****

Analytics Application Installation Guide

Document N.: Analytics Application Installation Guide

Revision: 1.0

Date: 13/03/2021

**Creator:** Raunak Sharma

For further information please contact:

[support@projectobjects.com](mailto:support@projectobjects.com)

© 2021 Project Objects Ltd

Table of contents

[1. Overview 2](#_Toc66725188)

[2. Analytics Installation Requirements: 2](#_Toc66725189)

[3. Configure Reporting and Clone Databases 2](#_Toc66725190)

[4. Configure Application on server machine 2](#_Toc66725191)

[4.1. Application Pool Creation 3](#_Toc66725192)

[4.1.1. Create Application Pool for POXAnalytics Application 3](#_Toc66725193)

[4.1.2. Create Application Pool for POXAnalyticsAPI Application 3](#_Toc66725194)

[4.1.3. Create Application Pool for ReportingAPI\_POXAnalytics Application 4](#_Toc66725195)

[4.2. Giving Full Access to Applications 4](#_Toc66725196)

[4.3. Configuration of Application In IIS 5](#_Toc66725197)

[4.3.1. ReportingAPI\_POXAnalytics 5](#_Toc66725198)

[4.3.2. POXAnalyticsAPI 6](#_Toc66725199)

[4.3.2.1. Generate Encrypted Password for POXAnalyticsAPI 7](#_Toc66725200)

[4.3.3. POXAnalytics 8](#_Toc66725201)

[4.4. Redirection from PO Application 11](#_Toc66725202)

[5. Configure Publish Process 11](#_Toc66725203)

# 

# Overview

This document is primed to describe the installation process of PO Analytics application over On-premises customers.

# Analytics Installation Requirements:

System must be equipped with the below mentioned software prior to the installation of PO Application. Although the Analytics application can be installed on the same application server where the POX application installed, but we recommend to use the different server for it for better experience.

|  |  |
| --- | --- |
| **Component** | **Minimum** |
| Access | Remote desktop or Web-ex or TeamViewer Sessions details before scheduled time. |
| Server | Window Server 2012 R2 Enterprise |
| Processor | 3.10 gigahertz (GHz) |
| RAM | 56 Gigabyte (GB) |
| Disk | NTFS file system–formatted partition with a minimum of 3 GB of free space |
| Display | 1366 × 768 |
| Network | 56 kilobits per second (Kbps) connection between client computers and server |
| IIS | IIS server 7 or above |
| IIS Folders | Read and write access of the folder C:\inetpub\wwwroot\ |
| SQL Server | SQL Server 2019 |
| Framework | Dot net framework 2, 3.5, 4, 4.5, & 4.7.2 |
| ASP.NET Core SDK | 3.1 & 5.0.301 |
| ASP.NET Core Runtime | 3.1 & 5.0.7 |
| Windows Server Hosting | 3.1.3 & 5.0.7 |

# Configure Reporting and Clone Databases

For making the Analytics application work, we will require to configure 2 more databases:

1. Reporting Database
2. Clone Database

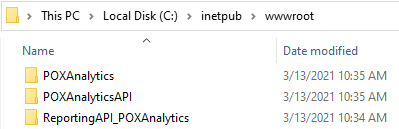
Please restore the reporting database and Clone database in the same database server where the POX transaction database exists. These two databases must have the similar access as the transaction database. And they must be accessible from the application servers as well.

# Configure Application on server machine

There are 3 applications required to be configured to make the analytics application work.

1. POXAnalytics
2. POXAnalyticsAPI
3. ReportingAPI\_POXAnalytics

Copy the provided application (mentioned above) under “C:\inetpub\wwwroot” folder and proceed with individual applications as mentioned in below steps.



## Application Pool Creation

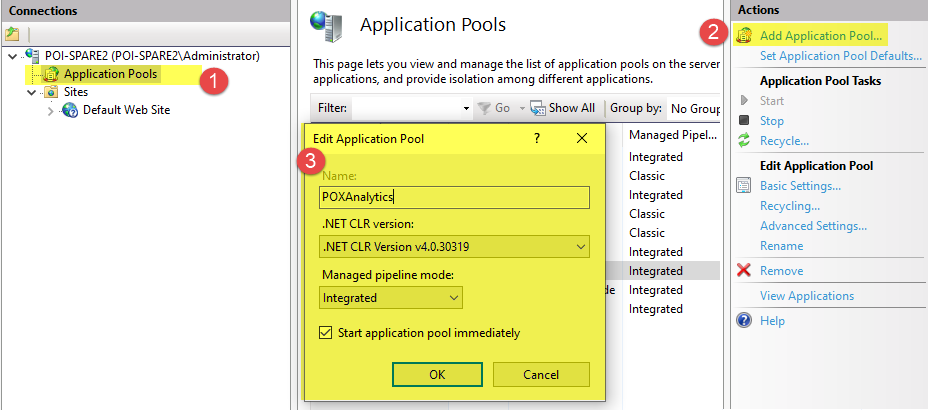
We require to create 3 application pools for these application to configure.

### Create Application Pool for POXAnalytics Application

**Step 1**: Open IIS

**Step 2:** Go to **Application Pools**, click on Add Application Pool and select the highlighted options (in below screen).

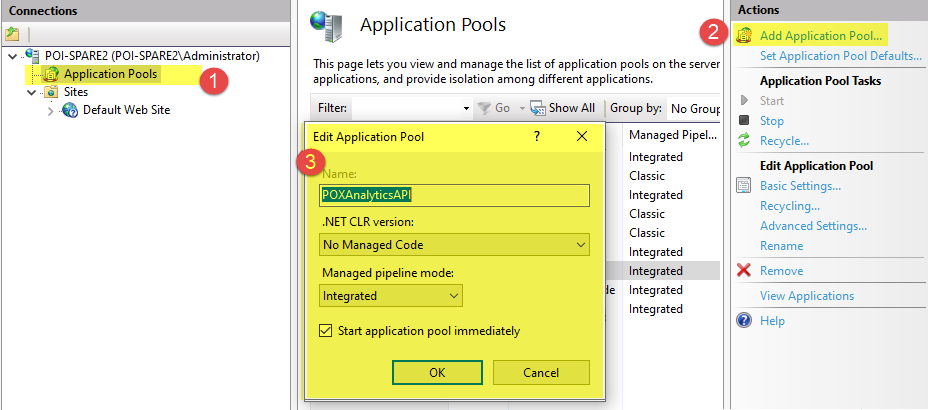
**Step 3**: Click Ok. It will create the application pool.



### Create Application Pool for POXAnalyticsAPI Application

**Step 1**: Open IIS

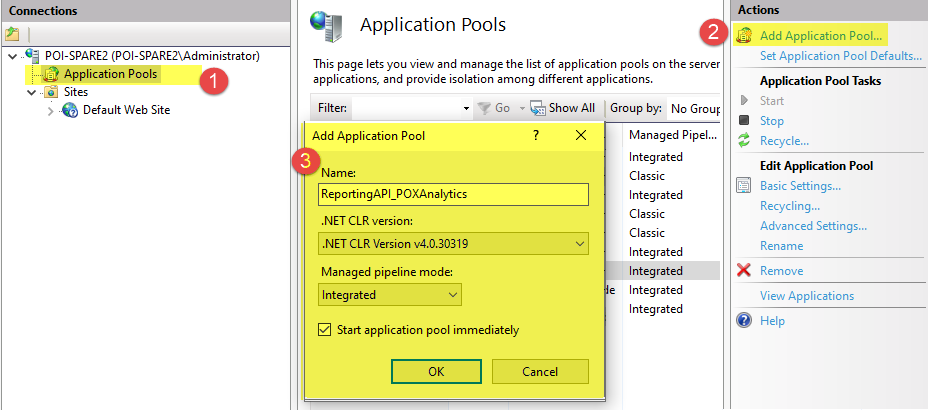
**Step 2:** Go to **Application Pools**, click on Add Application Pool and select the highlighted options (in below screen).  **Step 3**: Click Ok. It will create the application pool.



### Create Application Pool for ReportingAPI\_POXAnalytics Application

**Step 1**: Open IIS

**Step 2:** Go to **Application Pools**, click on Add Application Pool and select the highlighted options (in below screen).  **Step 3**: Click Ok. It will create the application pool.

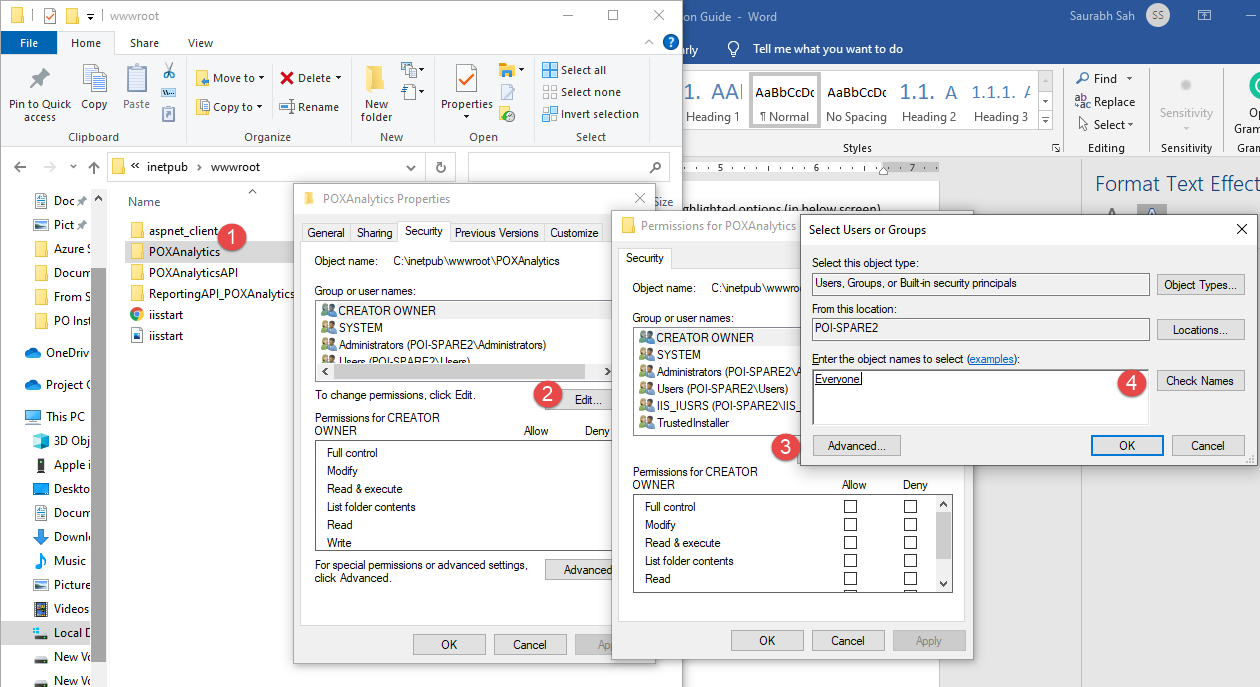


## Giving Full Access to Applications

**Step 1**: Go to “C:\inetpub\wwwroot”

**Step 2**: Right click on application and select **Properties**

**Step 3**: Add “**Everyone**” and give full access.

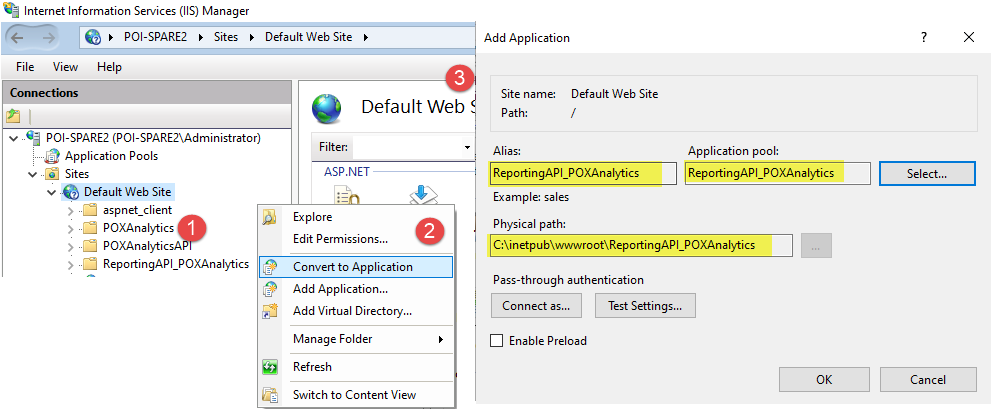


## Configuration of Application In IIS

### ReportingAPI\_POXAnalytics

**Step 1**: Go to IIS and right click on **ReportingAPI\_POXAnalytics**.

**Step 2**: Select “**Convert to Application**” and select the highlighted values, and click **Ok**.



**Step 3**: Open the **ReportingAPI\_POXAnalytics** application folder **“C:\inetpub\wwwroot\ ReportingAPI\_POXAnalytics”**

**Step 4**: Open file **web.config** and replace the highlighted names with correct information. **Most of this information can be found in PO application web.config file.**

**dataSourceName**: Database source name to be inserted here

**Username**: SQL Username

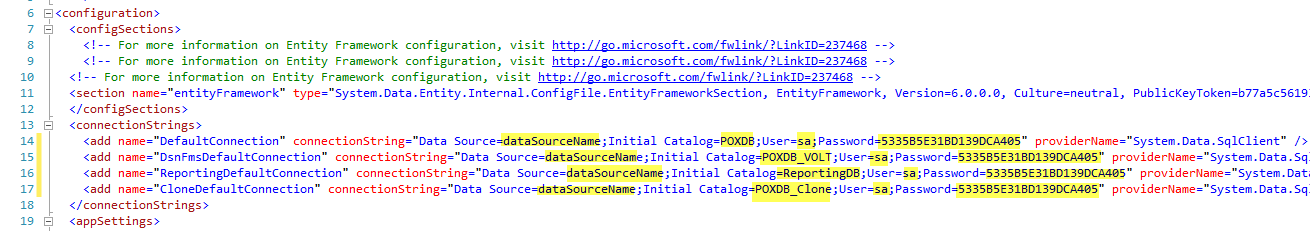
**Password**: Encrypted SQL user’s password

**Databases**: **POXDB** – POX database name to be mentioned here

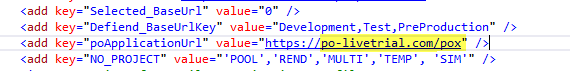
**POXDB\_VOLT** – VOLT database name to be mentioned here

**ReportingDB** – Reporting database name to be mentioned here

**POXDB\_Clone** – POX database’s copy will be mentioned here



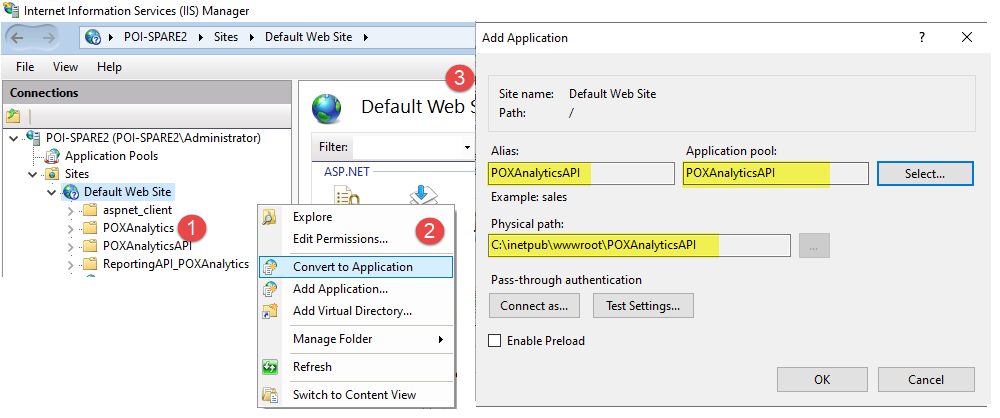
**Step 5**: In the same file, updated the value of poApplicationUrl with POX application link (highlighted below).



### POXAnalyticsAPI

**Step 1**: Go to IIS and right click on POXAnalyticsAPI.

**Step 2**: Select “**Convert to Application**” and select the highlighted values, and click **Ok**.



**Step 3**: Open the **POXAnalyticsAPI** application folder **“C:\inetpub\wwwroot\POXAnalyticsAPI”**

**Step 4**: Open file **appsettings.Development.json**

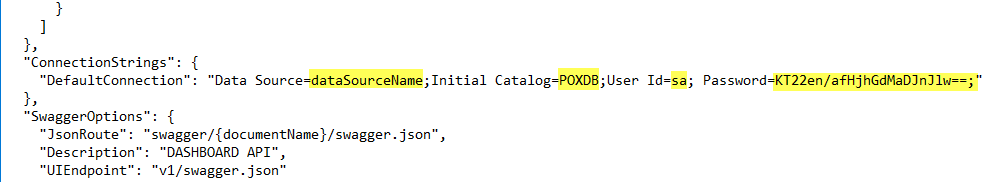
**Step 5:** Replace the highlighted field from below screen. You can take reference from POX application configuration file for the same.

**dataSourceName** – Database source name to be inserted here

**User Id**: SQL Username

**Password**: Encrypted SQL user’s password (to know how to generate the encrypted password in this step, [click here](#_Generate_Encrypted_Password))

**POXDB** – POX database name to be mentioned here



**Step 6**: Open file **appsettings.json**

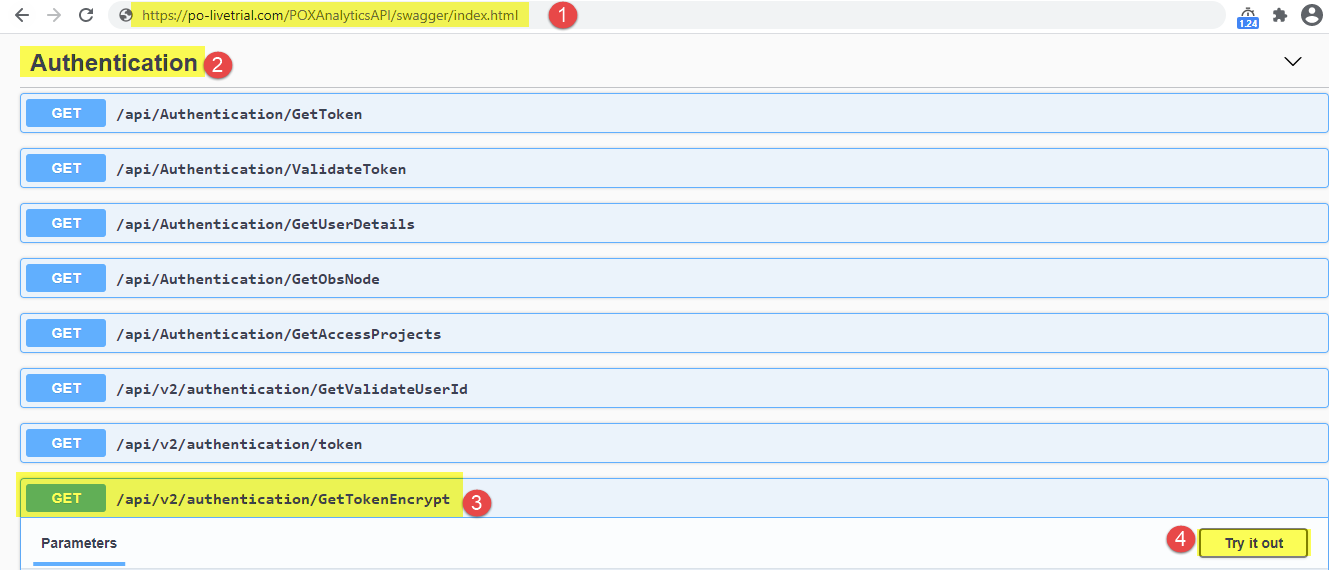
**Step 7:** Repeat Step 5 in this step.

### Generate Encrypted Password for POXAnalyticsAPI

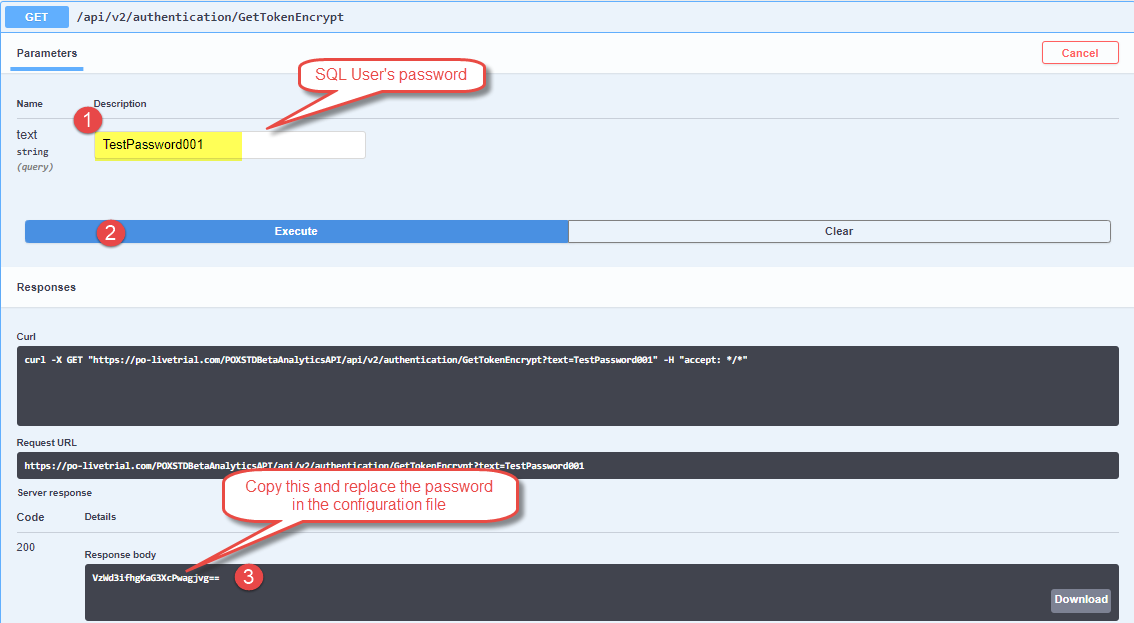
To generate the encrypted password, open the link for POXAnalyticsAPI. For instance,

<https://po-livetrial.com/POXAnalyticsAPI/swagger/index.html>

In the above link, the highlighted value can vary based on the URL of customer. Once link is opened, click on Authentication > then GetTokenEncrypt > and then Try it out (as highlighted below)



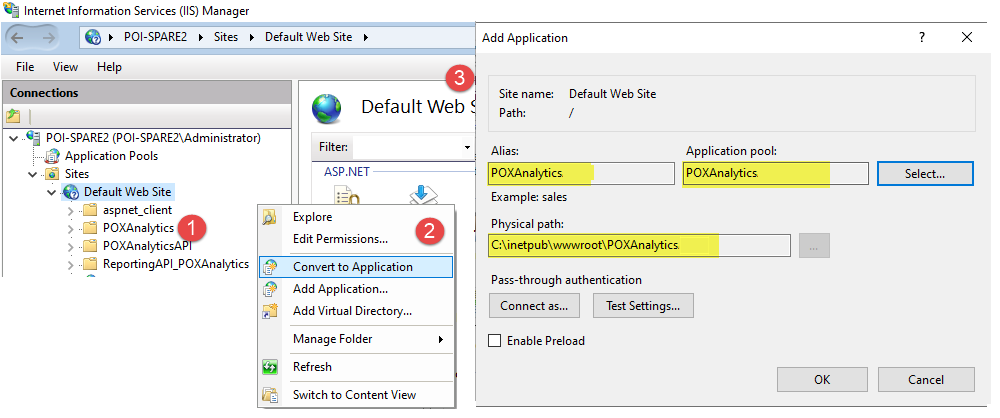
Now insert the SQL user’s password in the text field and click on Execute, it will provide the encrypted password. Copy the password and paste it in the configuration file.



### POXAnalytics

**Step 1**: Go to IIS and right click on POXAnalytics.

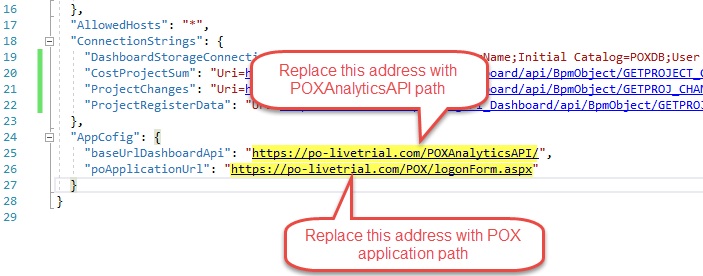
**Step 2**: Select “**Convert to Application**” and select the highlighted values, and click **Ok**.



**Step 3**: Open the **POXAnalytics** application folder **“C:\inetpub\wwwroot\POXAnalytics”**

**Step 4**: Open file **appsettings.Development.json**

**Step 5**: Replace the values highlighted below with correct links.



**Step 6**: Open file **appsettings.json**

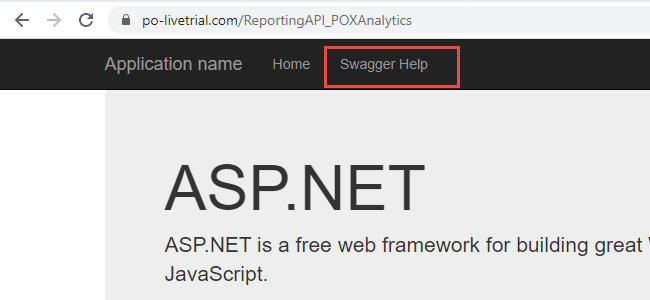
**Step 7:** Repeat step 5

**Step 8**: Open file **ApiDataSourceConfig.json**

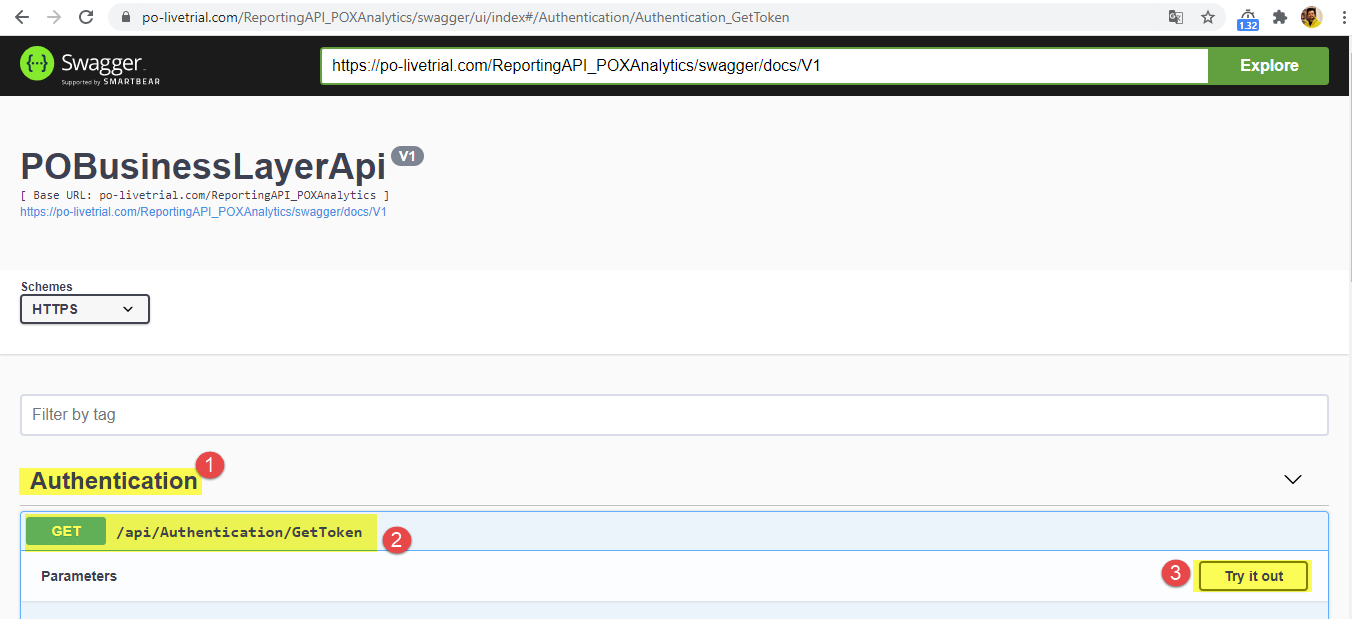
**Step 9**: In the link, replace the highlighted part of link with the reporting API link (which is configured on [**4.3.1**](#_ReportingAPI_POXAnalytics) in this document). Repeat this for all the links in this file.



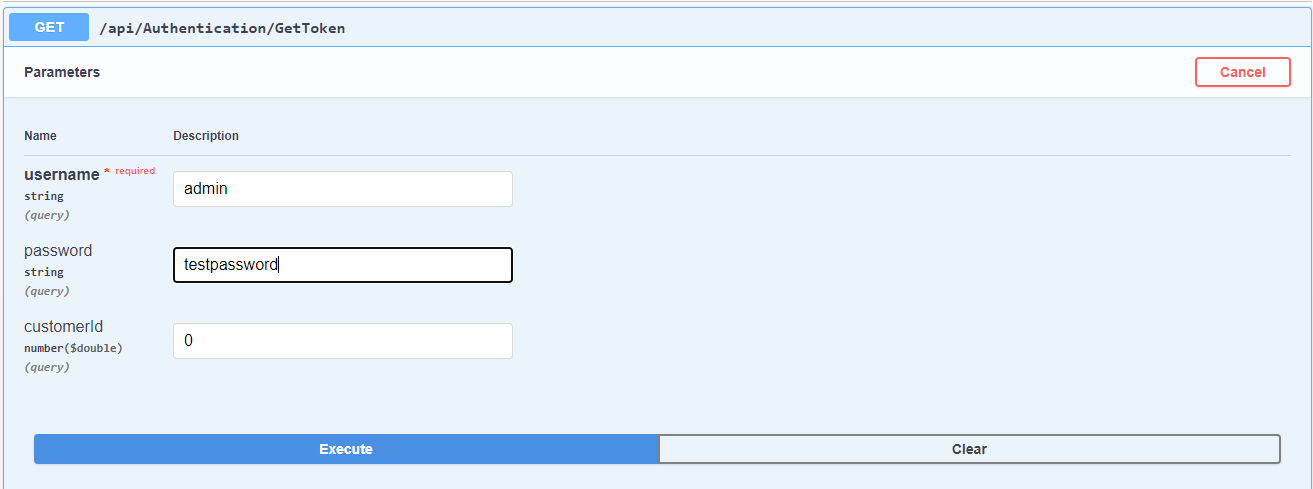
**Step 10**: Now you require to update the token (2nd point in the above screen). To generate the token, browse the reportingAPI in the browser and click on **Swagger Help**

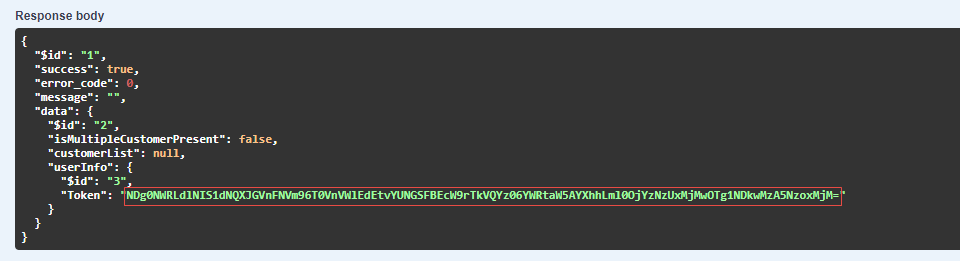


**Step 11**: Now click on **Authentication** > **GET** > **Try it out**



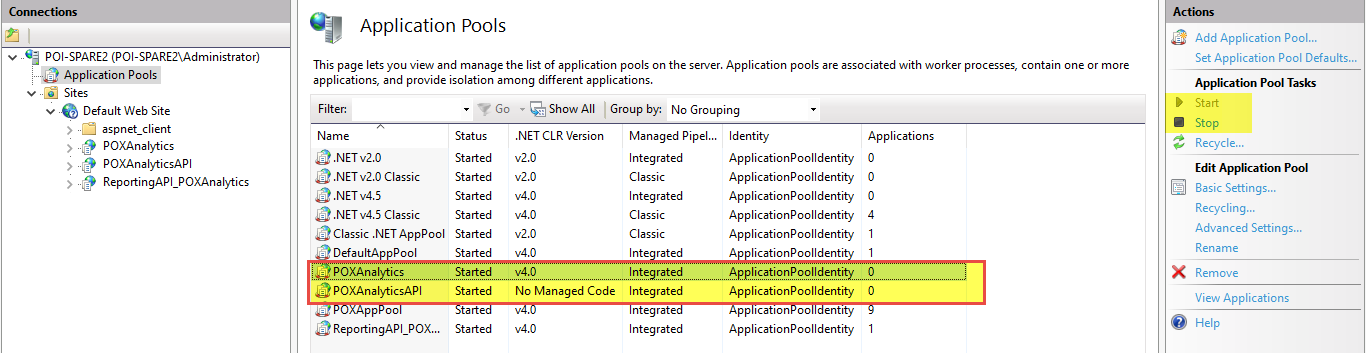
**Step 12**: Now insert valid username and password, and click on **Execute.** It will generate the token. Copy that token and replace it as mentioned in Step 9.





**Step 12**: Go to IIS > Application Pool > Select App pool “**POXAnalytics**” > Stop and Start

Go to IIS > Application Pool > Select App pool “**POXAnalyticsAPI**” > Stop and Start



## Redirection from PO Application

**Step 1**: Go to PO application folder > Environments > **Livetrial.config** file, and update the highlighed path with the valid path.

<add key="Path\_POAnalytics" value="https://po-livetrial.com/POXAnalytics/Home/PostLogon"/>

**Step 2:** Go to PO application folder > **web.config** file, and update the highlighted path with the valid path.

<add key="Path\_POAnalytics" value="https://po-livetrial.com/POXAnalytics/Home/PostLogon"/>

# Configure Publish Process

The auto-publish process publish the data from transaction database to reporting database. It runs based on the set frequency (for instance, once in a day) and publish the changed data into reporting database.

To configure the auto-publish process, PO will provide SQL Jobs. These jobs will execute based on set frequency and publish the data for us.