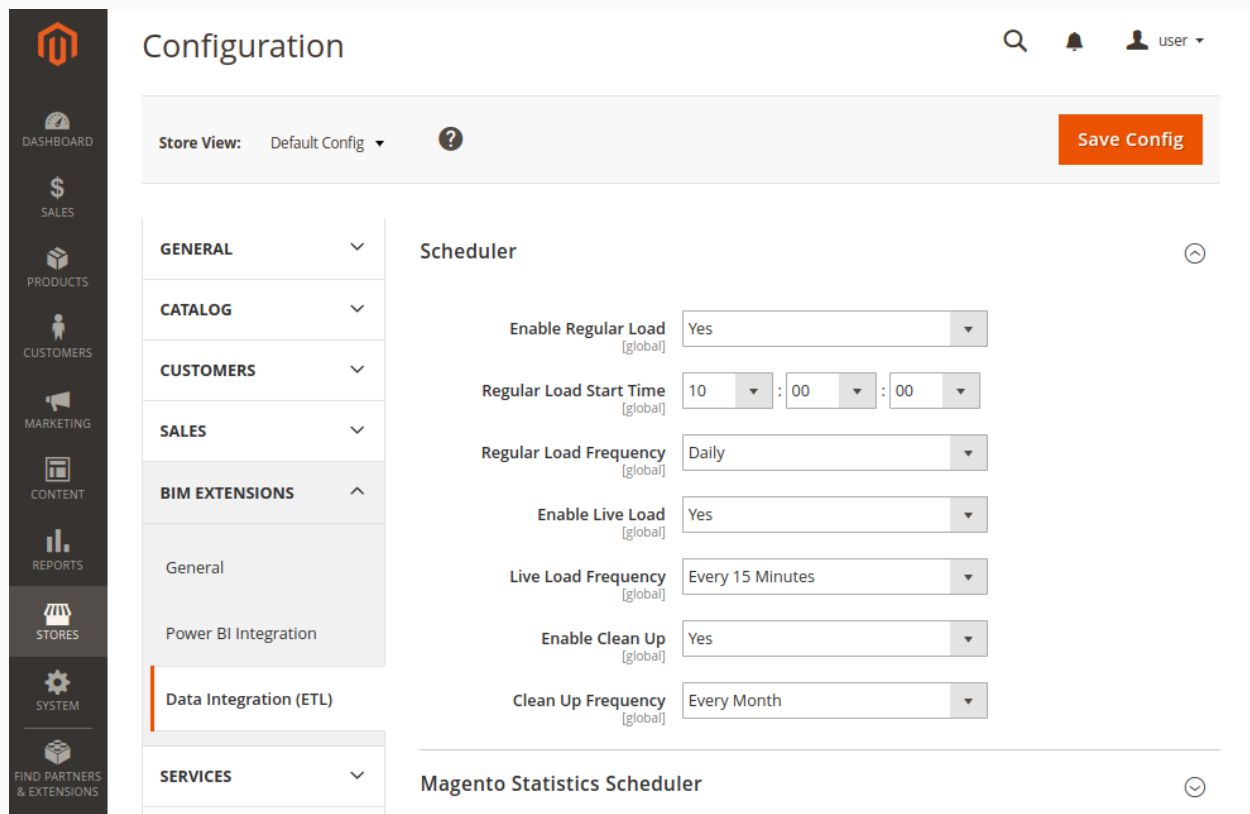


Picture: obtaining Client Secret in Azure AD

## SCHEDULED UPDATES

Power BI Integration Extension comes with a scheduler that allows performing periodically automated data actualization. To configure the schedule, navigate to Magento Configuration page using menu Stores -> Configuration in Magento 2 (System -> Configuration in Magento 1).

On the Magento Configuration page navigate to BIM Extensions -> Data Integration (ETL) -> Scheduler.



Picture: Scheduler configuration in Magento Platform

This section allows users to configure:

1. Regular schedule for jobs that needs to be performed once per day or less often.
2. Live schedule for jobs that need to be executed several times per day.
3. Clean Up schedule for executing data clean up jobs from time to time.

Below you will find details about each of the options.

**Enable Regular Load** option allows enabling/disabling execution of the jobs by regular schedule.

**Regular Load Start Time** is the time of the day when to execute regular jobs. Usually users may want to set this time to a period with lowest activity, at night.

**Regular Load Frequency** allows setting the frequency for regular jobs: daily, weekly, monthly.

**Enable Live Load** option allows enabling/disabling execution of the jobs by life schedule.

**Live Load Frequency** allows setting the frequency for live schedule: from once per minute to once in 2 hours.

**Enable Clean Up** option allows enabling/disabling execution of the clean-up jobs.

**Clean Up Frequency** allows setting the frequency for clean-up jobs: weekly, monthly or once in two months.

---

## GENERAL CONCEPTS

---

It is important to understand how Power BI REST API is working in order to know how to organize proper development of reports and dashboards. It is important to leave the possibility of making changes in the future while being able to keep stable data delivery to dashboards and reports.

The Power BI Integration Extensions were developed with intention to simplify usage of Power BI API as much as possible. From one side, the goal is to protect users from making mistakes that may lead to broken reports or dashboards. From another side, user should be able to deliver data to Power BI as easy and fast as possible. Reaching this goal is not a trivial issue and desired level of flexibility not always can be reached due to limitations of Power BI APIs.

Some additional concepts were introduced to find a way around those limitations. Those are among the others the *Temporary* and *Permanent* datasets.

Some of the basic terminology is described in this section. This includes terminology from Power BI Service and from Power BI Integration Extensions.

---

## TERMINOLOGY

---

---

### POWER BI SERVICE CONCEPTS

---

The 4 major building blocks of Power BI are: *dashboards*, *reports*, *workbooks*, and *datasets*. And they're all organized into workspaces. Datasets are organized in tables. These concepts are described in Power BI Service Documentation:

<https://docs.microsoft.com/en-us/power-bi/service-basic-concepts>

Those concepts will be referred as **Power BI Workspaces, Power BI Dashboards, Power BI Reports, Power BI Datasets** and **Power BI Tables**.

**Power BI Datasets** is a collection of data organized in one or more tables. It is not only holding the data itself, but also keeps information about relationships between tables and other information that helps Power BI Service to better interpret the data. It is important to note that deletion of the dataset might result in deleted reports and dashboards this dataset relates to.

**Power BI Table** is a table-like storage of data with predefined columns of specific data types.

**Power BI Report** is a visual representation of data from a Power BI Dataset. Unlike Power BI Dashboards, Power BI Report only shows data from one Power BI Dataset.

**Power BI Dashboard** is a collection of visualizations organized as **Power BI Tiles**. Unlike Power BI Report, Power BI Dashboard may show data from multiple datasets.

**Power BI Tile** is a single visual representation of data, placed on a dashboard.

---

## POWER BI INTEGRATION CONCEPTS

---

The main goal of the integration is to create Power BI Datasets inside Power BI Service. In order to do so, user first prepares a Dataset in Magento Platform.

**Magento Dataset** or simply **Dataset** is a representation of a Power BI Dataset inside Magento Platform. There are different types of datasets: Mixed Dataset Type, Sales Cube Dataset Type or other custom types. Datasets also can be classified as Temporary or Permanent.

**Dataset Publishing** is the process of Power BI Dataset creation based on Magento Dataset. Initially Dataset is created and configured inside Magento Platform and when it is ready, it can be published to create the Power BI Dataset inside the Power BI Service. Publishing delivers only prepares the storage for the data without sending the data itself.

**Dataset Republishing** is the process of updating the Power BI Dataset with the recent changes inside Magento Dataset structure. Not all changes might be applied. For example, adding new tables to Power BI Dataset is not possible at the moment (due to limitations of Power BI REST API). Therefore, republishing sometimes may result in Power BI Dataset deletion and re-creation (with loss of all related reports) and is only allowed for **Temporary Datasets**. In some cases the Power BI Integration Extension might try to update existing Power BI Dataset without deletion of the content (updating calculated measure definition or change of column type), but this is relevant only for advanced customizations or for applying the Power BI Integration Extension updates.

**Dataset Loading** or **Table Profile Loading** is the process of loading of the data from Magento Platform to Power BI Dataset. It does not change the amount of tables in a dataset or their names. Any Table Profile or Dataset can be loaded multiple times. If Table Profile is assigned to a Permanent Dataset, it can be loaded only after publishing of the Dataset. If Load Profile is assigned

to a Temporary Dataset, dataset publishing will be triggered during the load automatically if needed.

**Mixed Dataset Type** is a type of the dataset that may include Table Profiles. It is called as mixed because data of such dataset may come from different places.

**Table Profile** is a representation of a Power BI Table inside Magento Platform. It usually holds the information needed to link particular data from Magento Platform to Power BI Table.

**Sales Cube Dataset Type** is a complex dataset type with predefined tables and relationships that form comprehensive set of data around Sales process. Table Profiles cannot be assigned to this dataset type.

**Temporary Dataset** is a dataset that can re-published. Re-publishing might cause loss of reports and dashboards associated with the dataset. Current implementation supports re-publishing only through delete/create operations. It means that internally old dataset should be deleted from Power BI Service and new dataset has to be published again. Temporary datasets are not intended to be used to create reports and dashboards that will stay to be used in the future.

**Permanent Dataset** is a dataset that might be published only once. This is intended to protect user from unintentional loss of reports and dashboards. Permanent Dataset cannot be re-published. Once Permanent Dataset gets published, it should not be possible to make modifications to the Dataset structure (amount of tables and their names).

---

## BRINGING IT ALL TOGETHER

---

Every Power BI Service user has by default its own workspace called “My Workspace”. All content by default will be saved in that workspace: datasets, reports, dashboards.

Power BI Pro users can create additional workspaces and move there part of the content: datasets, reports, dashboards. Such workspaces can then be shared with other Power BI Pro users. This gives possibility to share content through the Power BI Service.

Reports and dashboards are created based on datasets. In order to create a dataset in Power BI Service with Magento data, user should perform the following:

1. Create and configure Dataset in Magento Platform.
2. Add Table Profiles to the Dataset if needed.
3. Publish Dataset to Power BI.
4. Load data to the Power BI Dataset.

As result of this action, Dataset from Magento Platform will result in dataset inside Power BI. If the dataset have Table Profiles, they will translate into Power BI Tables inside.

---

## LOADING PROCESS

---

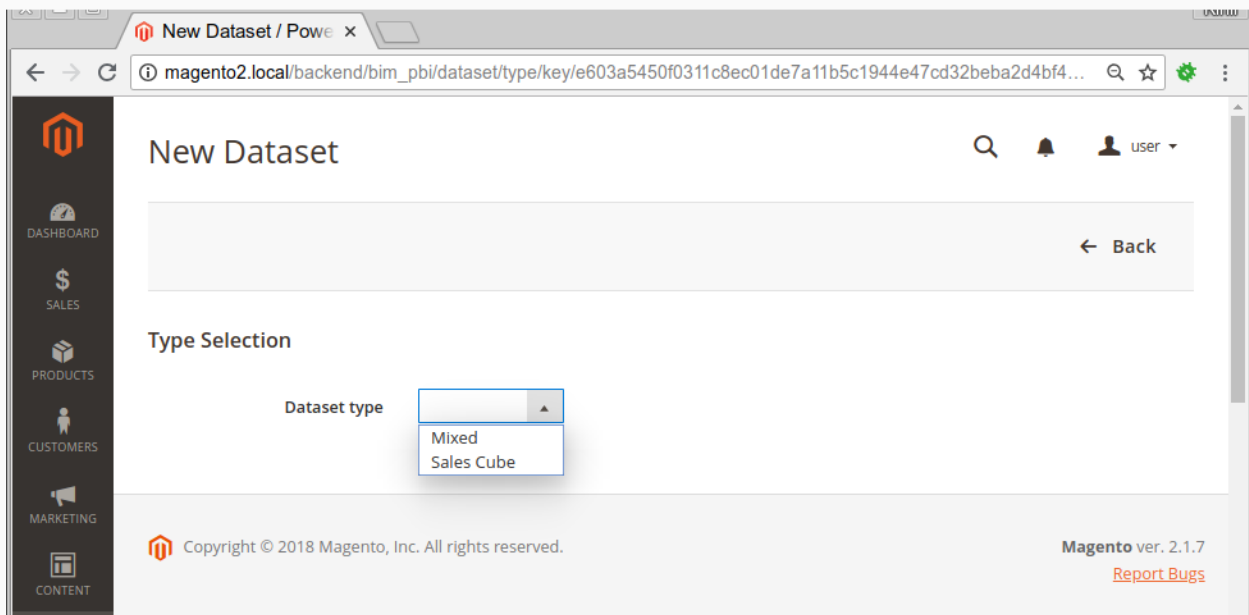
Loading process describes the process of transferring data from Magento Platform to Power BI Service with usage of Power BI Integration Extension from user's perspective.

## STEP 1: CREATION OF A DATASET

First, user creates a Dataset. Some datasets are self-contained and does not require any additional actions to include data into the dataset. Some Dataset require also Table Profiles to be created for the dataset to have tables with data in it.

To create dataset user goes to Reports -> Datasets page in Magento Admin. And on the Datasets page, user should press "Add New Dataset".

If Extension supports more than one dataset type, user will be offered to select the type of the dataset he wants to create:



Picture: Dataset type selection dialogue

Mixed Dataset Type would require Table Profiles assigned to it. If the dataset type does not include tables by default, user creates Tables Profiles for the dataset. To create a Table Profile based on a Magento Report, user navigates to a Magento Report page, fills in the values for the report options and triggers the report data to be shown. After that user should select "Power BI" in the list of export options and click "Export".

Sales Cube Dataset Type does not need Table Profiles.



After selecting dataset type, user will be redirected to dataset creation form:

The screenshot shows the 'New Dataset' creation form. On the left is a dark sidebar with icons and labels for various sections: DASHBOARD, SALES, PRODUCTS, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, SYSTEM, and FIND PARTNERS. The main content area has a white background with the title 'New Dataset' at the top left. To the right of the title are search, notification, and user profile icons. Below the title is a light gray bar containing a 'Back' button and a prominent orange 'Save Dataset' button. The 'Dataset Information' section follows, with several form fields: 'Dataset name \*' (text input), 'Workspace name' (dropdown menu showing 'My Workspace'), 'Load Mode' (dropdown menu showing 'Manual'), 'Is Temporary' (dropdown menu showing 'No'), 'Dataset type' (text input showing 'Mixed'), and 'Remote Dataset ID' (text input).

*Picture: dataset creation form*

**Dataset Name** option is the name that will be visible in the list of datasets in Magneto Platform and later in Power BI Service after the dataset will be created.

**Workspace name** option contains the Power BI Workspace where the Dataset will be published to. Other workspaces than “My Workspace” can be shared with other Power BI users, but in order to use this functionality Power BI Pro license needs to be purchased from Microsoft.

**Load Mode** option can be “Manual” or contain a schedule name. Default schedules are “Live” and “Regular”. If the name of the schedule is selected, the content of the dataset will be automatically updated according to the schedule configuration. In order to be loaded automatically, Dataset needs to be published first.

**Is Temporary** option allows to made dataset Temporary or Permanent. Usage of temporary dataset may result in loss of the created reports based on that dataset, but this type of the dataset is more usable for the quick preview of the data inside Power BI. Permanent dataset can be published only once.

**Dataset Type** option cannot be changed at this stage. It shows what type of the dataset user is going to create now.

**Report Dataset ID** field is not relevant for dataset creation. It will include Power BI Dataset ID once the Dataset will be published.

---

## STEP 2: DATA PREVIEW IN TEMPORARY DATASET

---

As the next step the user may transfer the report to Power BI Service and see if expected results might be achieved. To do so, user may navigate to created Table Profile and press “Load to Power BI” button.

Since created Table Profile is assigned to a Temporary Dataset, the dataset will be re-published automatically to add new table and the data will be transferred to populate the table with data inside Power BI Service.

Note that Temporary Dataset should not be used for reports or dashboards development if you plan to save them for the future. Every time Temporary Dataset gets republished, related data visualizations will be deleted.

---

## STEP 3: DATASET PUBLISHING

---

If you are ready with your Load Dataset and you have assigned all Load Profiles you want to use with this dataset, you may publish it.

Publishing of the dataset will create Power BI Dataset in Power BI Service with Tables associated with it. Publishing will prepare the storage for the data that will be sent during the loading.

Publishing can be triggered by clicking on “Publish” button on Dataset Properties page.

---

## STEP 4: LOADING

---

Loading of a Dataset will populate the Power BI Dataset with latest data available for it.

Loading can be triggered by clicking on “Load” button on Dataset Properties page.

---

## STEP 5: SCHEDULING DATA REFRESH

---

Finally user may want their reports and dashboards in Power BI Service to be updated automatically with recent data.

The following conditions should be met in order to make sure that scheduled updates are working:

1. Dataset Load Mode should not be set to match the name of one of the Schedules. This option is configured in Dataset Properties page in Magento Platform;
2. The Schedule to which the Dataset is assigned to has to be activated and configured in Magento Configuration as described in “Basic Configuration: Scheduled updates” section of this document;
3. Dataset needs to be published;
4. User should have his Power BI account linked with Power BI Service. State of the connection with Power BI Service can be verified on Datasets page (Menu *Reports* -> *Datasets*);
5. User should activate Automatic data actualization for his Magento account. This is also done on Datasets page (Menu *Reports* -> *Datasets*).
6. If the Dataset has Table Profiles assigned, those table profiles should be set to “Active” on Table Profile Properties page.

---

## DATASETS AND TABLE PROFILES

---

---

### MANAGING DATASETS

---

As mentioned in “Terminology: Power BI Integration Concepts”, a **Dataset** is a representation of a Power BI Dataset inside Magento Platform.

List of Datasets user can see inside Magento Platform by navigating to *Reports* -> *Datasets*:

PBI Datasets - Data Load to Power BI

Disable Auto Load    Disconnect from Power BI API    Add New Dataset

Connection with Power BI: **Enabled**  
Automatic data actualization for current user: **Enabled**

5 records found    20 per page    1 of 1

Dataset Name	Dataset Type	Is Temporary	Workspace	Load Mode	Status	Action
DEMO1_MAGE_REPORTS	Mixed	Permanent	Demo1	Manual	PUBLISHED	
DEMO1_SALES_CUBE	Sales Cube	Permanent	Demo1	Manual	PUBLISHED	
M2_SC	Sales Cube	Permanent	My Workspace	Manual	PUBLISHED	
NewDatasetSample	Mixed	Permanent	My Workspace	Manual	NOT PUBLISHED	
PBI_TEMP	Mixed	Temporary	My Workspace		PUBLISHED	

Copyright © 2018 Magento, Inc. All rights reserved.    Magento ver. 2.1.7

Picture: list of datasets inside Magento Platform

Information in the columns “**Dataset Name**”, “**Dataset Type**”, “**Is Temporary**”, “**Load Mode**” corresponds to dataset properties described in section “General Concepts: Loading Process: Step 1: Creation of a Dataset” of this documentation.

**Status** column in datasets list shows current status of the dataset. Status column may show the following values:

- **Not Published** – dataset is created, but was not published yet. It only exists inside Magento Platform.
- **Published** – dataset is published and is linked with remote dataset inside Power BI Service.
- **Inconsistent** – dataset is published and is linked with remote dataset, but remote dataset in Power BI Service has different structure (different tables, table names or columns)..
- **Disconnected** – there is no connection with Power BI Service or user did not connect to its Power BI account.
- **Missing** – dataset was published and has a link to remote dataset inside Power BI Service, but it does not exist in Power BI. This could be happen in dataset was deleted from Power

BI Service or if the user is connected to another Power BI Account, not to the one which was used during the dataset publishing.

- **Malformed** – some error occurred during retrieval of dataset status. An error can be found in exception logs.

To edit or perform any other actions on the dataset, user should click on the dataset's row and navigate to **Dataset Properties page**. For the simplicity some actions on single dataset can be performed also by selection appropriate option from the list of actions in the **Action** column.

Dataset Properties Page allows editing the dataset or performing some actions on top of the dataset.

The screenshot shows the 'Edit PBI Dataset' interface. The top navigation bar includes a search icon, a notification bell, and a user profile icon labeled 'user'. Below this is a toolbar with the following actions: 'Back', 'Delete Dataset', 'Disconnect dataset from PBI', 'Publish', 'Load', 'Delete loaded data', and a large orange 'Save Dataset' button. The main content area is titled 'Dataset Information' and contains the following fields:

- Dataset name \***: DEMO1\_MAGE\_REPORTS
- Workspace name**: Demo1
- Load Mode**: Manual
- Is Temporary**: No
- Dataset type**: Mixed
- Remote Dataset ID**: 45386f54-ae7f-4ba4-afae-c859cf137c49 (with a link to 'Open dataset in Power BI')

A vertical sidebar on the left side of the page contains icons and labels for various sections: DASHBOARD, SALES, PRODUCTS, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, SYSTEM, and FIND PARTNERS & EXTENSIONS.

Picture: dataset properties page

Listed options are described in section “General Concepts: Loading Process: Step 1: Creation of a Dataset” of this documentation.

User can trigger the following actions from the Dataset Properties page:

- **Delete Dataset** – delete dataset from Magento Platform. This action will not delete dataset from Power BI Service. Published dataset cannot be deleted.
- **Disconnect dataset from Power BI** – will remove the link to remote dataset and therefor will put the dataset into “Not published” state.
- **Publish** – performing the publishing of the dataset to Power BI Service.
- **Loading** – loading of the data to published dataset.
- **Delete loaded data** – this will clean up the data in the remote dataset inside Power BI Service.

Depending on the dataset type, Dataset Properties Page may include additional properties specific for particular dataset type.

Latest version of the extension will also include information about the data loading state. This will help to monitor the loading process.

Dataset Loading Info	
Schema Publishing State	Published
Remote Dataset ID	b44b478e-61f8-4a0f-94da-849345b0563e <a href="#">Open dataset in Power BI</a>
Data Loading State	Scheduled
Latest data actualization duration	40 seconds

*Picture: Data Loading Information*

---

## MANAGING TABLE PROFILES

---

Dataset may be self-containing with all tables and relationships or may include **Table Profiles**.

List of Table Profiles can be opened in Magento Platform by navigating to *Reports -> Table Profiles*.

<input type="checkbox"/>	Name	Dataset	Table	Type	Created At	Load Mode	Status	Action
<input type="checkbox"/>	Abandoned Carts	DEMO1_MAGE_REPORTS	AbandonedCarts	report	11/4/18, 3:14 PM	Manual	LOADED	<input type="button" value="Action"/>
<input type="checkbox"/>	Bestsellers	DEMO1_MAGE_REPORTS	Bestsellers	report	11/4/18, 3:18 PM	Manual	LOADED	<input type="button" value="Action"/>
<input type="checkbox"/>	Coupons	DEMO1_MAGE_REPORTS	Coupons	report	11/4/18, 3:17 PM	Manual	LOADED	<input type="button" value="Action"/>
<input type="checkbox"/>	Coupons (1)	PBI_TEMP	Coupons_1	report	11/4/18, 3:17 PM	Manual	INACTIVE	<input type="button" value="Action"/>
<input type="checkbox"/>	Customer Order Count	PBI_TEMP	CustomerOrderCount	model	11/29/18, 10:14 PM	Manual	INACTIVE	<input type="button" value="Action"/>
<input type="checkbox"/>	Customer Order Count (1)	PBI_TEMP	CustomerOrderCount_1	model	11/29/18, 10:18 PM	Manual	TEMPORARY	<input type="button" value="Action"/>
<input type="checkbox"/>	Customer Order	PBI_TEMP	CustomerOrderCount_2	model	11/29/18, 10:18	Manual	TEMPORARY	<input type="button" value="Action"/>

Picture: Table Profiles list

Each row in the table lists

- **Name** - Profile Name.
- **Dataset** - dataset to which the profile is assigned.
- **Table** - Power BI Table Name.
- **Type** - type of the Table Profile. This property is used internally. It depends on the way the profile was created.
- **Created At** - date of the Table Profile creation.
- **Load Mode** - configured Load Mode for this Table Profile.
- **Status** - status of the Table Profile.
- **Action** - list of quick actions that could be performed on the Table Profile.

**Status** column in Table Profiles list may get the following values:

- **Inactive** - Table Profile is set to inactive.
- **Inconsistent** - Table Profile is assigned to a dataset which is published, but Power BI Table is not in the dataset or has different columns.
- **Temporary** - Table Profile was published with its dataset, but the dataset is Temporary.
- **Published** - Table Profile was published with permanent dataset.
- **Not Published** - Table Profile is configured as active, but was not published yet (linked dataset is also not published).

- **Disconnected** - there is no connection with Power BI Service or user did not connect to its Power BI account.
- **Invalid Profile** – Table Profile is not supported by currently installed version of the Power BI Integration Extension or is malformed.

By clicking on the row in Table Profiles list, user can open Table Profile Properties page.

← Back   Delete Profile   Reset   Load To Power BI   Open Source Page   Duplicate Profile   Save Profile

**Profile Information**

Profile name \* Abandoned Carts

Table name \* AbandonedCarts

Dataset name DEMO1\_MAGE\_REPORTS

Is Active

Load Mode Manual

**Report Parameters**

Date From \* 11/6/2011  
Date in YYYY-MM-DD format or strings like "-1 day", "-1 week" or other strings, supported by PHP function strtotime().

Date To \* 11/6/2018  
Date in YYYY-MM-DD format or strings like "-1 day", "-1 week" or other strings, supported by PHP function strtotime().

Date Range Preview 2011-11-06 - 2018-11-06  
Demonstrates resulted date period generated based on Date From and Date To values.

Picture: Table Profile properties

User can change the following properties of Table Profile:

- **Profile Name** – name of the Table Profile for internal reference (used only inside Magento Platform)
- **Table Name** – name for the Power BI Table. Once the dataset with this Table Profile will be published, it will include table with the given name associated with its Table Profile.
- **Dataset Name** – the dataset where this Table Profile is assigned to.
- **Is Active** – determines if the table profile is active and can be published and loaded to Power BI or it is deactivated.
- **Load Mode** – property similar to dataset Load Mode. Can take value “Manual” for loading of this profile only if the user triggers loading manually or in cases when the whole dataset is loaded. Load Mode may also get a value of one of the available Schedules: Live or Regular. In this case the Table Profile will be loaded by the assigned schedule even if the associated dataset is loaded more rarely. In most of the cases this property may be left as “Manual” and



loading of the profile can be controlled through the dataset. It makes sense to set this property only in cases when different table of the same dataset should be loaded by different schedules.

Also, Table Profile may list additional properties depending of its type.

---

## MULTIPLE USERS

---

---

### ROLES

---

It is easy to understand how to use the solution if seen from the perspective of one user. But once more distinct roles are involved it becomes more complex. There are multiple options on how multiple users can organize their work with Power BI Service and Magento Platform and each option might also have different Power BI subscription costs.

To be able to review all use possible options, let's identify the roles that each person may take:

- **Magento Administrator** – Magento administrator may restrict access to certain functionality and allow only dedicated users to use Power BI Integration or parts of its functionality. In case if Power BI Embedded is used, it might also be possible to restrict access to certain embedded content. Administrators are allowed to view any content inside Magneto Platform
- **Azure AD Administrator** – role which is able to add new users and manage their permissions and licenses inside Azure AD.
- **Power BI Integration Datasets administrator** – role for the users who will create, configure and publish the datasets that will contain data from Magento Platform to Power BI Service.
- **Power BI Service Report Creator** – role for the users who will be creating reports and dashboards inside Power BI Service. They may also create reports using the Power BI Desktop and upload their reports to Power BI Service.
- **Power BI Service Content Consumers** – role for the users who will be accessing Power BI content using Power BI Service.
- **Power BI App Content Consumers** – users who may get access to created Reports that were shared with them either from Power BI Service or, if Power BI Embedded extension is used, from inside the Magento Platform.
- **Power BI Embedded Content Consumers** – such users cannot get access to shared reports inside Power BI Service, but it is possible to share content with such users within Magneto Platform using Power BI Embedded extension.

The general use case is the delivery of the content from the report creators to content consumers. Additionally it might be desired to be able to restrict access to certain content. Report creators may wish to deliver particular reports to particular content consumers.

To split the use cases we review first the delivery and creation of the content in Power BI Service. Other use cases relate to different ways in which the content can be shared with other users.

---

### SIMPLEST USE CASE: SINGLE USER SCENARIO

---

The simplest use case is to have one user who will play all roles and do everything: from administration and reports creation to consuming the content with Power BI Service and Power BI Mobile App.

In this case it might even be enough to have Power BI Free subscription.

Upgrading to Power BI Pro subscription would make sense to extend the limitations of FREE account and deliver more data, more frequently.

---

### SIMPLEST MULTIUSER USE CASE: MULTIPLE INDEPENDENT USERS

---

Independent users will create and publish their own datasets and create reports based on those datasets. In this scenario they do not share the content between each other.

One of the advantages of this approach is that each user may stay with Power BI Free subscription and have full flexibility in configuring and publishing the datasets.

The disadvantage is that there might be duplicated data sent from Magento Platform to Power BI Service for multiple users. The more independent users there will be the more performance it will consume from the Magento Platform. At the moment the Power BI Integration Extension can handle only one loading job at a time. And in case if multiple users will be frequently updating their datasets, the process of loading of the data for one user may wait for another processes to finish. Therefore it is not a recommended approach.

The better approach would be to share dataset between the users, but this may require purchasing Power BI Pro subscription for every report creator.

---

## USE CASES OF CONTENT DELIVERY

---

### MULTIPLE INDEPENDENT REPORT CREATORS

As in the example with multiple independent users, independent report creators will create and publish their own datasets and create reports based on those datasets.

The advantage would be that each report creator will have the possibility to create datasets that better meets their needs. This might be useful in cases where the dataset is highly configurable and each report creator may prefer its own configuration for the dataset.

As in the independent users scenario, same data might be sent from Magento Platform to Power BI for multiple users. The more independent report creators there will be the more performance it will consume from the Magento Platform. This will also slow down data delivery for independent report creators.

#### MULTIPLE REPORT CREATORS SHARING THE SAME DATASET

This approach is more optimal since data might be delivered from Magneto Platform to Power BI Service only once and then be shared with multiple reports creators.

Sharing the access to a dataset would require each report creator to have a Power BI Pro subscription.

One of the users should take over the role of Power BI Integration Datasets Administrator. This user will create dataset and will configure the scheduled update of the data. The dataset can then be published to a Power BI Workspace shared with report creators.

#### SPECIAL USE CASE: SHARE REPORT TEMPLATES

One special option on how the content of Power BI might be delivered to other users is though Report Templates.

In this case the user may receive a report template that he can upload to Power BI Service. Even though, he was not developing the report himself, but in fact he plays a role of report creator.

This becomes possible if the purchased edition of Power BI Integration Extension supports this functionality.

---

#### USE CASES OF CONTENT CONSUMING

---

The content from Power BI Service might be delivered to end users in the following ways:

1. As their own content from their workspace.
2. As the content shared with them through workspace.
3. As the content embedded to Magento Platform.
4. As the public content shared in the web.

Let's review each option.

#### ASSESSING OWN CONTENT IN POWER BI SERVICE

To access the Power BI Service user needs to have at least Power BI Free subscription (which means that user should be registered and have Power BI Service Account).

Of course in order to have own content, user will need to be create it. So, the user should be a Report Creator.

Own content can be accessed through the Power BI Service, on Power BI App or through embedding into Magento Platform.

If the Power BI Integration Extension supports embedded content in Magento Platform, User can navigate to *Reports -> Power BI -> Reports* (for reports) or *Reports -> Power BI -> Dashboards* (for dashboards) to access the content from Power BI Service.

#### ACCESSING CONTENT IN SHARED WORKSPACES

Report Creators with Power BI Pro subscription can create workspaces and place their content there. Such workspaces can be shared with other users.

In order to have access to shared content, users also need to have Power BI Pro subscription.

Content shared through workspaces can be accessed through the Power BI Service, on Power BI App or through embedding into Magento Platform. If the Power BI Integration Extension supports embedded content in Magento Platform, User can navigate to *Reports -> Power BI -> Reports* (for reports) or *Reports -> Power BI -> Dashboards* (for dashboards) to access the content from Power BI Service shared with him through workspaces.

#### ACCESS CONTENT EMBEDDED INTO MAGENTO PLATFORM

Report Creators with Power BI Pro subscription can create workspaces and place their content there.

There are two ways, how other users can access embedded content:

1. Access content from their Power BI account linked with Magento Platform.
2. Access embedded content without the need to have Power BI Service account.

First option is just an extension to the use cases listed before: user either has his own content in Power BI Service or the content is shared with him through Power BI Workspaces.

Second option would require purchasing Power BI Embedded capacity from Microsoft and assigning this capacity to the Workspace where the shared content is placed.

More detailed about embedding with Power BI Integration Extensions are described in “Embedding Power BI Content” section of this documentation.

---

#### MULTIPLE USERS WITH DIFFERENT PERMISSIONS

---

Special use cases may popup when different users are intended to have different permissions. Please, contact Extension provider to discuss the solution that will fit the concrete requirements.



## TABLE PROFILES BASED ON MAGENTO REPORTS

### CREATION OF TABLE PROFILES BASED ON MAGENTO REPORTS

To create a Load Profile, user navigates to a Magento Report. If the report supports export there is an option of exporting to Power BI available on the page. Triggering the export will create a Load Profile that will keep association with selected filter values. Please note that Export button should be used after the selected filter values are applied to the report. First user enters the filter values, then click on “Show Report” and only after that he can select “Power BI” export option and click on “Export”.

Each report if it has a date filter should have the date range specified in that filter before it could be used as a source for Table Profile creation.

Orders Report Show Report

Date Used  The Order Updated report is created in real time and does not require a refresh.

Period

From

To

Order Status  Applies to Any of the Specified Order Statuses except canceled orders

Empty Rows

Show Actual Values

Export to:

1 records found

Interval	Orders	Sales Items	Sales Total	Invoiced	Refunded	Sales Tax	Sales Shipping	Sales Discount	Canceled
5/2017	2	2	\$76.03	\$76.03	\$7.64	\$5.03	\$10.00	\$0.00	\$0.00
Total	2	2	\$76.03	\$76.03	\$7.64	\$5.03	\$10.00	\$0.00	\$0.00

Picture: Magento Orders report with Export to Power BI capabilities

Click on “Export” button will result in Table Profile creation and a message informing user about successful profile creation will be shown.

### CONFIGURING REPORT BASED TABLE PROFILE

Table Profiles based on Magento Reports may have extra properties on their details page. If the Magento Report supports specifying a date range, there will be **Date From** and **Date To** fields visible on Table Profile Properties page.

← Back Delete Profile Reset Load To Power BI Open Source Page Duplicate Profile Save Profile

### Profile Information

Profile name \* Abandoned Carts

Table name \* AbandonedCarts

Dataset name DEMO1\_MAGE\_REPORTS ▼

Is Active

Load Mode Manual ▼

### Report Parameters

Date From \* 11/6/2011  
Date in YYYY-MM-DD format or strings like "-1 day", "-1 week" or other strings, supported by PHP function strtotime().

Date To \* 11/6/2018  
Date in YYYY-MM-DD format or strings like "-1 day", "-1 week" or other strings, supported by PHP function strtotime().

Date Range Preview 2011-11-06 - 2018-11-06  
Demonstrates resulted date period generated based on Date From and Date To values.

Picture: Properties of report based Table Profile

**Date From** and **Date To** options allows changing the start of the period that data will cover.

User can also enter dynamic values in those fields such as “-1 day”, “-3 week” or other values supported by *strtotime* function in PHP (<http://php.net/manual/de/function strtotime.php>). This allows having a table that will always hold data for particular period relative to current moment, such as for example the data for latest 3 months.

**Date Range Preview** field will show how system interprets the entered period.

## SUPPORTED REPORTS

This section describes what Magento Reporting data can be sent to Power BI Service with current version of the Power BI Integration extension.

Table Profiles are created based on tables or grids in Magneto backend. Most of them are taken from Magento Reporting pages. The Power BI Integration is supported for the grid if there is an option of exporting to Power BI available in the list of possible exports as it is shown on Image 2.

#### SUPPORTED REPORTS LIST FOR MAGENTO 1

The Magento Reports that can be exported to Power BI Service in **Magento 1** at the moment are the following:

1. Total Ordered Report, *Reports -> Sales -> Orders*
2. Order Taxes Report Grouped by Tax Rate, *Reports -> Sales -> Tax*
3. Total Invoiced vs. Paid Report, *Reports -> Sales -> Invoiced*
4. Total Shipped Report, *Reports -> Sales -> Shipping*
5. Total Refunded Report, *Reports -> Sales -> Refund*
6. Coupons Usage Report, *Reports -> Sales -> Coupons*
7. Products in carts, *Reports -> Shopping Cart -> Products in carts*
8. Abandoned carts, *Reports -> Shopping Cart -> Abandoned carts*
9. Products Bestsellers Report, *Reports -> Products -> Bestsellers*
10. Products Ordered, *Reports -> Products -> Products Ordered*
11. Most Viewed (Products) , *Reports -> Products -> Most Viewed*
12. Low stock (Products) , *Reports -> Products -> Low stock*
13. Downloads, *Reports -> Products -> Downloads*
14. New Accounts, *Reports -> Customers -> New Accounts*
15. Customers by Orders Total, *Reports -> Customers -> Customers by orders total*
16. Customers by number of orders, *Reports -> Customers -> Customers by number of orders*
17. Customers Reviews, *Reports -> Reviews -> Customers Reviews*
18. Products Reviews, *Reports -> Reviews -> Products Reviews*
19. Search Terms, *Reports -> Search Terms*

Additionally creation of Table Profile is possible based on non-reporting lists (only one list is connected in current version):

Orders List, *Sales -> Orders*

#### SUPPORTED REPORTS LIST FOR MAGENTO 2

The Magento Reports that can be exported to Power BI Service in **Magento 2** at the moment are the following:

1. Products in Cart
2. Products Views
3. Downloads
4. Shipping
5. Refunds
6. Orders
7. Invoiced



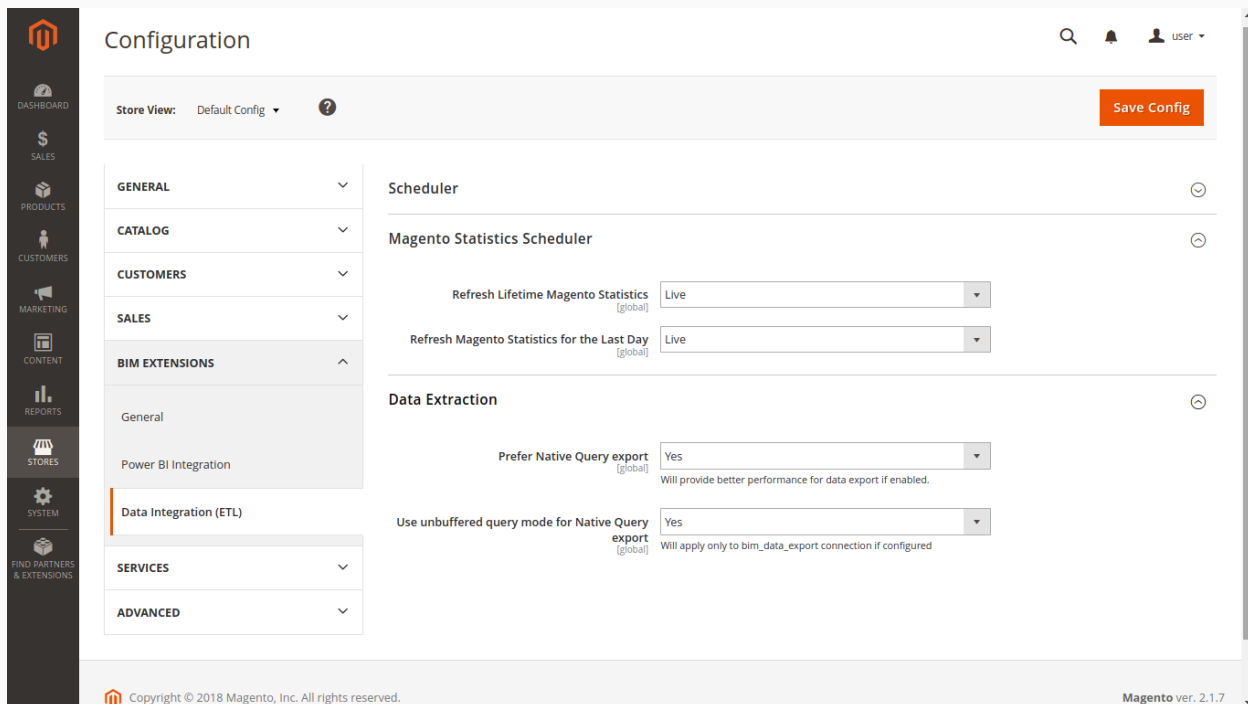
8. Coupons
9. Tax
10. Bestsellers
11. Abandoned Carts
12. Customer Totals
13. New Accounts

## MAGENTO STATISTICS REFRESHING

Table Profiles based on Magento Reports will use those reports as a data source. Since the date in Magento Reports needs to be refreshed in order to get actual values, the same relates to report based Table Profiles.

To facilitate automated data actualization of report based Table Profiles, the Power BI Integration Extensions come with a functionality to automatically refresh Magento Statistics.

To activate statistics refreshing, user can navigate to Magento Configuration page: *Stores -> Configuration* in Magento 2 or *System -> Configuration* in Magento 1. Section *BIM Extensions -> Data Integration (ETL) -> Magento Statistics Scheduler*.



The screenshot displays the Magento Configuration interface. On the left is a vertical sidebar with navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area is titled 'Configuration' and shows a 'Store View' dropdown set to 'Default Config'. A 'Save Config' button is in the top right. The 'BIM EXTENSIONS' section is expanded to show 'Data Integration (ETL)'. Under this, the 'Scheduler' section is expanded, containing two settings: 'Refresh Lifetime Magento Statistics' (set to 'Live') and 'Refresh Magento Statistics for the Last Day' (set to 'Live'). Below that, the 'Data Extraction' section is expanded, containing two settings: 'Prefer Native Query export' (set to 'Yes') and 'Use unbuffered query mode for Native Query export' (set to 'Yes'). The footer shows 'Copyright © 2018 Magento, Inc. All rights reserved.' and 'Magento ver. 2.1.7'.

Picture: Magento statistics refresh schedule config

The configuration has two options in Magento Statistics Scheduler, each taking as value of one of the available schedules. Typical values are Life or Regular.

**Refresh Lifetime Magento Statistics** option allows configuring schedule for refreshing complete Magento statistics. User should be careful with this option since refreshing Magento statistics might be computationally intensive task. It is not recommended to refresh this statistics too often, especially if the store has lot of data. This option can be enabled if user wants to refresh data less frequently then just once per day.

**Refresh Magento Statistics for the Last Day** option allows to configure schedule for refreshing only statistics for the last day. This option is more preferable to be used in connection with automated data updates when data is refreshed at least once per day.

---

## USAGE OF FILTERS IN MAGENTO REPORTS

---

Most of the Magento reports come with filters that allow filtering data by date range, stores, websites, etc. Users can select the desired filters and create Table Profiles with different combinations of filters. This provides some flexibility in making reports.

Some examples on how the data might be combined based on Magento report based Table Profiles:

- Several tables with Magento sales data, each containing data from different stores.
- Tables Profiles with data covering different periods: one table for data for the last year, second table with data for the year before.

# SALES CUBE USAGE

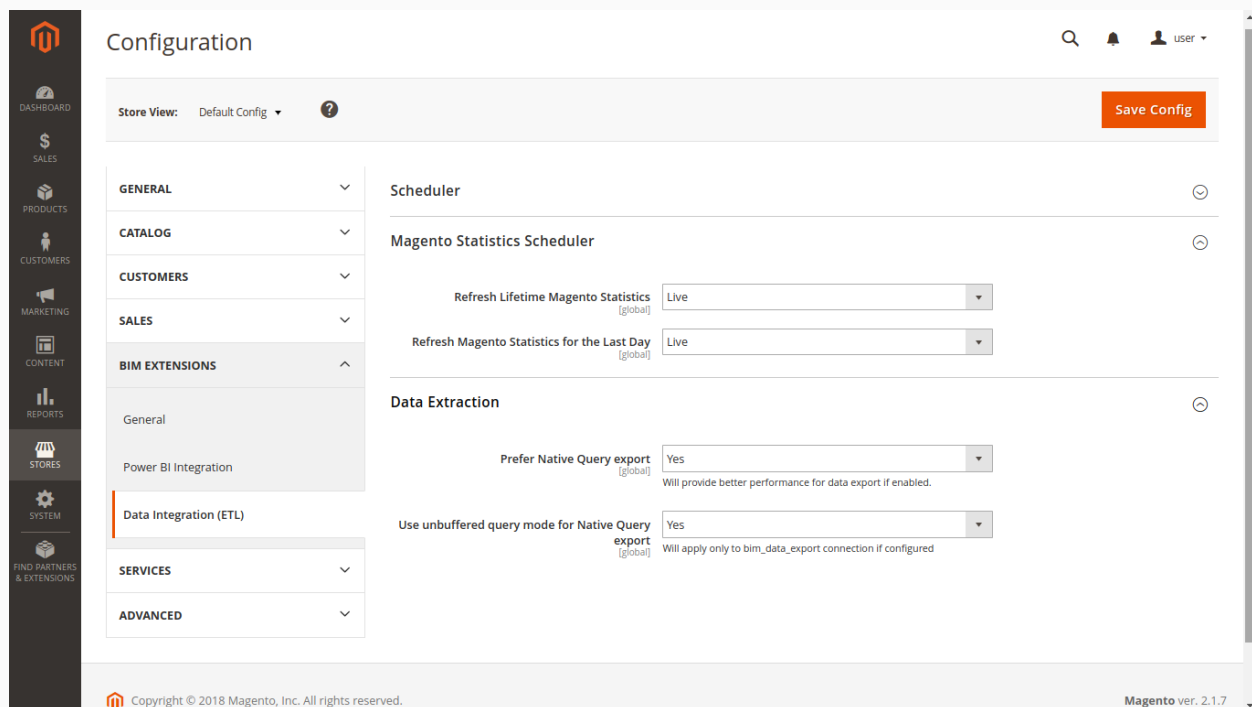
## WHAT IS SALES CUBE DATASET?

Power BI Integration Extension may be supplied with the support of Sales Cube dataset type (**Sales Cube Dataset**). Datasets of this type are self-containing and does not require user to create Table Profiles. The Sales Cube Dataset delivers wide range of data related to sales process. All tables and their data is linked with the dataset type and all what user needs to do to deliver extensive set of data related to sales is to create dataset, publish it and let the data to be loaded.

More detailed information about the data included in the Sales Cube could be found in Sales Cube Data Dictionary document.

## CONFIGURING SALES CUBE ETL SETTINGS

There are additional options related to data extraction logic in Magento Configuration (*Stores -> Configuration* in Magneto 2 or *System -> Configuration* in Magneto 1), section *BIM Extensions -> Data Integration (ETL) -> Data Extraction*.



The screenshot displays the Magento Configuration interface. On the left is a vertical sidebar with navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area is titled 'Configuration' and shows a 'Store View' dropdown set to 'Default Config'. A 'Save Config' button is in the top right. The left sidebar is expanded to 'BIM EXTENSIONS', with 'Data Integration (ETL)' selected. Under 'Data Integration (ETL)', the 'Data Extraction' section is active. It contains two sub-sections: 'Scheduler' and 'Data Extraction'. The 'Scheduler' section includes 'Magento Statistics Scheduler' with two dropdowns: 'Refresh Lifetime Magento Statistics' (set to 'Live') and 'Refresh Magento Statistics for the Last Day' (set to 'Live'). The 'Data Extraction' section includes 'Prefer Native Query export' (set to 'Yes') and 'Use unbuffered query mode for Native Query export' (set to 'Yes'). Both dropdowns have explanatory text below them. The footer shows 'Copyright © 2018 Magento, Inc. All rights reserved.' and 'Magento ver. 2.1.7'.

Picture: Data Extraction settings

**Prefer Native Query export** option is controlling the way the data will be extracted from Magento Platform. When this option is enabled, the logic that extracts data from Magento will try to use SQL queries instead of relying on Magento standard data reading process which might not be well optimized for intensive data extraction process. This option is enabled by default. Disabling this option is not recommended. Only disable this option in case if Native Query cannot be used for some reason.

**Use unbuffered query mode for Native Query export** option will change the connection properties used to extract data from Magento Database. Unbuffered query mode is more preferable for data extraction logic. When this option is enabled, memory consumption may increase on the database server. Disabling this option will increase memory consumption of Magento Platform (PHP). This option is enabled by default. Disabling this option is not recommended. Only disable this option in case if unbuffered query cannot be used for some reason.

Neither of options should have impact on core functionality of Magento Platform.

---

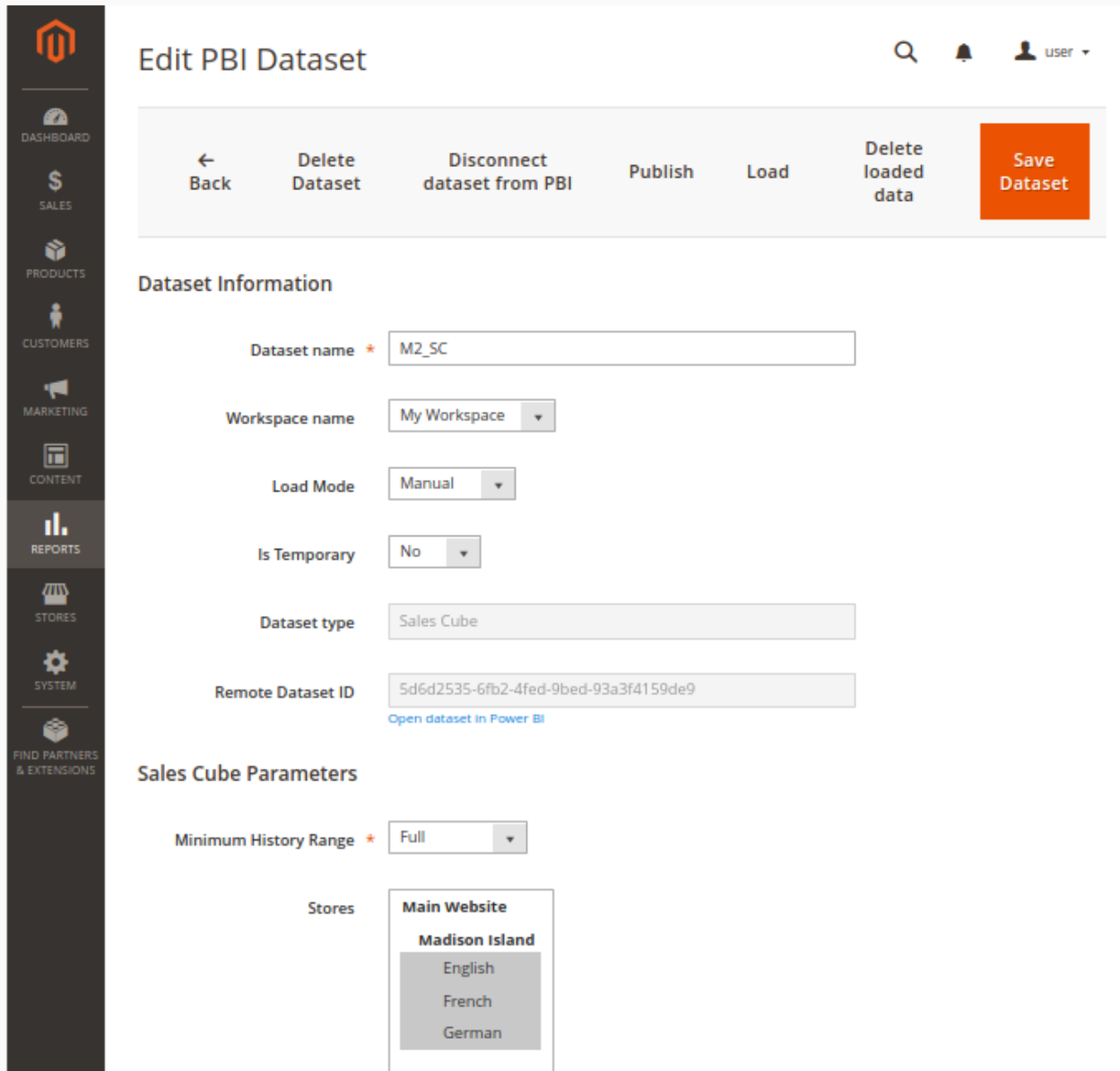
## CONFIGURING SALES CUBE DATASET

---

Additionally to standard Dataset properties, Datasets of Sales Cube type will have some additional settings on Dataset Properties page. Those settings allow decreasing amount of data in the dataset. This might be desired in case if complete data will go over limitations of Power BI Service.

**Minimum History Range** property allows limiting the period of history that will be included into the dataset.

**Stores** option allows selecting list of Magento stores that the data of the dataset will cover.



**Edit PBI Dataset**

← Back   Delete Dataset   Disconnect dataset from PBI   Publish   Load   Delete loaded data   **Save Dataset**

**Dataset Information**

Dataset name \* M2\_SC

Workspace name My Workspace

Load Mode Manual

Is Temporary No

Dataset type Sales Cube

Remote Dataset ID 5d6d2535-6fb2-4fed-9bed-93a3f4159de9  
[Open dataset in Power BI](#)

**Sales Cube Parameters**

Minimum History Range \* Full

Stores

- Main Website
- Madison Island
- English
- French
- German

Picture: Sales Cube dataset properties page

## SALES CUBE DATA LOADING

As mentioned before it is easy to use Sales Cube Dataset. But in case if dataset has lot of data, loading process may take quite some time. The dataset includes multiple tables, some of which may have several millions of rows. With Power BI Pro subscription users can upload data on the rate of

1 million rows per hour as maximum. And this limit cannot be increased (maybe only with Premium capacity based subscription).

Initial load of single Sales Cube Dataset may take more than 24 hours. For this reason it is not recommended to trigger initial loading of Sales Cube dataset using Load button in the Magento Platform, but let it be loaded by schedule.

Once the initial load will finish, all consequent loads will load only newly added data and this will be done faster.

The following actions are recommended to perform initial loading of the data:

1. Create Dataset with Sales Cube type.
2. Publish Dataset.
3. Configure dataset with Load Type set to Live schedule.
4. Enable Live schedule (in Magento Configuration).
5. Enable Auto Data Actualization for current user (on Datasets List page).
6. After waiting for the next scheduled data load, data should start loading process. This can be seen in ETL Jobs page (*Reports -> ETL Jobs*).

Additionally user might be able to estimate the amount of data in the dataset and time it will take to load the full dataset using Dev Tools. More detailed it is described in “Advanced Features: Usage of Dev Tools: Sales Cube Dev Tools” section of this documentation.

---

## REPORT TEMPLATES FOR SALES CUBE DATASET

---

Sales Cube functionality comes with set of reports that could be uploaded to Power BI Service and assigned to the Sales Cube Dataset.

At the moment the following report templates are delivered with Sales Cube:

1. **SC Embedded Customer Statistics** - Report for embedding as customer statistics page.
2. **SC Embedded Order Statistics** - Report for embedding as order statistics page.
3. **SC Embedded Product Statistics** - Report for embedding as product statistics page.
4. **SC Mage Order Taxes Report** - Replication of Magento Order Taxes Report.
5. **SC Mage Total Invoiced vs Paid Report** - Replication of Magento Total Invoiced vs Paid Report.
6. **SC Mage Total Ordered Report** - Replication of Magento Total Ordered Report.

Read “Embedding Power BI Content” for the information on how to upload the reports to Power BI Service.

Additional reports can easily be created inside Power BI Service.

---

## SALES CUBE DATA CLEAN-UP SCHEDULE

---

Every created and published Sales Cube dataset will be automatically processed by Clean Up scheduler. The schedule for the Clean Up could be configured on the Magento Configuration page, section BIM Extensions -> Data Integration (ETL) -> Scheduler.

The clean-up fill trigger full refresh of the data inside the Sales Cube dataset. This is required to actualize data with recent changes of product/category names and other attributes that cannot be easily updated inside Power BI after being published.

Set **Enable clean-up** option to “Yes” to enabling execution of the clean-up jobs.

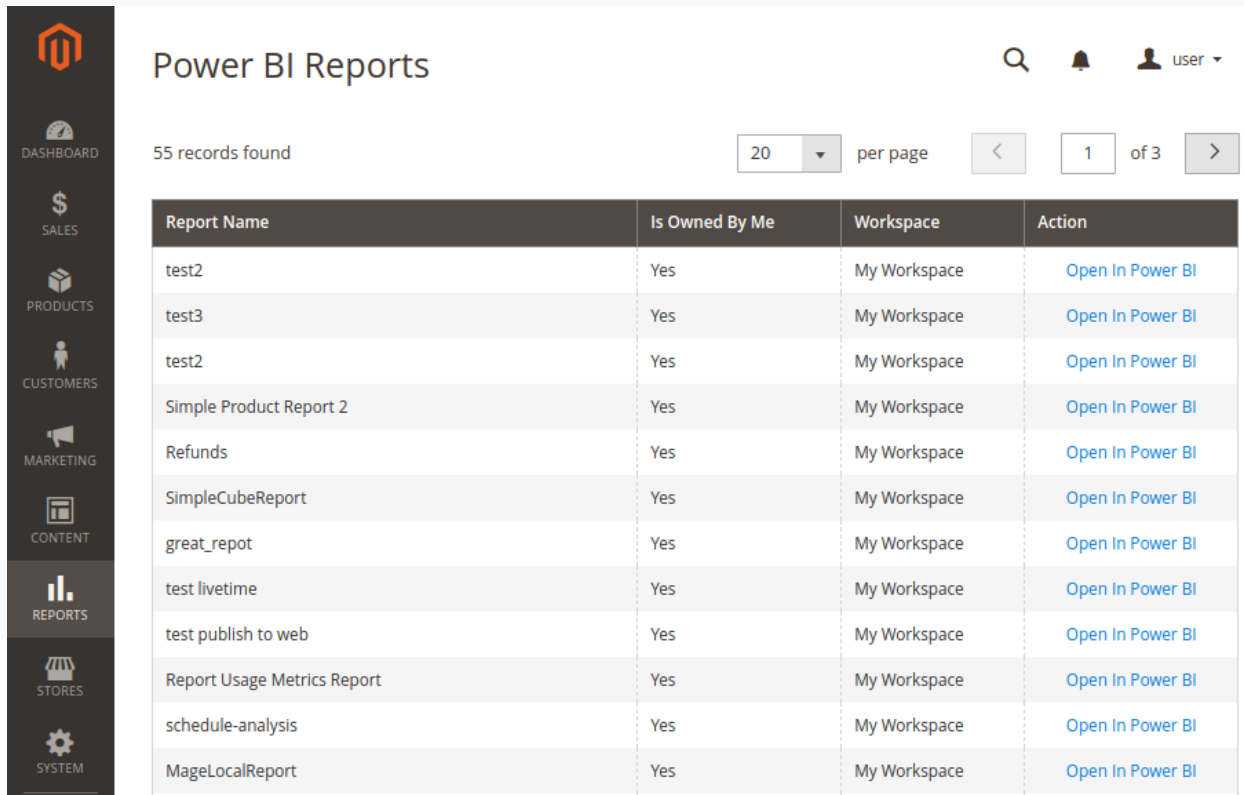
Configure **Clean-up Frequency** with desired frequency for the clean-up jobs: weekly, monthly or once in two months.

# EMBEDDING POWER BI CONTENT

## ACCESS POWER BI CONTENT WITHIN MAGENTO PLATFORM

Users that have their Power BI accounts can access their content directly from Magento Platform.

Opening page *Reports -> Power BI -> Reports* will show list of reports user have access to.



The screenshot displays the 'Power BI Reports' interface. On the left is a vertical navigation menu with icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, and System. The main content area shows the title 'Power BI Reports', a search icon, a notification bell, and a user profile dropdown. Below this, it indicates '55 records found' and a pagination control set to '20 per page', showing '1 of 3' pages. A table lists the reports with the following columns: Report Name, Is Owned By Me, Workspace, and Action.

Report Name	Is Owned By Me	Workspace	Action
test2	Yes	My Workspace	<a href="#">Open In Power BI</a>
test3	Yes	My Workspace	<a href="#">Open In Power BI</a>
test2	Yes	My Workspace	<a href="#">Open In Power BI</a>
Simple Product Report 2	Yes	My Workspace	<a href="#">Open In Power BI</a>
Refunds	Yes	My Workspace	<a href="#">Open In Power BI</a>
SimpleCubeReport	Yes	My Workspace	<a href="#">Open In Power BI</a>
great_repot	Yes	My Workspace	<a href="#">Open In Power BI</a>
test livetime	Yes	My Workspace	<a href="#">Open In Power BI</a>
test publish to web	Yes	My Workspace	<a href="#">Open In Power BI</a>
Report Usage Metrics Report	Yes	My Workspace	<a href="#">Open In Power BI</a>
schedule-analysis	Yes	My Workspace	<a href="#">Open In Power BI</a>
MageLocalReport	Yes	My Workspace	<a href="#">Open In Power BI</a>

Picture: Power BI Reports list inside Magento Platform

Clicking on the row in the list of reports will open report within Magento.



← Back    Open in Power BI

<b>\$162,557.72</b> Grand Total	<b>\$71,738.81</b> Total Invoiced	<b>\$56,334.92</b> Net Sales
------------------------------------	--------------------------------------	---------------------------------

Ordered	Invoiced	Refunded	Sales
\$158,409.11 Total Net Amount	\$69,796.67 Net Invoiced	\$13,461.75 Net Refunded	\$56,334.92 Net Sales
\$6,575.55 Total Shipping	\$3,161.39 Shipping Invoiced	\$1,370.18 Shipping Refunded	\$1,791.21 Sales Shipping
\$5,487.54 Total Tax	\$2,590.54 Tax Invoiced	\$433.39 Tax Refunded	\$2,157.15 Sales Tax
(\$7,714.48) Total Discount	(\$3,809.79) Discount Invoiced	(\$66.05) Discount Refunded	(\$3,743.74) Sales Discount
(\$200.00) Total Adjustments	\$0.00 Invoiced Adjustm...	\$63.00 Recorded Adjust...	\$0.00 Sales Adjustments
\$162,557.72 Grand Total	\$71,738.81 Total Invoiced	\$15,262.27 Total Refunded	\$56,476.54 Sales Total

Invoice Number	Invoice Date	Total Invoiced	Net Sales
100000004	4/4/2013	\$170.41	\$140.00
100000005	4/4/2013	\$681.84	\$660.00
100000006	4/4/2013	\$1,143.98	\$960.00
<b>Total</b>		<b>\$71,738.81</b>	<b>\$56,334.92</b>

Payment Id	Amount Authorized	Amount Paid Online	Amount Paid
41	0.00	0.00	0.00
42	0.00	0.00	0.00
43	0.00	0.00	0.00
<b>Total</b>	<b>0.00</b>	<b>0.00</b>	<b>71,738.81</b>

Creditmemo #	Credit Memo Date	Net Refunded	Total Refunded
100000005	05/29/2013	\$560.00	\$315.09
100000006	05/29/2013	\$110.00	\$134.23
100000007	06/18/2013	\$560.00	\$302.57

Parent Product Name	Average Price	Quantity Ordered	Ordered Row Total	Total Row Discount	Consumer Tax	Net Refunded	Net Sales
Order Items							

Picture: Opening Power BI report within Magento Platform

Opening page Reports -> Power BI -> Dashboards will show list of dashboards.

## Power BI Dashboards

3 records found    20 per page    1 of 1

Dashboard Name	Workspace
test	My Workspace
Today's Refunds	My Workspace
EC1 Test	My Workspace

Picture: list of Power BI dashboards inside Magento Platform

Same as with reports, user can open dashboard by selecting the row in the list.

## ENABLE AND CONFIGURE EMBEDDED CONTENT

### PAGES WITH EMBEDDED CONTENT

Embedded analytics can significantly improve the usability of statistical information by bringing the statistics to the place where it is needed.

Power BI Integration Extension may come with functionality that supports embedding of Power BI content to some of the pages inside Magento Platform. At the moment this includes the following pages (all in Magento Admin area):

- **Question & Answer (QnA)** section on Overview page.
- **Product statistics** tab on Product properties page
- **Customer statistics** tab on Customer properties page
- **Order statistics** tab on Order details page

Each embedded report will be described below.

In “AAD Token: User owns data” mode that will be described later in “Embedded Modes” section of this documentation, the embedded content will not be shown in case if the user did not connect Magento Platform to its Power BI Service account.

### QUESTION & ANSWER SECTION

Question & Answer section is based on natural language processing features provided by Power BI. This will allow user to enter a question to a text field and get answer to that question provided by Power BI. The Question & Answer embedded functionality at the moment can only take data from one dataset, though Microsoft have plans to extend this to more datasets.

Customer	Items	Total
Drew France	4	-€23.69
Drew France	2	-€16.83
Drew France	2	€1,909.80
Drew France	2	€1,909.80
Drew France	1	€3,577.75

Revenue	Tax	Shipping	Quantity
€0.00	€0.00	€0.00	0

Picture: embedded Question & Answer section inside Magento Platform

This is quite powerful features with very user friendly interface.

The quality of the answers may depend a lot on the underlying data modeling in the dataset. To get most benefit from Question & Answer section it is recommended to use Sales Cube Dataset.

### ORDER STATISTICS TAB

Embedded report might be added to order details page in Magento Platform. This will allow opening separate report for each order.

The embedding functionality expects the embedded report to have *Order* table with *Order Id* field and will filter data by this field.

This functionality is compatible with Sales Cube Dataset. In particular, report template “SC Embedded Order Statistics” that comes with Sales Cube functionality might be used to be embedded to order statistics tab.

The screenshot shows a Magento order page with an embedded 'ORDER VIEW' sidebar and an 'ORDER STATISTICS' report. The report displays the following data:

Category	Value
Grand Total	\$1,566.11
Total Invoiced	(Blank)
Net Sales	(Blank)

**Ordered**

Total Net Amount	\$1,475.00
Total Shipping	\$37.86
Total Tax	\$103.25
Total Discount	(\$50.00)
Total Adjustments	\$0.00
Grand Total	\$1,566.11

**Invoiced**

Net Invoiced	(Blank)
Shipping Invoiced	(Blank)
Tax Invoiced	(Blank)
Discount Invoiced	(Blank)
Invoiced Adjustments	(Blank)
Total Invoiced	(Blank)

**Refunded**

Net Refunded	(Blank)
Shipping Refunded	(Blank)
Tax Refunded	(Blank)
Discount Refunded	(Blank)
Recorded Adjustments	(Blank)
Total Refunded	(Blank)

**Sales**

Net Sales	(Blank)
Sales Shipping	(Blank)
Sales Tax	(Blank)
Sales Discount	(Blank)
Sales Adjustments	(Blank)
Sales Total	(Blank)

**Order Items**

Parent Product Name	Average Price	Quantity Ordered	Ordered Row Total	Total Row Discount	Consumer Tax	Net Refunded	Net Sales
Draper Pant	\$287.00	5.00	\$1,475.00	(\$50.00)	\$103.25		
<b>Total</b>	<b>\$287.00</b>	<b>5.00</b>	<b>\$1,475.00</b>	<b>(\$50.00)</b>	<b>\$103.25</b>		

Picture: Embedded Order Statistics report on an order's page

### PRODUCT STATISTICS TAB

Embedded report might be added to product edit page in Magento Platform. This will allow opening separate report for each product.

The embedding functionality expects the embedded report to have *Products* table with *Product Id* field and will filter data by this field.

This functionality is compatible with Sales Cube Dataset. In particular, report template “SC Embedded Product Statistics” that comes with Sales Cube functionality might be used to be embedded to product statistics tab.

#### CUSTOMER STATISTICS TAB

Embedded report might be added to customer details page in Magento Platform. This will allow opening separate report for each customer.

The embedding functionality expects the embedded report to have *Orders* table with *Customer Id* field and will filter data by this field.

This functionality is compatible with Sales Cube Dataset. In particular, report template “SC Embedded Customer Statistics” that comes with Sales Cube functionality might be used to be embedded to customer statistics tab.

---

#### EMBEDDING MODES

---

There are two embedding modes (types of tokens) for embedding the content supported by Power BI Integration Extension:

- **AAD Token: User owns data** – in this mode user who will get access to the embedded content needs to have an account in Power BI Service.
- **Embed Token: App owns data** – in this mode user who will get access to the embedded content does not need to have an account in Power BI Service, but the functionality requires purchasing Power BI Embedded capacity.

The suitable mode may depend on the amount of users who should get access to the embedded content. If the amount of users is less than 80 users, then it might be preferable to first option, if more than 100 – second option might become more cost efficient.

In the first option content should either be created by the user or be shared with him. Sharing the content would require to have Power BI Pro subscription for all users who want to share or access shared content inside Power BI. Second option requires Power BI Embedded capacity.

Let's review all of the possibilities in details.

#### AAD TOKEN WITH POWER BI FREE ACCOUNTS

This is the option where there might be no need to pay any license fees for Power BI Service. But this is very inefficient and limited option. Power BI Free subscription is more intended to be used as a trial version of Power BI Service and is very limited.

First of all, the limitations relate to the capacity and slow data delivery:

- Max amount of rows in a table is limited by 10,000 rows (compared with 5,000,000 rows in Pro subscription)

- Max amount of rows sent to Power BI per hour per dataset is limited by 10,000 rows

Additionally, with Power BI Free subscription users cannot share content with each other. This means that each user should load his data to Power BI separately. More about this case can be read in “General concepts: Multiple users: Simplest multiuser use case: multiple independent users”.

#### AAD TOKEN WITH POWER BI PRO ACCOUNTS

With Power BI Pro accounts users can enjoy full flexibility of usage of Power BI Service. Limitations will be extended compared to free account to allow storing more data and sending it faster to Power BI Service.

Users can get access not only to embedded content, but also access reports and dashboards with Power BI Service or Power BI Mobile App, create and store their reports, share their reports, configure notifications and exchange their ideas with their colleagues through Power BI Service.

But each Power BI Pro license needs to be purchased from Microsoft and the more users there will be, the more expensive it becomes.

#### EMBED TOKEN: APP OWNS DATA

This is a typical embedded use case and ideal for companies that have multiple users who will access embedded content without the need to have Power BI account.

The users will not be able to create and share reports. They will not get access to content in Power BI Service and will not access the reports through Power BI Mobile App.

This use case requires purchasing Power BI Embedded capacity from Microsoft.

Note: for development purpose Microsoft allows to use this kind of token with very limited capacity. This is useful for development and testing of this embedding mode.

---

#### CONFIGURING THE EMBEDDED CONTENT

---

Embedded content can be configured in Magento Configuration (*Stores -> Configuration* in Magento 2 or *System -> Configuration* in Magento 1), section *BIM Extensions -> Power BI Integration -> Power BI Embedded*.

The screenshot shows the 'Configuration' page for 'Power BI Embedded' in the Magento Platform. The left sidebar contains navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main configuration area is titled 'Power BI Embedded' and includes a 'Save Config' button in the top right corner. The configuration is organized into sections: 'General', 'Power BI Integration', and 'Data Integration (ETL)'. The 'Power BI Integration' section contains the following settings:

- Embedding Mode: User Owns Data vs App owns data** (global): AAD Token: User owns data (dropdown menu)
- Enable QnA Section on Overview Page** (global): Yes (dropdown menu)
- QnA Workspace ID** (global): 94050cf5-5c81-4be8-84c9-cd95e6aa (text input)
- QnA Dataset ID** (global): c074cf51-5297-4ce0-8b25-4523a66f (text input)
- Enable Statistics Tab on Product Page** (global): No (dropdown menu)
- Enable Statistics Tab on Order Page** (global): Yes (dropdown menu)
- Order Statistics Workspace ID** (global): b6c02c33-eee6-4e33-b109-1b0cb3fc (text input)
- Order Statistics Report ID** (global): 39aadeab-cbcf-4452-ba15-d117ae37 (text input)
- Enable Statistics Tab on Customer Page** (global): No (dropdown menu)

Picture: Power BI Embedded content configuration inside Magento Platform

**Embedding Mode** allows configuring the embedding mode that will be used for the embedded content. Read “Embedded Modes” section of this documentation for more information on the modes.

**Enable QnA Section on Overview Page** will add Question & Answer section on Magento Platform’s overview page (home page of Admin Panel).

**QnA Workspace ID** includes ID of the workspace from Power BI Service. This workspace should include a dataset that will be used for QnA. The easiest way to find out this value is to copy this value from the URL after navigating to the workspace inside Power BI Service.

**QnA Dataset ID** includes ID of the dataset from Power BI Service that will be used for the QnA section. The easiest way to get this value for a dataset is to copy this value from the URL after navigating to the dataset inside Power BI Service.

**Enable Statistics Tab on Product Page** allows enabling on the statistics tab on products page in Magento Platform.

**Product Statistics Workspace ID** includes ID of the workspace from Power BI Service. This workspace should include a report that will be embedded to statistics tab on product edit page. The easiest way to find out this value is to copy this value from the URL after navigating to the workspace inside Power BI Service.

**Product Statistics Report ID** includes ID of the report from Power BI Service that will be embedded to statistics tab on product edit page. The easiest way to find out this value is to copy this value from the URL after navigating to the report inside Power BI Service.

**Enable Statistics Tab on Order Page** allows enabling on the statistics tab on order details in Magento Platform.

**Order Statistics Workspace ID** includes ID of the workspace from Power BI Service. This workspace should include a report that will be embedded to statistics tab on order details page. The easiest way to find out this value is to copy this value from the URL after navigating to the workspace inside Power BI Service.

**Order Statistics Report ID** includes ID of the report from Power BI Service that will be embedded to statistics tab on order details page. The easiest way to find out this value is to copy this value from the URL after navigating to the report inside Power BI Service.

**Enable Statistics Tab on Customer Page** allows enabling on the statistics tab on customer details in Magento Platform.

**Customer Statistics Workspace ID** includes ID of the workspace from Power BI Service. This workspace should include a report that will be embedded to statistics tab on customer details page. The easiest way to find out this value is to copy this value from the URL after navigating to the workspace inside Power BI Service.

**Customer Statistics Report ID** includes ID of the report from Power BI Service that will be embedded to statistics tab on customer details page. The easiest way to find out this value is to copy this value from the URL after navigating to the report inside Power BI Service.

# ODATA API

---

## ODATA API OVERVIEW

---

OData API provides an additional way of accessing sales data of Magento platform based on Sales Cube (and other datasets in the future).

OData supports wider range of BI software, so it is not only limited to Power BI.

With OData API you might be able to integrate Magento data with other BI Vendors. Among the list of BI software vendors that support OData are:

- Power BI
- Qlik
- Tableau
- Tibco Spotfire
- MicroStrategy 10.5 supports now OData via Data Import
- Dundas
- SAP
- Targit

Our implementation of OData API is tested against Power BI and is not guaranteed to work with other software. Please, contact support if you wish to utilize the OData API using other BI software.

OData connectivity might be used to create very customized reports which will be harder to create with default Sales Cube data model. Also, it could be used for other cases where more data manipulation could be done before reports creation. This includes possibilities to introduce custom grouping, custom data transformations and merging with other data sources.

Using OData API, users may add their Magento data to a Power BI Service Dataflow.

## ACTIVATING THE ODATA API

---

Usage of OData requires installation of Power BI Sales Cube Magento extension with OData support. Please, request access to this functionality if it is missing in your installation.

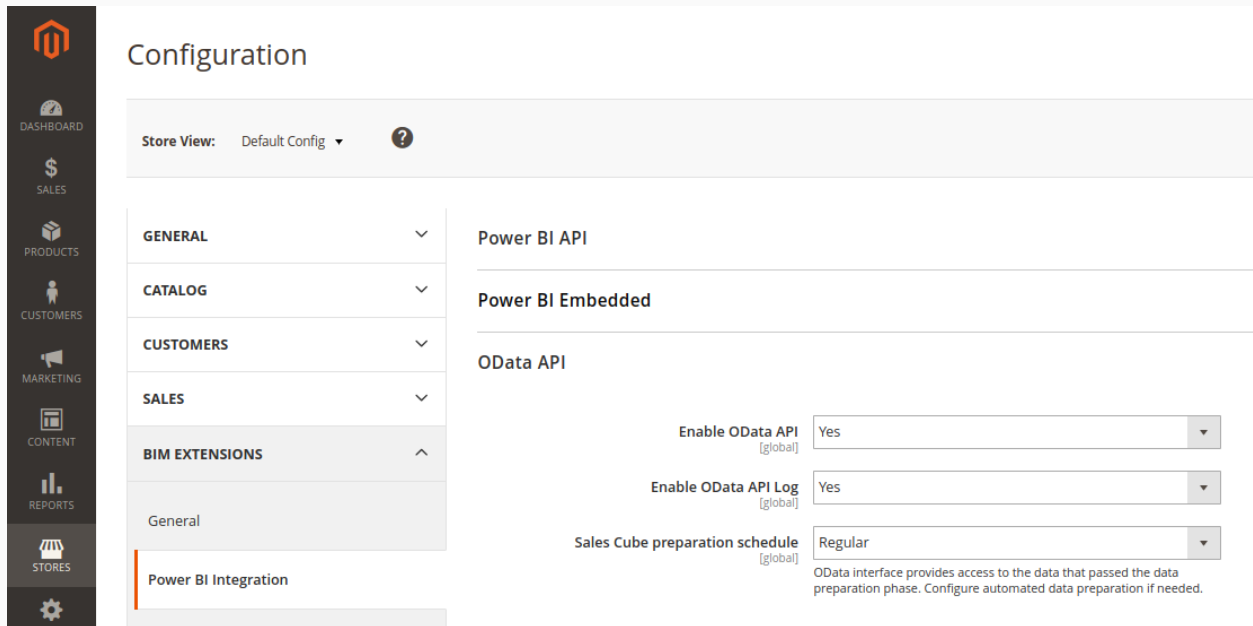
Navigate to Magento Configuration page (**Stores -> Configuration**) and select **BIM EXTENSIONS -> Power BI Integration -> OData API** section.

Set "**Enable OData API**" to "**Yes**" to enable the API.

Set "**Enable OData API Log**" to "**Yes**" if you are facing issues with usage of OData API. This will write the communication with OData service to the log file that could help to resolve those issues with our support.



Option **“Sales Cube preparation schedule”** is useful if your data needs to be refreshed periodically in order to be prepared for being read through the OData API. Read more details on it in the section **“Scheduling data actualization for OData API”**.



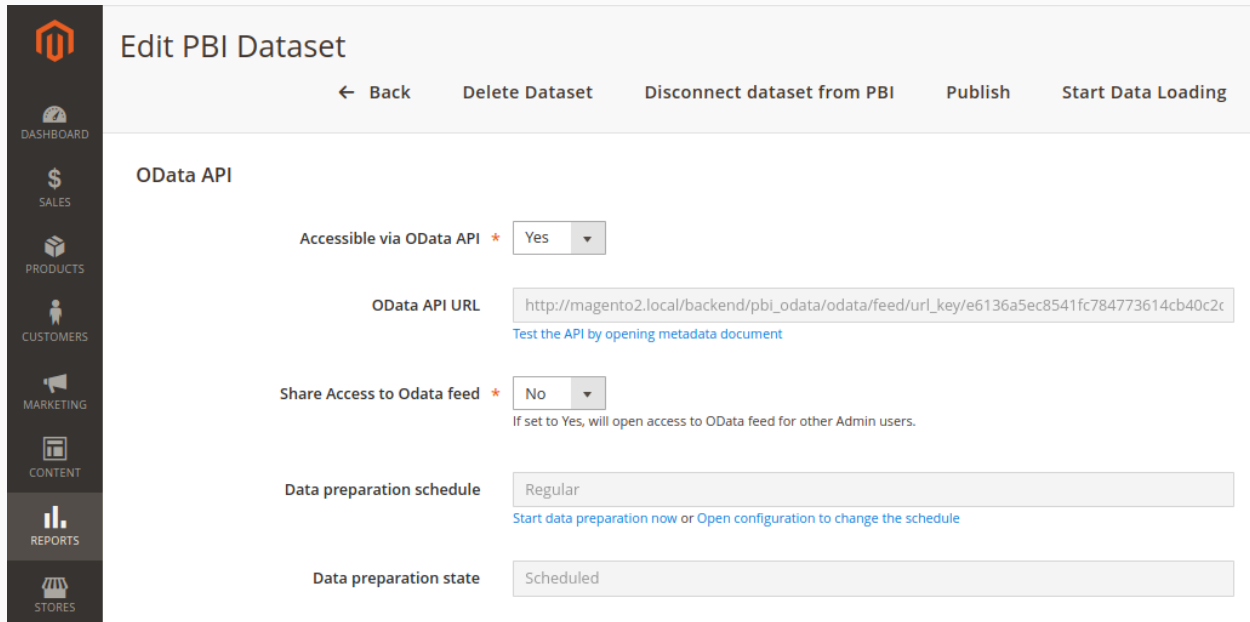
Picture: OData API Configuration for Magento Platform

## ACTIVATING ODATA ACCESS TO A DATASET

Once you have created your dataset, you may activate access to this dataset through OData. In order to do this, save the dataset with option **“Accessible via OData API”** set to **“Yes”**.

Setting **“Shared Access to OData feed”** to **“Yes”** would allow other admin users to access the dataset via the API. If this option is left with its default value **“No”**, only the owner of the dataset will be able to access it. If only certain user group should be able to use the API, the access control list could be modified by disabling/enabling **“BIM OData API”** resource.

Please, note that data for the dataset could only be read after data preparation will be run for the dataset. In order to trigger the data preparation, you may click on the **“Start data preparation now”** link under **Data Preparation Schedule** field.

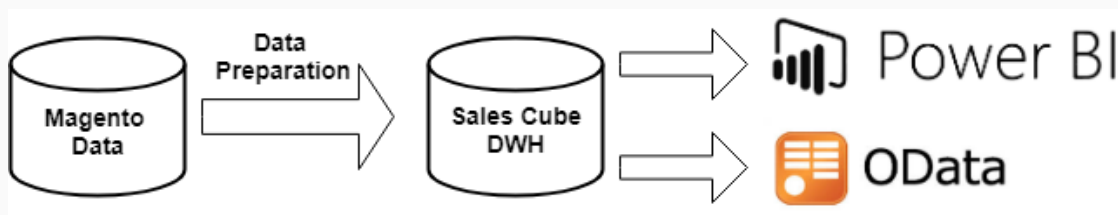


Picture: OData API Section inside Dataset properties page in Magento Platform

## SCHEDULING DATA ACTUALIZATION FOR ODATA API

Data that is available through OData interface is not just the underlying data of the Magento system. It is the data, prepared for the analytical processing. And it is stored separately from the original data.

The data preparation step will move the data from Magento operation system into a small Data Warehouse for faster reading for analytical purpose. Simple schematic of this process is presented on the picture below.



Picture: Data Preparation in Power BI Integration process

Therefore, such data preparation should happen every time before the data is accessed via OData API. This data preparation process is very smart and will only process the new data that was not processed before.

Such process could be triggered by the schedule. In case if the dataset is loading to Power BI by the schedule, such data preparation is already scheduled. It should not even be the same dataset. This data preparation is done for all datasets at once.

For such cases when automated loading of the data to Power BI datasets is not activated, users can configure data preparation step especially to keep data prepared for OData access. In such cases, users can navigate to **OData API** configuration section as described in “**Activating the OData API**” section before and select desired schedule in the option “**Sales Cube preparation schedule**”.

Alternatively, users can always trigger the data preparation manually. This could be done on dataset properties page by clicking on the small link “**Start data preparation now**” under **Data Preparation Schedule** field.

## LIMITATIONS

---

There are certain limitations associated with Power BI API that put some boundaries on the usage of the extension.

Here is the list of known limitations of Power BI Service APIs:

- 75 max columns
- 75 max tables
- 10,000 max rows per single POST rows request
- 1,000,000 rows added per hour per dataset
- 5 max pending POST rows requests per dataset
- 120 POST rows requests per minute per dataset
- If table has 250,000 or more rows, 120 POST rows requests per hour per dataset
- 200,000 max rows stored per table in FIFO dataset
- 5,000,000 max rows stored per table in 'none retention policy' dataset
- 4,000 characters per value for string column in POST rows operation

Additionally there are limitations on the capacity depending on used Power BI subscription

- Max of 1 GB of storage for Power BI Free subscription
- Max of 10 GB of storage for Power BI Pro subscription
- Max of 100 GB of storage per Power BI Premium / Power BI Embedded instance

Free subscription does not support functionality that allows sharing Power BI Service content with colleagues.

Please, refer to official Power BI documentation from Microsoft for more recent information about the limitations of Power BI Service.

## LIMITATIONS OF REPORT BASED TABLE PROFILES

---

Dataset with Table Profiles does not support incremental load. Therefore each time, data of a table is loaded to Power BI Service, this table gets truncated and new data is sent.

Amount of rows sent to a dataset within an hour should not go over 1 000 000 rows. For the Power BI Integration Extension, this means the following:

- Amount of rows in a dataset should not go over 1,000,000 rows.
- If the dataset is updated often than once per hour, limit of 1,000,000 rows should be divided by frequency of data actualizations per hour. Example: dataset updated each 10 minutes will be updated 6 times per hour, thus it should have no more than  $1,000,000 / 6 = 166,666$  rows.

## LIMITATIONS OF SALES CUBE DATASET

---

Data of Sales Cube Dataset cannot be loaded faster than 1,000,000 rows per hour.

Total limit for the amount of rows per dataset is limited by Power BI Service: max 75 tables with 5,000,000 rows per each table.

Some limitations related to data in the Sales Cube Dataset are described in “Sales Cube Data Dictionary” document.

## USAGE EXAMPLES

---

This section is intended to help users to easier gain most value of the integration with Power BI Service. The use cases described here might not be directly related to Power BI Integration Extensions, but cover more of the possibilities that users will get after the data will be delivered to Power BI Service.

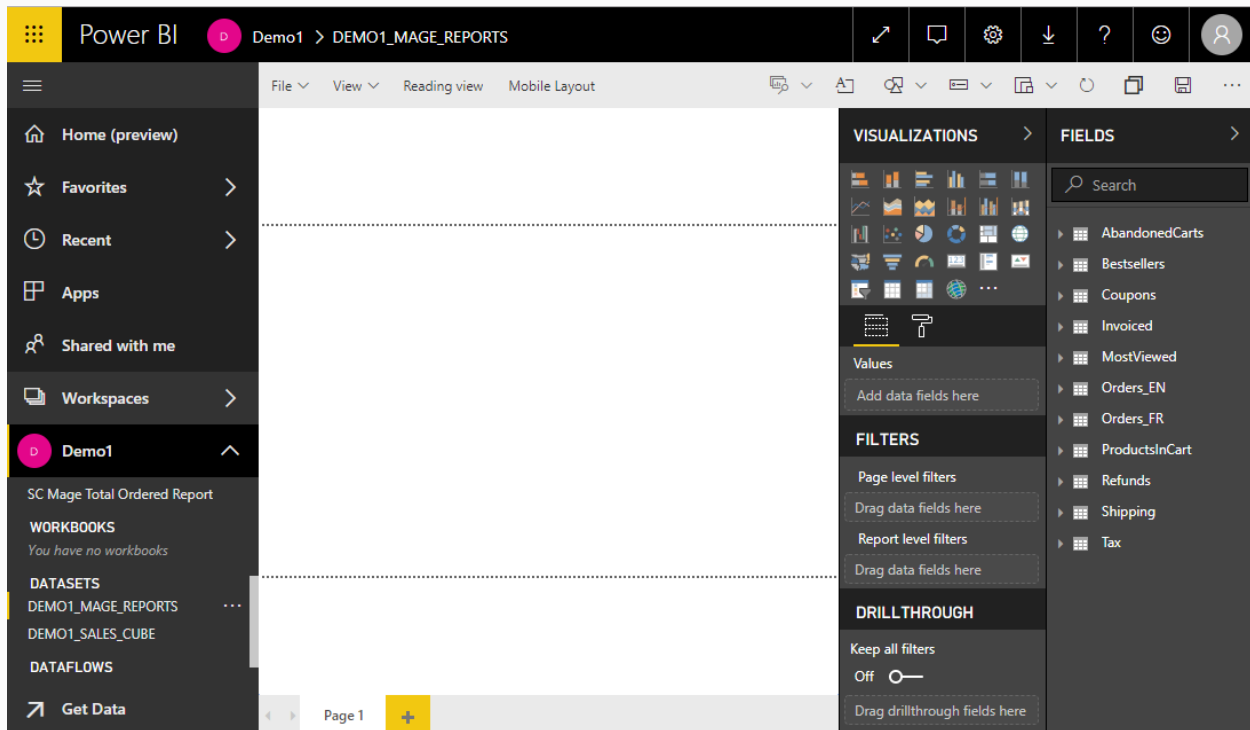
Below you will find some useful references and some notes on how to use features of Power BI Service that will be available to Magneto users with the Power BI Integration Extensions: reports creation, dashboards, access through the mobile App and usage of functionality of alerts and notifications.

### CREATING POWER BI REPORTS IN POWER BI SERVICE

---

To create a report inside Power BI Service:

1. Open Power BI Service.
2. In the list of the datasets, select a workspace and the dataset with data from Magento Platform that will be used as data-source for the report. After selecting the dataset, user will see the empty report that can be filled in with data
3. Drag and drop desired data visualizations and the data to the report and adjust their look as desired.
4. Click File -> Save to save the report. Report will be saved in the same workspace as the dataset.



Picture: Power BI Report creation screen

Please, refer to the Power BI documentation for more detailed example of reports creation process:

<https://powerbi.microsoft.com/en-us/documentation/powerbi-service-create-a-new-report/#add-a-radial-gauge-to-the-report>

---

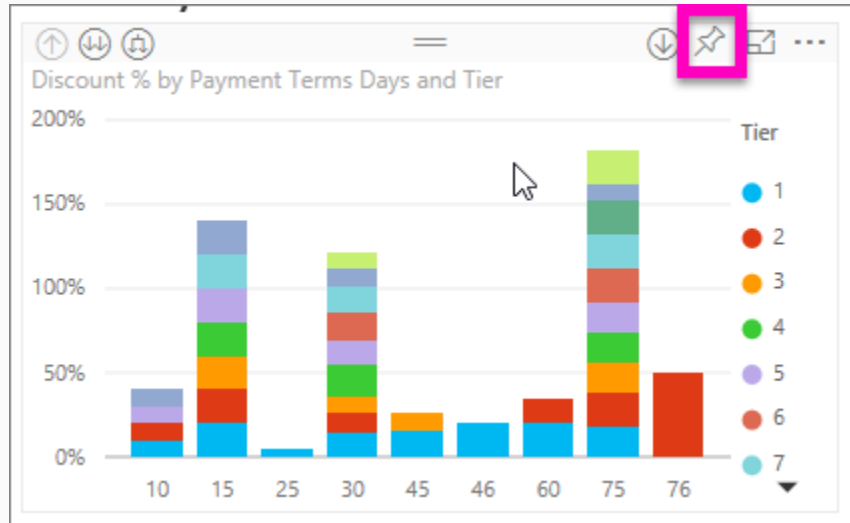
## CREATING POWER BI DASHBOARDS

---

Dashboards are useful because they allow user to combine data from different sources on a single view (not on a single data visualization component).

To start with dashboards creation, users need to create some report(s) first.

1. Open existing report by navigating to the workspace with the report and select the report.
2. Switch the report from “Reading view” to “Editing view”.
3. Each data visualization in the report have “Pin to dashboard” button:



Picture: "Pin to dashboard" button inside Power BI Service

4. In the opened dialogue user can select option to add visualization to existing dashboard or to create a new dashboard. Select "New Dashboard" and give it a name to create a new dashboard.
5. Confirm the action and Power BI Service will create a new dashboard.
6. After that action, Power BI Service may navigate user to the created Dashboard.

Please, refer to the Power BI documentation for the information on how you can create dashboards:

<https://powerbi.microsoft.com/en-us/documentation/powerbi-service-create-a-dashboard/>

## CREATING REPORTS IN POWER BI DESKTOP

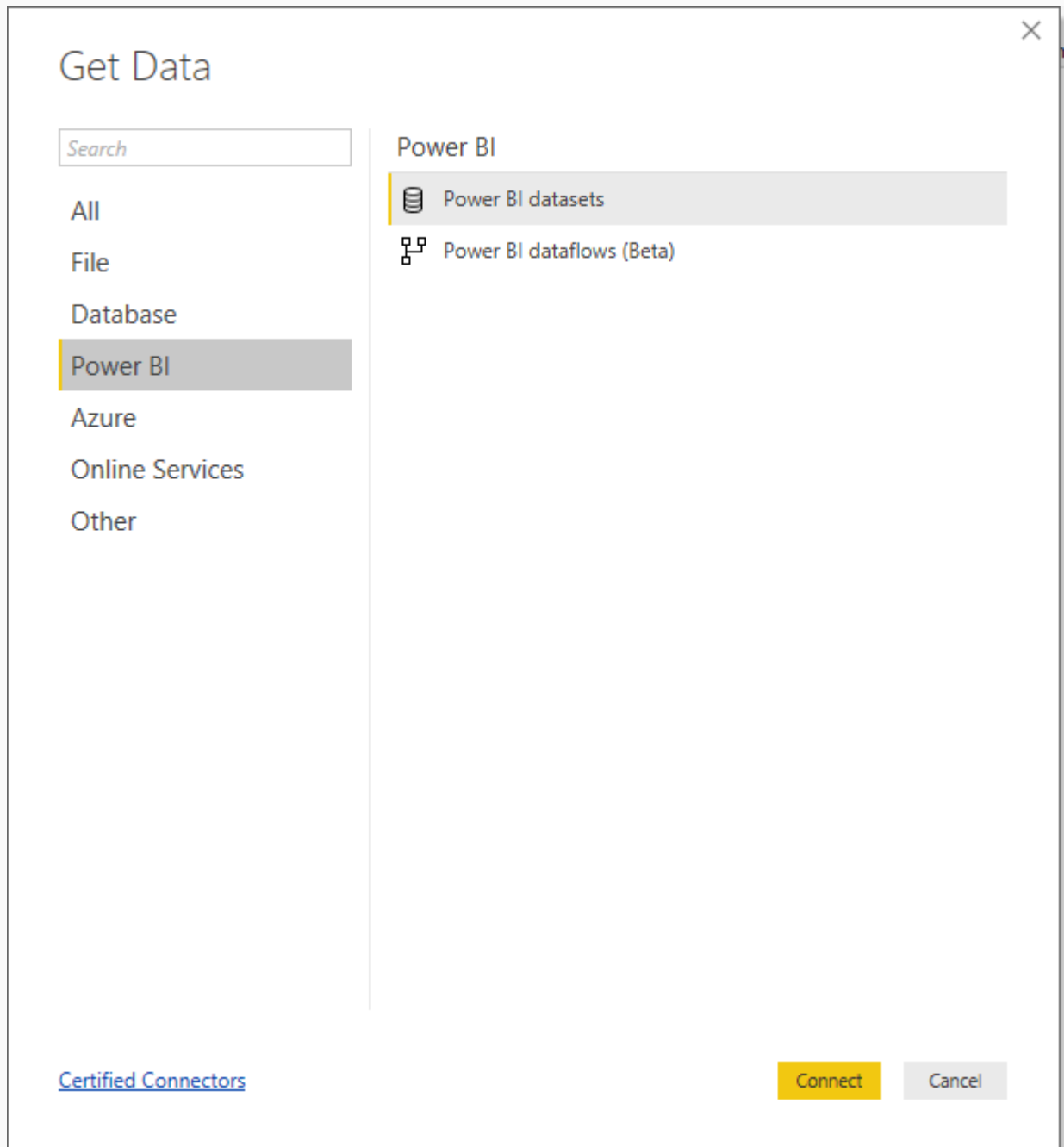
Users are not only restricted by Power BI Service, but also can use Power BI Desktop while operating with Magento data delivered by Power BI Integration Extension.

Power BI Desktop may have more advanced features compared to Power BI Service. In particular, with Power BI Desktop users may add calculated measures on top of the dataset created by Power BI Integration Extension.

To create a report using Power BI Desktop, do the following:

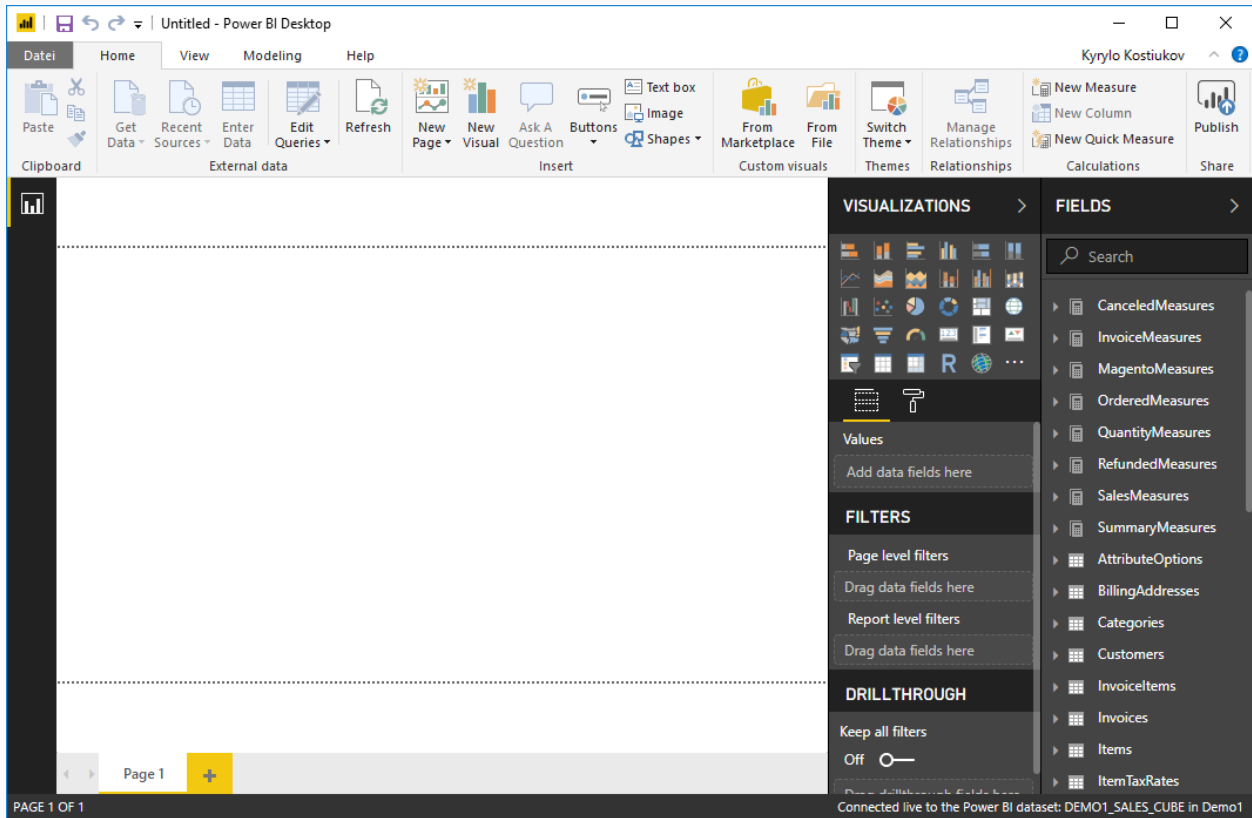
1. Install Power BI Desktop on PC or laptop with Windows OS.
2. Open Power BI Desktop. If asked to enter user information, click to "Already have a Power BI account? Sign in" and sign in using credentials for Power BI Service account.
3. Click on "Get data", Get Data dialogue will be shown.





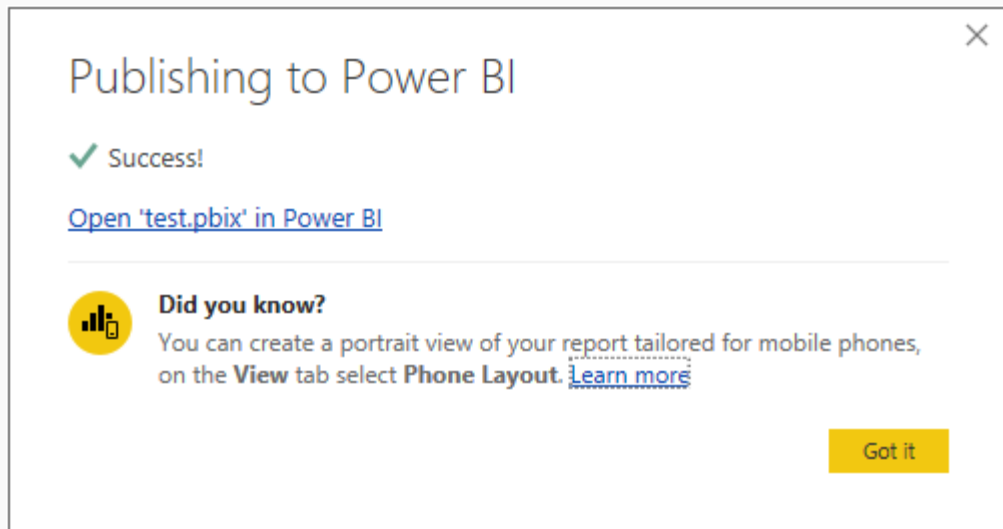
Picture: Get Data dialogue inside Power BI Desktop

4. Select "Power BI" -> "Power BI datasets" and click "Connect".
5. Next dialogue will allow to navigate to desired workspace and to select a dataset.
6. After connecting to the dataset, user can proceed and add visualizations to the report.



Picture: Report creation screen inside Power BI Desktop

7. Once the report is ready, user can save the report file on the filesystem: “File” -> “Save”.
8. Once report is saved, it could be published to Power BI Service. Click on “Publish” button on the “Home” tab.
9. If the report was not saved, Power BI will ask to save the report with a given name and will create a copy of the report in Power BI Service.



*Picture: Publish to Power BI dialogue*

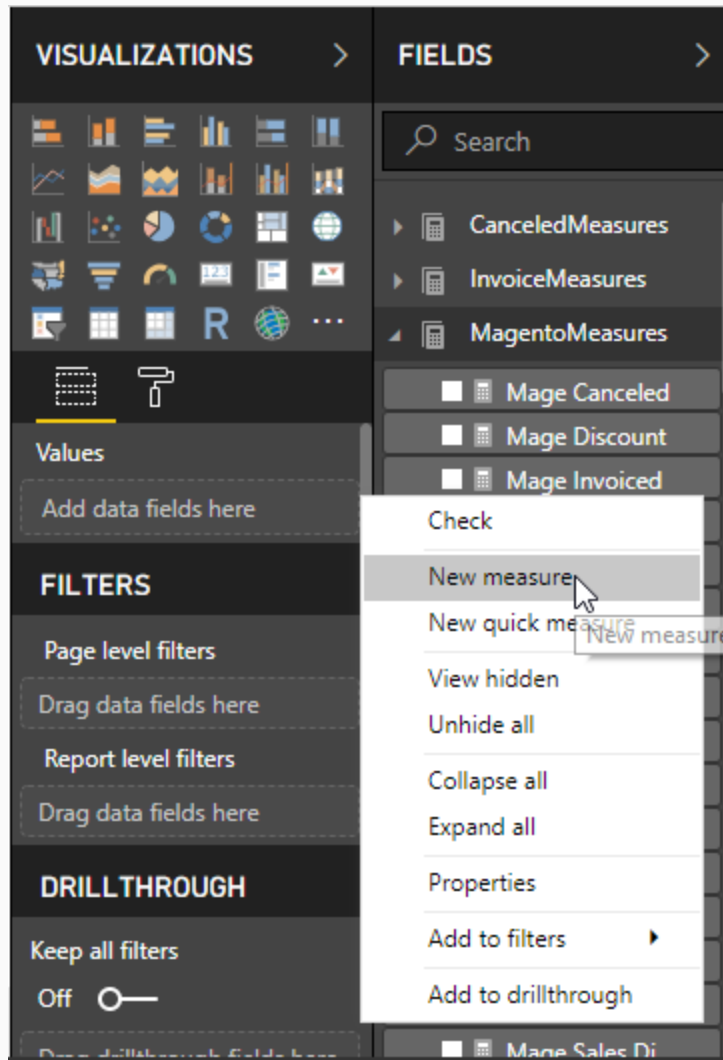
---

## ADD CALCULATED MEASURES

---

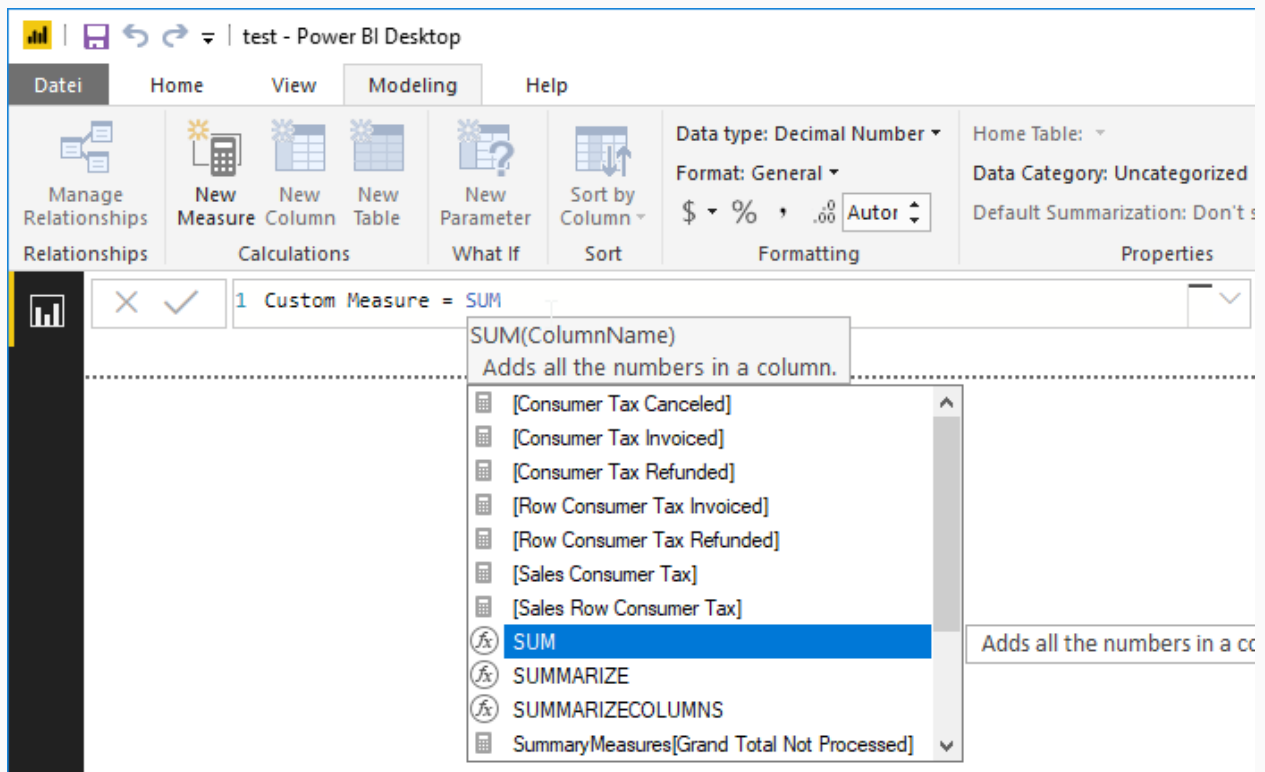
One of the big advantages of using Power BI Desktop is the ability to create Calculated Measures.

1. To create a measure, connect to a dataset from Power BI Service as described in “Creating reports in Power BI Desktop section”.
2. In Power BI Desktop, hover over a measure and select “New Measure” in the context menu or navigate to “New Measure” button on “Modeling” tab.



Picture: "New measure" button for creating calculated measures inside Power BI Desktop

3. Enter a formula following DAX syntax. More about the syntax, read in Power BI Documentation from Microsoft.



Picture: entering DAX expression for calculated measure inside Power BI Desktop

4. Optionally you may select different “Home Table” using the configuration on the “Modelling” part.
5. Confirm the measure creation and the new measure will be added to the data model and will be ready to be used on data visualization components.

## ACCESS CONTENT WITH POWER BI MOBILE APP

Users who have their own account in Power BI Service can access the content with Power BI Mobile App. There are different apps available for iPhone, iPad, Android (phone/tablet), Windows Phone.

Power BI Mobile Apps provides unique experience to Power BI users. In addition to being able to see recent data through mobile devices, user can also get notifications and make and read annotations shared with other users.

Here is the list of recent possibilities of Power BI Mobile App for Android devices:

- View your reports.
- View your dashboards.
- Explore the tiles on your dashboards.

- Open Power BI reports.
- Annotate and share tiles.
- Share dashboards.
- Scan a Power BI QR code with your Android phone to open a related dashboard tile or report.
- View notifications about updates to your Power BI account, such as dashboards that colleagues share with you.
- Get alerts about reached KPI targets

To get up and running with Power BI Mobile, do the following:

1. Install Power BI Mobile app for your device.
2. Sign in with your Power BI account credentials.

Please, use the following page to find the information about installing a Mobile App on your device:

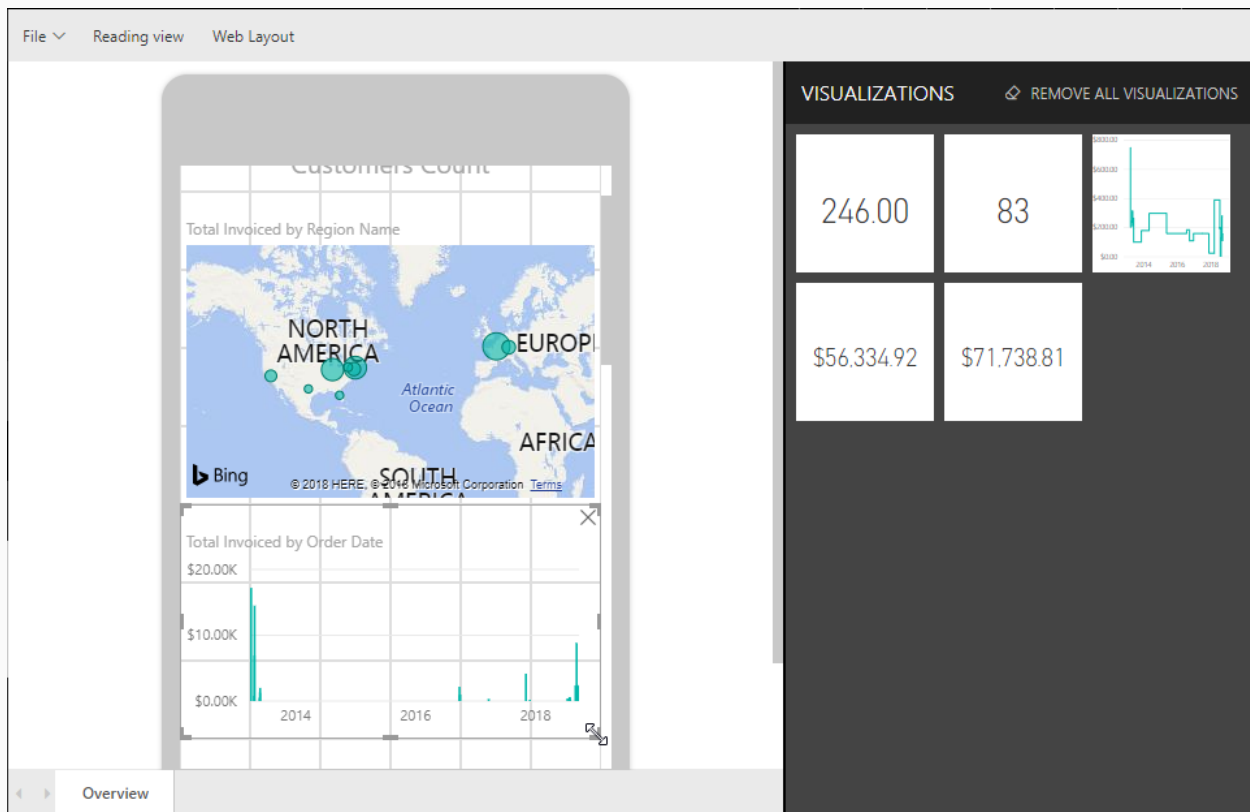
1. <https://powerbi.microsoft.com/en-us/mobile/>

After logging in with your Power BI user's credentials you will be able to access Reports and Dashboards you have created in Power BI Service. Power BI Pro users can also access content shared with them.

Users may also control look of the reports in mobile app by switching the "Editor view" in to "Mobile layout" in Power BI Service or in Power BI Desktop.

In order to adjust the report for mobile device, do the following:

1. Open a report in Power BI Service or in Power BI Desktop.
2. Switch report to "Editing view".
3. Switch to "Mobile layout".



Picture: configuring Mobile Layout for a report inside Power BI Service

## SETUP NOTIFICATIONS (ALERTS), REPORTS SCHEDULE

Setting up notifications may appear a bit tricky since this functionality does not seem to be very intuitive at the beginning. To setup notifications, you need to have a dashboard with a tile that supports notifications. Simple example is a KPI. So, before setting up notifications, you need to create a dashboard. Please, refer to the documentation of Power BI for more information:

<https://powerbi.microsoft.com/en-us/documentation/powerbi-service-set-data-alerts/>

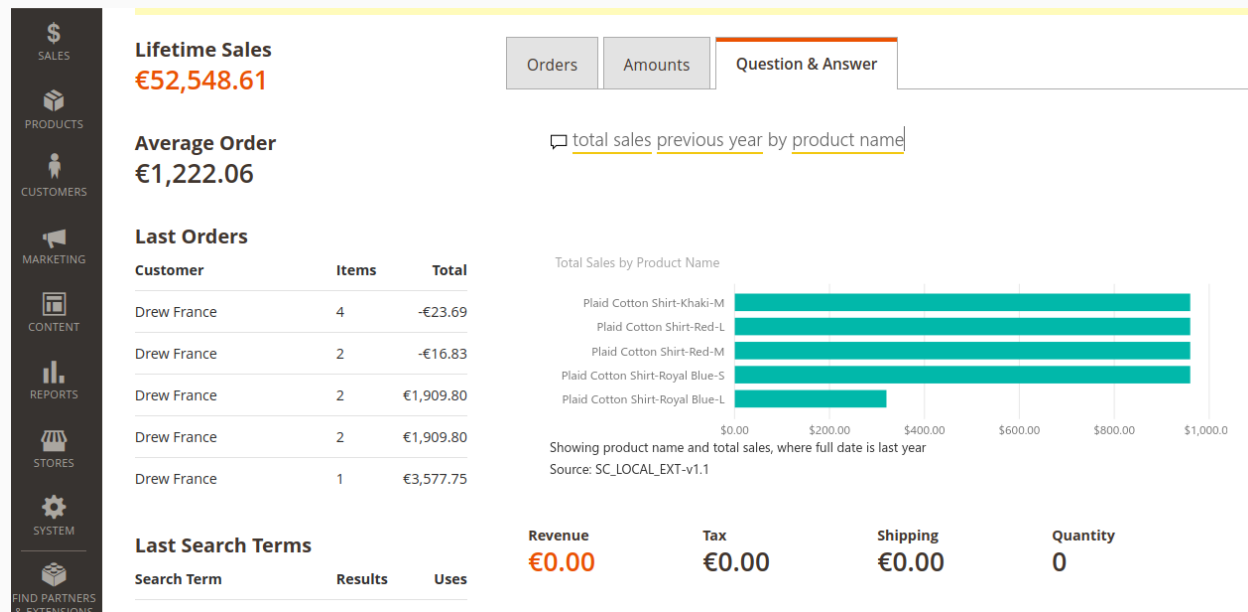
Being able to subscribe to reporting was always an attribute of a complete BI solution. Power BI users would require having Power BI Pro subscription in order to be able to subscribe for emails with their reports. Please refer to Power BI documentation for more information about subscriptions:

<https://docs.microsoft.com/en-us/power-bi/consumer/end-user-subscribe>

## INTEGRATION WITH Q&A

Power BI Dataset published to Power BI Service can be used as a data source for Q&A functionality.

Q&A functionality provides possibility to write questions in natural language and get answer based on data in Power BI dataset.



Picture: Example of an embedded Q&A content

This is very powerful feature that allows getting quick insight into areas that might not even be covered by reports or dashboards.

At the moment, it only supports questions asked in English. Functionality requires having Power BI Pro or Power BI Premium subscription or can also be used as part of content provided to non-registered users through Power BI Embedded.

The Q&A can be accessed through various apps:

- As embedded functionality in Magento Platform
- On dashboards in Power BI Service
- On a dashboard in Power BI Mobile

The dataset should be well optimized for being used as source for Q&A and to provide meaningful answers to the asked questions. A good example of well optimized dataset for Q&A is **Sales Cube Dataset**.

## INTEGRATION WITH CORTANA

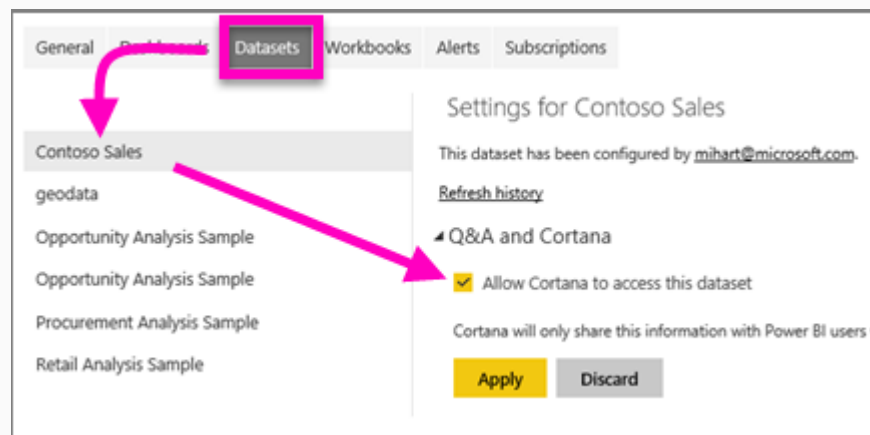
Power BI Dataset published to Power BI Service can be used as a data source for Cortana.



Cortana is a virtual assistant created by Microsoft and can be used with wide range of devices, some of them are Windows 10, Windows Phone, Microsoft Band, Surface Headphones, Xbox One, iOS, Android, Windows Mixed Reality, and Amazon Alexa. Unfortunately, not all Android mobile devices are supported.

To allow Cortana to have access to the dataset, simply do the following steps after a dataset with data from Magento Platform will be delivered to Power BI service:

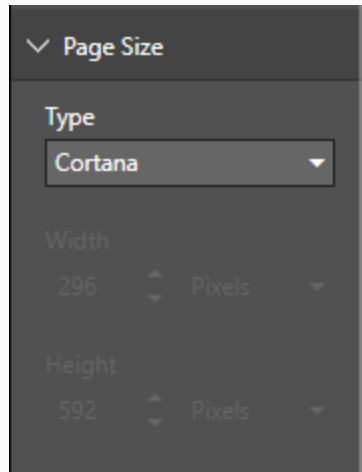
1. Login to Power BI Service.
2. Navigate to Datasets tab in the Workspace
3. Select the dataset from the list of datasets.
4. Select *Q&A and Cortana > Allow Cortana to access this dataset -> Apply*



Picture: Allowing Cortana to access to a dataset

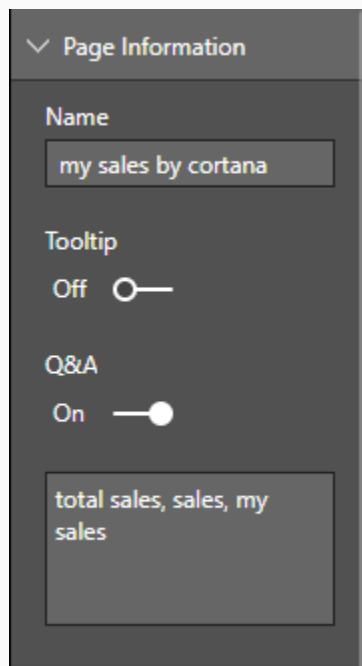
To be able to get answers from Cortana, user should have dashboards or special kind of reports that includes a “Cortana answer cards”. Follow the following instructions to create a report with “Cortana answer card”:

1. Open or create a report that is connected with a dataset that is allowed to be used by Cortana.
2. Switch to the page of the report that will be used as “Cortana answer card”.
3. Change Page Size settings to “Cortana”



Picture: Configuring report properties for a report to be accessible by Cortana, Page Size

4. Fill in Page Information, enable Q&A and add alternative page names that could be used in queries to Cortana.



Picture: Configuring report properties for a report to be accessible by Cortana, Page Information

5. Save the report.

More details about how to create “Cortana answer cards” you will find in Power BI documentation:

<https://docs.microsoft.com/en-us/power-bi/service-cortana-answer-cards>

It might also be required to connect your Cortana to Office 365. This can be done by making changes in Windows settings related to apps used that are activated by Cortana.

After that setup is ready, you may try to make a request to Cortana like “powerbi show net sales”. The phrase should consist of at least 2 words, where “show” and “powerbi” will not count as a word.

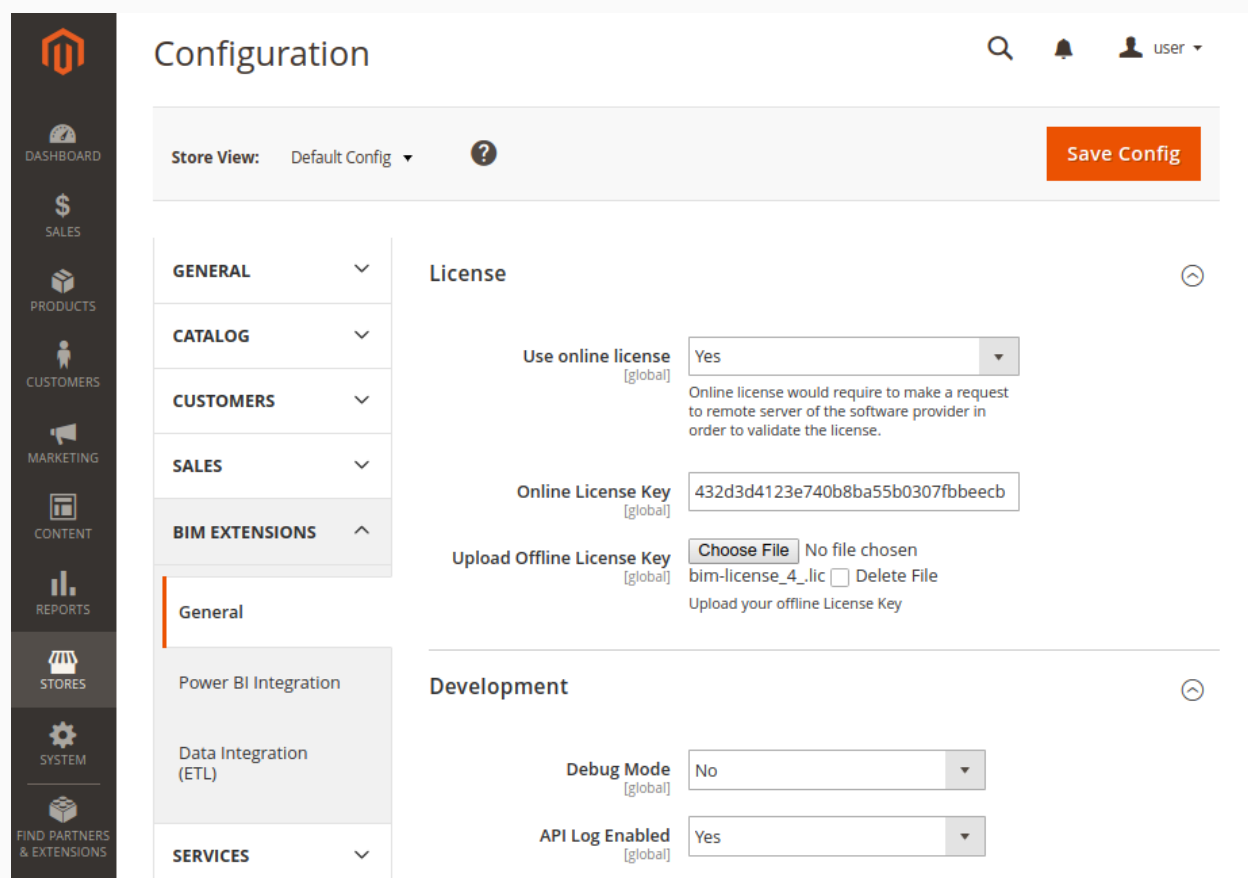
Please, note that Cortana - Power BI integration may work only with English language. More details about setting up Cortana and Power BI to work together you will find in related Power BI documentation:

<https://docs.microsoft.com/en-us/power-bi/service-cortana-intro>

# TROUBLESHOOTING

## DEVELOPMENT CONFIGURATION OPTIONS

Users can configure some development options by navigating to Magneto Configuration (*Stores -> Configuration* in Magneto 2 or *System -> Configuration* in Magento 1), section *BIM Extensions -> General -> Development*.



The screenshot displays the Magneto Configuration interface. On the left is a vertical sidebar with navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area is titled 'Configuration' and shows a 'Store View' dropdown set to 'Default Config' and a 'Save Config' button. The 'BIM EXTENSIONS' section is expanded to show 'General' and 'Development' sub-sections. Under 'License', the 'Use online license' is set to 'Yes', the 'Online License Key' is '432d3d4123e740b8ba55b0307fbbeebc', and the 'Upload Offline License Key' section shows a 'Choose File' button and a 'Delete File' checkbox. Under 'Development', the 'Debug Mode' is set to 'No' and the 'API Log Enabled' is set to 'Yes'.

Picture: Development configuration options

**Debug Mode** option may enable/disable some additional debugging output or control elements that help to get more insights for debugging/development purposes. Since such information may reveal some protected information such as location of system files to all admin users, it is recommended to set this option set to “No” on production/preproduction servers.

**API Log Enabled** option enables/disables output of communication between Magento Platform and Power BI REST API. Since such log may get quite bit with time, it is not recommended to set this option to “No” on production server.

## DUMP OF DEBUGGING INFO

---

Power BI Integration Extensions come with functionality that allows developers to easily prepare an archive file with debugging information that includes configuration and recent logs from the extensions.

In order to prepare a debugging dump, the following command needs to be executed from the command line:

```
cd <magento-base-dir>

# in Magento 2:
bin/magento bim:debug:dump

# in Magento 1:
cd shell
php bim_app.php prepare_debug_info
```

# ADVANCED FEATURES

## SCHEMA EDITOR

If more flexibility is required, users may also change the schema definition of the dataset. Typical use cases might be:

- duplicate some tables to create reports that have two filters over the same dimension, introducing role-playing dimensions
- hide some columns, tables
- adding tables with custom made measures
- changing relationships between tables

In order to open schema editor, navigate to datasets list and select action “Edit Schema (advanced)” in the actions menu as presented on the picture below.

The screenshot shows the 'PBI Datasets - Data Load to Power BI' interface. It includes a sidebar with navigation options like Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, and System. The main area displays connection status (Enabled), automatic data actualization (Enabled), and a table of 5 records. The 'NewDatasetSample' dataset is highlighted, and its actions menu is open, showing options like Publish, Start Loading, One Click Load, and Edit Schema (advanced).

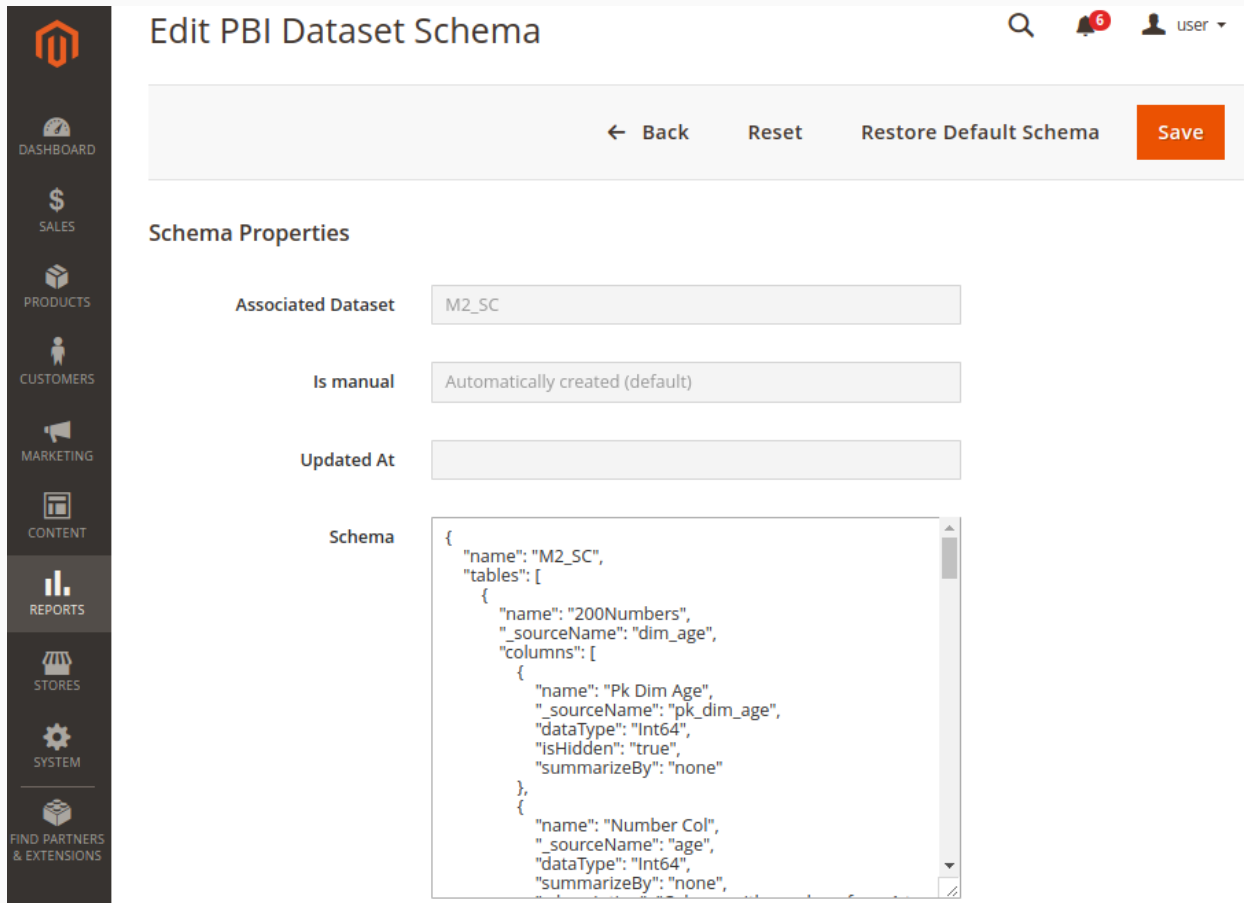
Dataset Name	Dataset Type	Is Temporary	Workspace	Created At	Load Mode	Schema Publishing	Data Loading	Action
DEMO1_MAGE_REPORTS	Mixed	Permanent	Demo1	11/5/18, 7:44 AM	Manual	PUBLISHED	INACTIVE	[Dropdown]
DEMO1_SALES_CUBE	Sales Cube	Permanent	Demo1	11/5/18, 7:51 AM	Manual	PUBLISHED	INACTIVE	[Dropdown]
M2_SC	Sales Cube	Permanent	My Workspace	11/3/18, 1:02 PM	Regular	PUBLISHED	SCHEDULED	[Dropdown]
NewDatasetSample	Mixed	Permanent	My Workspace	12/14/18, 6:00 AM	Manual	NOT PUBLISHED	INACTIVE	[Dropdown]
PBI_TEMP	Mixed	Temporary	My Workspace	11/4/18, 1:25 AM		PUBLISHED	INACTIVE	[Dropdown]

Picture: actions menu with action for opening the Schema Editor

Sales Cube dataset will have quite advanced schema from by default. You may use this schema as an example to understand how it could be adjusted.

Once finished, press “Save” button to save the changes in the schema. The changes will not be applied to the dataset in Power BI. To apply this changes, the dataset will have to be published again. In case if changes in the schema are not adding or removing any tables, the changes might be

applied without reloading of the data. But in some cases, data reload or even, delete/publish actions might be required.



Picture: Schema Editor

In order to cancel the manual changes in the dataset, simply press on “Restore Default Schema”.

## REPORT TEMPLATES

---

### USE CASES

Report Templates is a functionality that allows to use Power BI APIs to download and upload Power BI reports from Magento Platform to Power BI Service and back.

Please, not that this is not the same templates that could be created from Power BI Desktop.

This functionality is very useful for some use cases and gives possibility to save the work that was done by user during report creation. In particular, it becomes possible to do the following:

1. Receive initial set of reports with purchasing of some Power BI Integration Extensions.
2. Download Power BI reports created on top of a dataset that needs to be deleted.
3. Migrate Power BI reports from one dataset to another.
4. Backup Power BI reports to make sure, they will not be lost.
5. Restore the backed up Power BI reports.
6. Migrate Power BI reports from one account to another (if workspaces could not be used for some reason).
7. Migrate Power BI reports from one workspace to another.

## LIST OF REPORT TEMPLATES

List of Report Templates could be found by navigating to *Reports -> Power BI Templates*.

The screenshot shows the 'PBI Report Templates' page in the Magento Platform. The page has a dark sidebar on the left with icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, and System. The main content area is white and features a search bar, a notification bell, and a user profile icon labeled 'user'. Below these is a prominent orange button that says 'Create Template from remote Report'. The central part of the page is a table with the following data:

Template Name	File Name	Source	Description	Action
SC Embedded Customer Statistics	embedded-customer-statistics-report.pbix	Sales Cube Module	Report for embedding as customer statistics page. Based on Sales Cube dataset.	<a href="#">View</a>
SC Embedded Order Statistics	embedded-order-statistics-report.pbix	Sales Cube Module	Report for embedding as order statistics page. Based on Sales Cube dataset.	<a href="#">View</a>
SC Embedded Product Statistics	embedded-product-statistics-report.pbix	Sales Cube Module	Report for embedding as product statistics page. Based on Sales Cube dataset.	<a href="#">View</a>
SC Mage Order Taxes Report	sc-mage-order-taxes-report.pbix	Sales Cube Module	Replication of Magento Order Taxes Report based on Sales Cube.	<a href="#">View</a>
SC Mage Total Invoiced vs Paid Report	sc-mage-total-invoiced-report.pbix	Sales Cube Module	Replication of Magento Total Invoiced vs Paid Report based on Sales Cube.	<a href="#">View</a>
SC Mage Total Ordered Report	sc-mage-total-ordered-report.pbix	Sales Cube Module	Replication of Magento Total Ordered Report based on Sales Cube.	<a href="#">View</a>

Picture: list of report templates inside Magento Platform

**Template Name** column shows a name given to the report template.

**File Name** column has a report name in form of the file name.

**Source** column shows how the template was created, where it comes from.

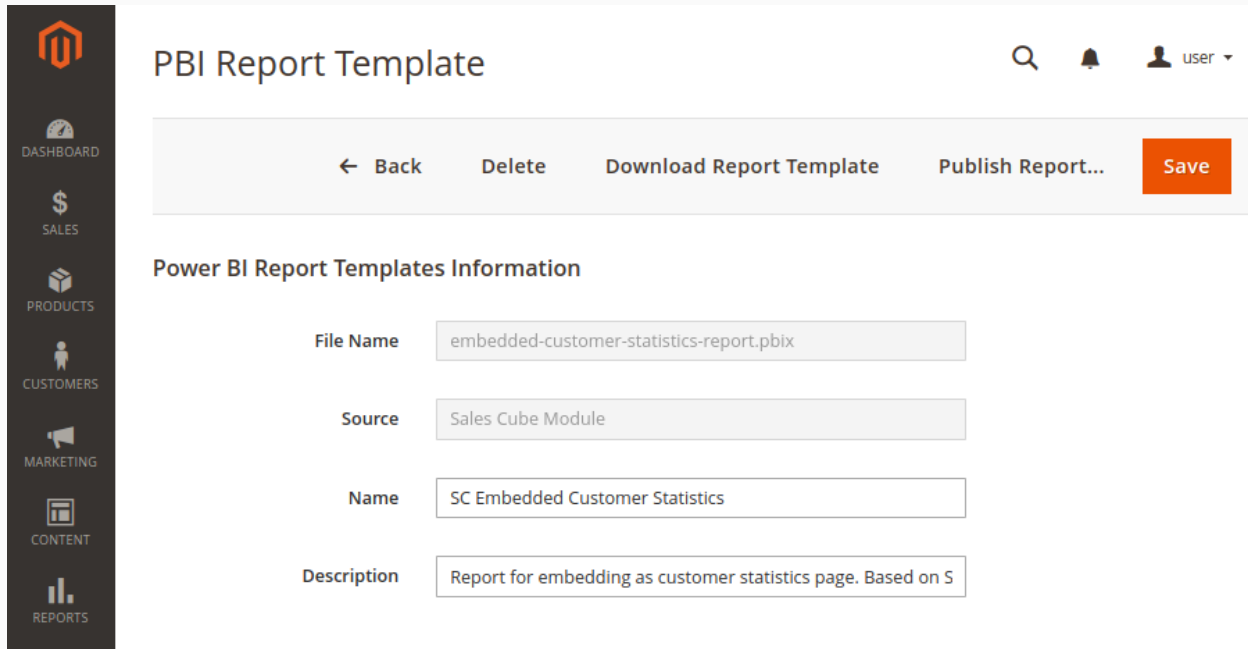
**Description** provides short description of the content and the purpose of the report.

Quick action column helps users to navigate to template details page.



## REPORT DETAILS PAGE

Opening the Template Details page will show a form where user can change some properties of the Report Template.



The screenshot shows the 'PBI Report Template' details page. On the left is a dark sidebar with navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, and Reports. The main content area has a header with the title 'PBI Report Template', a search icon, a notification bell, and a user profile dropdown labeled 'user'. Below the header is a toolbar with buttons: 'Back', 'Delete', 'Download Report Template', 'Publish Report...', and a prominent orange 'Save' button. The main section is titled 'Power BI Report Templates Information' and contains a form with the following fields:

File Name	embedded-customer-statistics-report.pbix
Source	Sales Cube Module
Name	SC Embedded Customer Statistics
Description	Report for embedding as customer statistics page. Based on S

*Picture: report template details page*

**Delete** action will delete the report template from the Magento Platform.

**Download Report Template** action will download the report as pbix file.

**Publish Report** action will redirect user to report publishing dialogue to upload a report template to Power BI and create Power BI report based on the template.

## UPLOADING OF THE REPORT

To upload a Report Template, open template details and click on “Publish Report” button.

This will open the report publishing form where user can selecte target workspace, dataset that will be bound to the report and the name for the report that will be created in Power BI Service.

## DOWNLOADING OF THE REPORT

To download the report from Power BI Service and save it as a Report Template, user can press button “Create Template from remote Report” on the page with Report Templates list.

This will open the form where user can select the remote report to download and specify the properties of the report template.

The screenshot shows the 'PBI Report Export' interface. At the top, there's a search icon, a notification bell, and a user profile icon labeled 'user'. Below this is a navigation bar with a 'Back' link and an orange 'Export Report' button. The main section is titled 'Remote Report' and contains four input fields: 'Report to export from Power BI' (a dropdown menu showing 'My Workspace / test2'), 'New Template Name' (a text input), 'File name' (a text input with a note: 'Allowed characters are latin letters, numbers, dash and dots: sample-report-v1.1. File extension will be added automatically.'), and 'New Template Description' (a larger text area). The left sidebar has icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, and System.

*Picture: exporting report template form*

**Report to export from Power BI** option allows user to select a workspace and a report in Power BI Service.

**New Template Name** will include short name for the report template.

**File Name** is a name that will be given to the report for internal storage and while downloading of the pbix file from Magento Platform.

**New Template Description** should include short description of the content and the purpose of the report.

Click on “Export Report” will trigger the downloading process and will create a new Report Template.

---

## ETL JOBS

---

Most of the functionality of the Power BI Integration extension is using scheduler that performs ETL jobs. ETL translates into Extract-Transform-Load. Those jobs are mostly responsible for data handling: extraction of data from Magento Platform, transforming the data for fitting the expected format and structure for Power BI Service, loading of the data to Power BI.

List of ETL jobs can be opened by navigating to *Reports -> ETL Jobs*. There user can see the list of the scheduled jobs and some information about them.

The screenshot shows the 'BIM ETL Jobs' interface. It features a sidebar with navigation options: DASHBOARD, SALES, PRODUCTS, CUSTOMERS, MARKETING, CONTENT, REPORTS (highlighted), STORES, and SYSTEM. The main content area displays a table of ETL jobs with the following columns: Job Type, Job ID, Schedule, Status, Error, Ex. Time, Last Run, Next Run, and Action. The table contains 7 rows of job data.

Job Type	Job ID	Schedule	Status	Error	Ex. Time	Last Run	Next Run	Action
Setup	pbi-sales-cube-drop-dwh-schema	Manual			-	2018-09-28 13:08:11	-	
Setup	pbi-sales-cube-drop-staging-schema	Manual			-	2018-09-28 13:08:11	-	
Setup	pbi-sales-cube-truncate-dwh-schema	Manual			-	-	-	
Setup	pbi-sales-cube-truncate-staging-schema	Manual			-	-	-	
Setup	pbi-sales-cube-setup-dwh-schema	Manual			6 ms	2018-09-28 13:08:11	-	
Setup	pbi-sales-cube-setup-staging-schema	Manual			3 ms	2018-09-28 13:08:11	-	
Export	job_lastday_statistics_update	Live			2.3 sec	2018-11-08 06:40:28	2018-11-08 06:45:00	

Picture: list of ETL Jobs

**Job Type** includes indication of the nature of the job. Possible values are: “Setup”, “Export” (data extraction), “Transform” and “Load”.

**Job ID** is internal unique identified of the job.

**Schedule** is the schedule by which the job is running. Please, note some jobs might be triggered as dependencies from other jobs, not only by its own schedule.

**Status** column shows the current status: if the job is in the jobs queue or is already running. Empty values means that job is not in the currently processed queue.

**Error** column may indicate if there were some issues with latest run of the job.

**Ex. Time** column shows execution time – how long it took to process this job last time it was run.

**Last Run** – date and time when the job was run the last time.

**Next Run** - date and time when the job is planned to be run for the next time.

User can also perform some quick actions on the jobs:

- **Run** action will trigger the processing of the job;
- **Run Chain** action will trigger the job and all other jobs that this job depends on.

## USAGE OF DEV TOOLS

Additionally Power BI Integration Extensions come with extra page that includes some functionality that might be useful during the development and maintenance.

User can open Dev Tools page by navigating to *Reports -> BIM Dev Tools*.

**BIM Development Tools**

**BIM Development Tools**

**BIM Power BI Integration - Development Tools**

**Warning:** Usage of functionality presented on this page may break the functionality of your Power BI datasets. Please, use it only if you know what you are doing.

**Dataset assignment**

This functionality is useful to recover the dataset which was already published to Power BI and user has dashboards and reports created based on that dataset. The form will help to attach dataset which was not published yet to remote dataset.

Click here to go to [Dataset assignment form](#).

**Dataset assignment removal**

This functionality might be useful if dataset assignment was not working as expected. May also be used in order to switch dataset ownership between different systems or for other rare cases.

Click here to go to [Dataset assignment removal form](#).

**BIM Sales Cube - Development Tools**

**Sales Cube Data Dictionary**

Click here to view [Sales Cube Data Dictionary](#).

**Order Export**

This functionality allows to export data of single order for debugging and testing purposes.

Click here to go to [Order export form](#).

**Sales Cube Reload (Clean Up)**

This functionality intended to be used for troubleshooting the Sales Cube by reloading all its data.

**Warning:** reload of the DWH requires consequent data clean up and reload of published Sales Cube datasets. In the same way the reload of the staging area will require the reload of DWH and published datasets. This form does only clean up of the data. The reload will happen once next the data loading will be triggered for a sales cube dataset.

Click here to go to [Sales Cube Clean Up Form](#).

**Sales Cube Capacity Check**

The information given by capacity check gives some estimation on how big the data in datasets' is and if it fits the capacity of Power BI Service.

The analysis is based on DWH data. In order to update the dwh, you may [run preprocessing logic manually](#) (might take some time).

Click here to do [Capacity Check](#).

*Picture: Dev Tools page*

This page includes some descriptions on the listed functionality could be used. Here we review some use cases when such functionality might be useful.

## ASSIGNING DATASETS AND ASSIGNMENT REMOVAL

From the Dev Tools page user can navigate to “Dataset assignment form” and to “Dataset assignment removal form”.

The standard workflow of using the Power BI Integration Extensions is to create a dataset in Magento Platform, publish it to Power BI Service and then just keep loading data to that dataset. But in some cases this workflow might need to have some adjustments. The manual dataset assignment functionality helps to manually assign dataset inside Magento Platform to a dataset in Power BI Service.

PBI Dataset assignment

← Back   Reset   Check Compatibility

Local Dataset

Local Dataset

Dataset name DEMO1\_MAGE\_REPORTS

Remote Dataset

Remote Dataset

Workspace Name, Dataset name de0746 - My Workspace / BIM

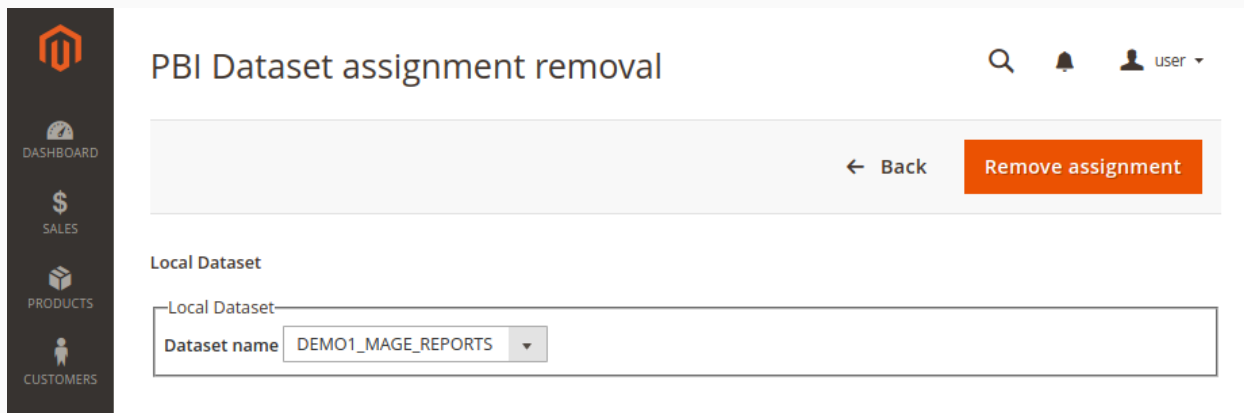
*Picture: dataset assignment form*

On the dataset assignment form, user selects the dataset from Magento Platform, then the dataset in Power BI Service and first need to run the compatibility check.

**Check Compatibility** will compare the structure of both datasets to validate that there will be no error happening during loading of the data.

This functionality does not guarantee the complete compatibility between the datasets, but helps to prevent some of the errors.

If the system did not determine incompatibilities between the datasets, user will be offered to assign the datasets together. After the assignment, it becomes possible to load the data to Power BI dataset that was disconnected from Magento before.



Picture: dataset assignment removal form

Dataset assignment removal is quite simple form that only accepts the dataset inside Magento Platform. Triggering the “**Remove assignment**” action will put the dataset from “Published” state to a “Not Published”.

---

## SALES CUBE DEV TOOLS

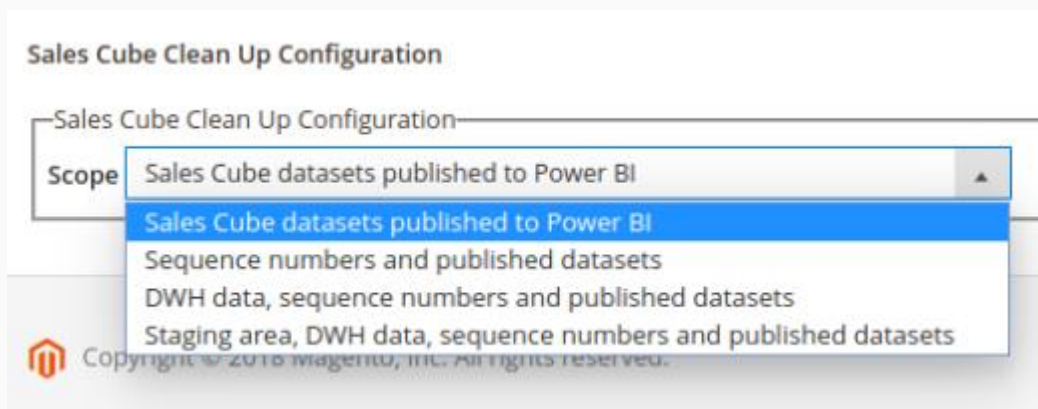
---

If the Sales Cube Dataset functionality is available in Magneto, user will get access to some development tools for maintenance of Sales Cube Datasets.

Most important actions are: doing manual Sales Cube clean-up and performing capacity check.

### SALES CUBE CLEAN-UP

From Dev Tools page users can navigate to Sales Cube clean-up page. The clean-up might be configured to cover different scope.



Picture: Configuring sales cube clean-up tool

**Sales Cube datasets published to Power BI** will only clean-up all datasets published to Power BI. Please, note that not only datasets of current admin user will be affected, but all Sales Cube Datasets created in Magento Platform and linked with remote Power BI datasets.

**Sequence numbers and published datasets** will do the same as previous option, but will also clean-up sequence numbers. Clean-up of sequence numbers is useful because it will increase the performance of the reload of the data. Sequence numbers influence the amount of rows that will be sent with every data load request. If the iterative load was running for some time, sequence numbers will not be generated efficiently.

**DWH data, sequence numbers and published datasets** will do the same as previous option, but will also clean-up DWH data. This option is useful if the DWH building was changed and the data transformation logic needs to be run again.

**Staging area, DWH data, sequence numbers and published datasets** covers full scope of data handling for Sales Cube Datasets. This scope covers the scope of previous option, but will additionally clean-up the staging area. Staging area is where the data are stored after being extracted from Magento Platform. This means that the data will need to be extracted from Magento Platform again. This is the option that needs to be selected if user wants to refresh completely the data inside the DWH. This will do Slowly Changing Dimensions (SCD) refreshing by updating product names, category names, linking between categories and products and so on. Read about SCD in “Sales Cube Data Dictionary” document.

#### SALES CUBE CAPACITY CHECK

Due to limitations of Power BI, even Power BI Pro users may not always be able to load full history of data within one dataset in Power BI Service. Loading of data to Power BI Service may take several days until all data that will fill in the dataset will be delivered to Power BI. In order to be able to know upfront if the data for a dataset will fit inside Power BI Service, user can do Capacity Check.

The outcome of the capacity check will show what the maximum amount of rows for a single table is and will show amount of tables that exceed maximum allowed size of 5,000,000 rows.

#### CHANGING THE STRUCTURE OF PERMANENT DATASET

---

It is better to plan your dashboards and reports development in advance and start with dataset that contains all data that will be needed. This is not always possible and sometimes it might be required to update the dataset structure even though it has already been published and was used to create some reports. It does not apply to Sales Cube dataset which might be changed only during upgrade of the Power BI Integration Extension.

If you want to modify Permanent Dataset, you will need to first remove the link to Power BI Dataset. After removing the link from the old Power BI Dataset it will not be possible to update data

in that dataset. Read “Advanced features: Assigning datasets and assignment removal” to learn how to unlink a Dataset from Power BI Service.

Once modifications are done, the Dataset can be published again as a new Power BI Dataset. Alternatively it could be possible to try to assign modified dataset to the remote Power BI dataset, but only if modifications are not related to amount of tables or to table names.

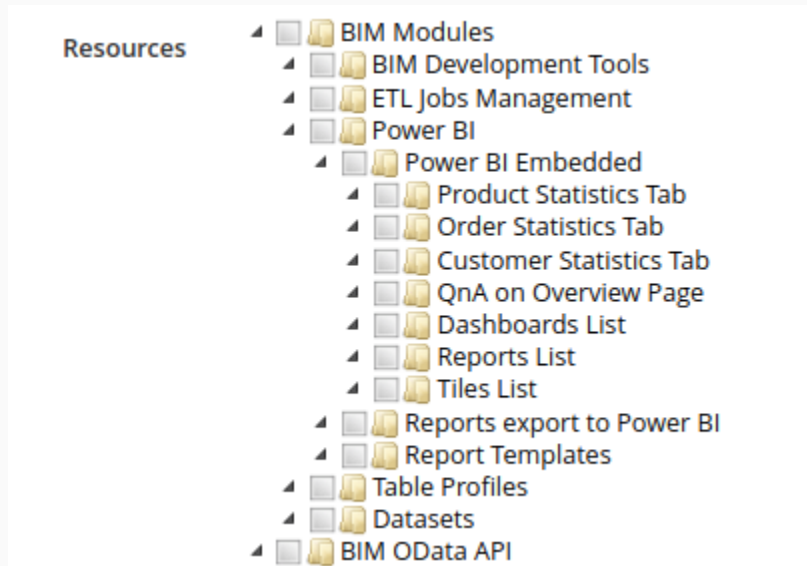
Users who have Report Templates functionality coming with their Power BI Integration Extension can create Templates from the reports created based on the old version of the dataset and publish those Templates with link to the new Dataset. Read “Advanced features: Report Templates” to learn how to use Templates.

---

## CONFIGURING ACCESS CONTROL

---

Magento supports flexible configuration of access rights for different backend user roles. The access could be restricted to certain resources. The Power BI Integration extensions introduce additional access control resources.



*Picture: access control resources*

Most of them will restrict access to the underlying backend pages. “BIM OData API” resource allows restricting access to OData API feed.



## FINAL NOTES

---

### ROADMAP

---

Power BI product from Microsoft is relatively young product but very promising. Almost every month they publish their release notes.

The extension integrates Magento store with Power BI Online offering and the more functionality Microsoft provides, the more Magento store owners can benefit from the integration.

One of the main areas that will allow enhancing the extension in the future is the API. Some of the great improvements such as specifying relations between datasets are already announced and we are working on extending the functionality based on those improvements.

Please, contact extension provider to get more details about the road map.

### EXTENDING THE FUNCTIONALITY

---

The license of the extension allows store owners to develop additional functionality by extending the base functionality provided with the module.

One use case is **extending the data delivered to Power BI**. Each store owner may have different strategy and may want to concentrate on specific data, not covered by current set of data delivered to Power BI Service. This would require creation of custom data grid or just a php class that will play role of a data provider. Such provider might be connected to the extension to deliver the data to Power BI. Please, contact extension provider to request development documentation and IT support that may also include professional services to help you deliver the data you want.

The Power BI Integration Extension might be used to transfer more data than it is supported at the moment. Current architecture supports creation of modules on top of Power BI Integration that will allow connecting more data.

Another use case might be **embedding** of the reports, dashboards or just separate tiles into other areas of Magento. For example, one might want to add some KPIs or charts on main dashboard of Magento backend. This will be possible by requesting the code samples for embedding functionality to Magento Platform from extension provider. Such support might require extra IT Support package.

Please note that redistribution of the extension or its modifications is not allowed without agreement with the extension provider.

Please, contact the extension provider if you are interested in extending the functionality.