A picture containing text, clipart

Description automatically generated



**Deployment of SAAS-based application**

Document nr.:Deployment of SAAS-based application on Server

Revision: 1

Date: 24th Jan 24

**Author: Sunil Kumar**

Contents

[1. SPM PO application configuration. 3](#_Toc157008644)

[1.1 Deploy the PO application on the IIS 4](#_Toc157008645)

[2. TT application Configure 6](#_Toc157008646)

[2.1 Deploy the TT application on IIS. 6](#_Toc157008647)

[3. Configure the Business API. 8](#_Toc157008648)

[1.1 Deploy the Business API on IIS 8](#_Toc157008649)

[4. Analytics Configurations 10](#_Toc157008650)

[4.1 Analytics App Configuration 10](#_Toc157008651)

[4.1.1 Deploy Analytics App On IIS 11](#_Toc157008652)

[4.2 Configure the Analytics API 12](#_Toc157008653)

[4.2.1 Deploy the Analytics API on the IIS 13](#_Toc157008654)

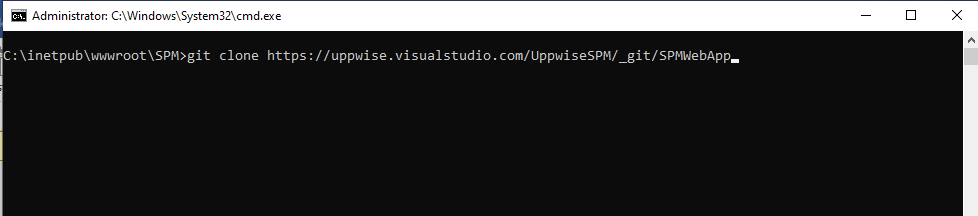
[4.3 Reporting API Configuration 14](#_Toc157008655)

[4.3.1 Deploy the Reporting API to IIS 14](#_Toc157008656)

First, we configure parameters on the application side to tailor it to specific needs. Then, we fine-tune settings on the database server side for seamless integration and optimal performance.

# 1. SPM PO application configuration.

**Step 1.** Establish a directory and navigate to the designated path using the terminal or manual means where the repository is to be cloned. Execute the following command after updating the repository URL. Alternatively, if opting for manual deployment, paste the ZIP file into the designated folder  
**eg.** Git clone <https://uppwise.visualstudio.com/UppwiseSPM/_git/SPMWebApp>



After cloning the application, it is crucial to update critical configurations such as the database connection settings, redirection paths, BusinessAPI URL, and Analytics URLs in both the Web.config file and the configuration files within the environment folder.

**Step 2.** Grant permissions to the root folder where both the TT and PO applications exist for everyone.

**Step 3.** Please provide the specific names of the folders and their corresponding paths that you would like to create as per the below details.

* Create a "temp" folder.
* Create a "temp" folder.
* Create a "JsonData" folder under the "sclicklib" folder.

**Step 4**. Update Database Connection String as per the requirement

<add key="DSN" value="Provider=SQLNCLI11;Data Source=**Sql server IP**;User ID=POIRE;Password=5335B5E31BD139DCA405;Initial Catalog=**Database Name**" />

**Step 5.** Activate SAAS functionality by setting its value to 1 in the relevant configuration

<add key="SAAS\_ENABLED" value="1" />

**Step 6.** Updated the TT application redirection Path remember we need to host the TT application with the same name.

<add key="PATH\_TT" value="**https://po-livetrial.com/SPMTT\_DEV/logonform.aspx**" />

**Step 7.** Updated the Business API and Authentication API URL

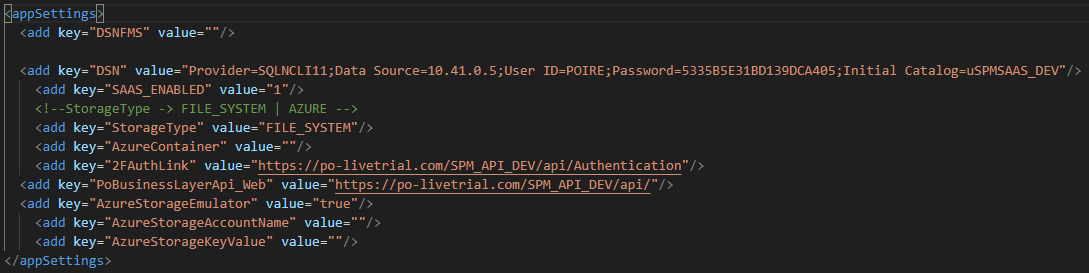
<add key="PoBusinessLayerApi\_Web" value="https://po-livetrial.com/SPM\_API\_DEV/api/" />

<add key="2FAuthLink" value="https://po-livetrial.com/SPM\_API\_DEV/api/Authentication" />

**Step 8.** Update the Analytics app URL and also, we need to host Analytics app same name as we added in this key.

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

**Step 9.** We need to make the same modification in the Environment folder file config file which will be on the web.config

**Eg.** If the path **<appSettings file=".\Environments\Development.config">** is mentioned in the PO application web.config file, the required modification should be made in the **Development.config** file located within the Environments folder. ****

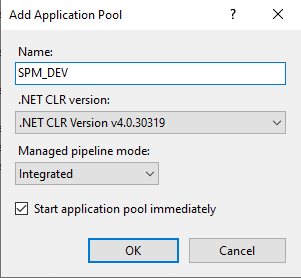
## 1.1 Deploy the PO application on the IIS

To deploy the PO app on IIS, the initial Step involves creating an application pool named, for instance, **PO application**. Ensure that the application pool is configured with the **Requirement specifications** and a managed pipeline mode set to **Integrated/ Classic** as per requirement.

**Note:** Before moving forward with IIS deployment, make sure to download and install the SDK and hosting bundle versions as per the specified requirements. This step ensures that the application aligns seamlessly with the necessary runtime environment.

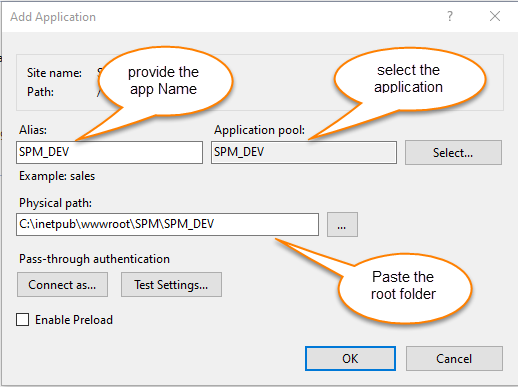
**Step 1.** Create the application pool with the instance name by using the following instructions.

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2.** To host an application on IIS, follow these general Step s:

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.
* Click on OK.



# 2. TT application Configure

To deploy the TT application, we need to apply the same modifications as those performed in the main application, such as updating files like Web.config and the environment folder.

**Step 1.** Navigate to the TT folder, whether cloned or manually pasted. Create a "**Temp**" folder at the root directory of the TT application and a "**JsonData**" folder under "**Slicklib**" similar to the configuration for the PO application.

**Step 2.** Open the web.config file and proceed to update the following critical configurations such as the Database name, Server IP, encrypted password, SAAS enable etc.

**Step 2.1** Activate SAAS functionality by setting its value to 1 in the relevant configuration

<add key="SAAS\_ENABLED" value="1" />

**Step 2.2.** Add the PO application URL to redirect from the TT application to the PO application.

<add key="PATH\_PDE" value="https://po-livetrial.com/SPM\_DEV/logonform.aspx"/>

<add key="PATH\_TS" value="https://po-livetrial.com/SPM\_DEV"/>

**Step 3.** Navigate to the Environment folder and open the configuration file corresponding to the web.config. Proceed to update the database connection string within that configuration file.

<add key="DSN" value="Provider=SQLNCLI11;Data Source=Server Ip;User ID=POIRE;Password=5335B5E31BD139DCA405;Initial Catalog=database name"/>

Eg. Database name, server Ip, encrypted pass etc.

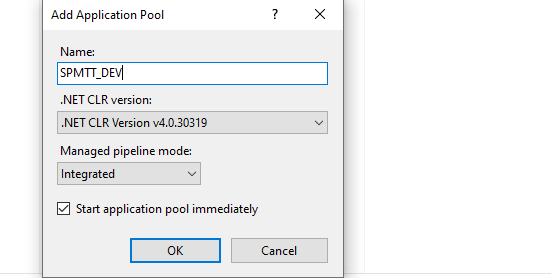
## 2.1 Deploy the TT application on IIS.

To deploy the TT application, we need to follow the same approach as the PO application.

**Note:** Before moving forward with IIS deployment, make sure to download and install the SDK and hosting bundle versions as per the specified requirements. This step ensures that the application aligns seamlessly with the necessary runtime environment.

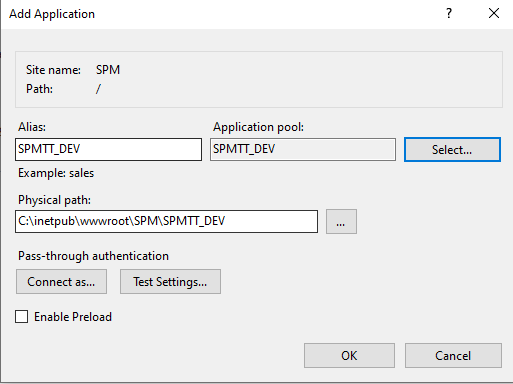
**Step 1.** To Create the application pool, use the following the below instructions

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2.** To deploy the TT application on IIS, follow the Step s outlined below:

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.
* Click on OK.



# 3. Configure the Business API.

To deploy the business API of the application, the following Steps are required:

**Step 1.** Create a new folder with the API name and Grant permissions to the root folder where both the TT and PO applications exist for everyone.

**Step 2**. Publish the business API code using visual studio to a specific path.

**Step 3**. Paste the published files to the newly created folder.

**Step 4.** Open the web.config file of Business APItoreflect the necessary configurations to it.

**Step 4.1.** Modify the **database connection string** as indicated in the following code snippet.

<add name="DefaultConnection" connectionString="Data Source= **server IP**;Initial Catalog=**SAAS database**;User=POIRE;Password=5335B5E31BD139DCA405" providerName="System.Data.SqlClient"/>

<add name="DsnFmsDefaultConnection" connectionString="Data Source=**server IP**;Initial Catalog=**Volt\_database** name;User=POIRE;Password=5335B5E31BD139DCA405" providerName="System.Data.SqlClient"/>

**Step 4.2** Activate SAAS functionality by setting its value to 1 in the relevant configuration

<add key="SAAS\_ENABLED" value="1"/>

**Step 4.3** Include the URL of the PO application within the configuration to facilitate redirection functionalities.

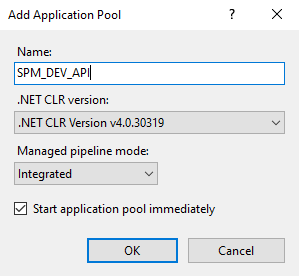
<add key="RedirectUrl" value="https://po-livetrial.com/SPM\_DEV/TeamsAuth.aspx"/>

## Deploy the Business API on IIS

To deploy Business API on IIS, the initial step involves creating an application pool named, for instance, **SPM\_DEV\_API**. Ensure that the application pool is configured with the **.Net CLR version 4.0** or depends on the requirements and a managed pipeline mode set to **Integrated/Classic**.

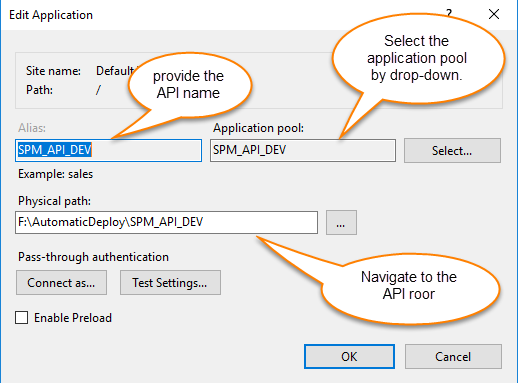
**Step 1.** Create the application pool using the following application

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2**. To deploy the TT application on IIS, follow the Step s outlined below:

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.
* Click on OK.



# 4. Analytics Configurations

To facilitate the configuration of Analytics for the specific application, it is imperative to obtain and deploy the following published files:

1. Analytics App Publish file
2. Analytics API publish files
3. Reporting API published files.

## 4.1 Analytics App Configuration

To effectuate the deployment of the Analytics App, it is imperative to procure the published files encompassing the analytics code, the requisite SDK version, and the Runtime Hosting Bundle. Additionally, create an application pool configured with .NET CLR version 4.0 and an Integrated managed pool.

**Step 1.** Create a folder called "**Analytics**," and allow everyone to access it, and then inside "**Analytics,"** create another folder named "**Analytics App.**"

**Step 2.** Published the analytics App code using the visual studio to the specific path.

**Step 3.** Copy the published files from the designated path and meticulously transfer them to the "**Analytics App**" folder for seamless integration.

**Step 5.** Open the App "**appsettings.json**" file and Update the Analytics API URL.

"SecurityCheckUrl": “Analytics API URL"

**Step 5.1** Additionally, it is imperative to update the **BaseDashboardAPI** URL and the **PO** application URL in the "**appsettings.json**" file, the corresponding code snippet is provided below

"baseUrlDashboardApi": "dashboard API url",

"poApplicationUrl": "PO application URL"

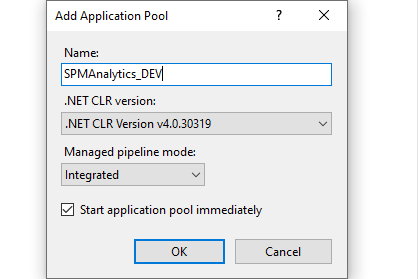
**Step 5.2** Duplicate the code from the "**appsettings.json**" file and paste it into the "**appsettings.development.json**" file. Ensure to save the changes afterwards.

### 4.1.1 Deploy Analytics App On IIS

Before moving forward with IIS deployment, make sure to download and install the SDK and hosting bundle versions as per the specified requirements. This step ensures that the application aligns seamlessly with the necessary runtime environment.

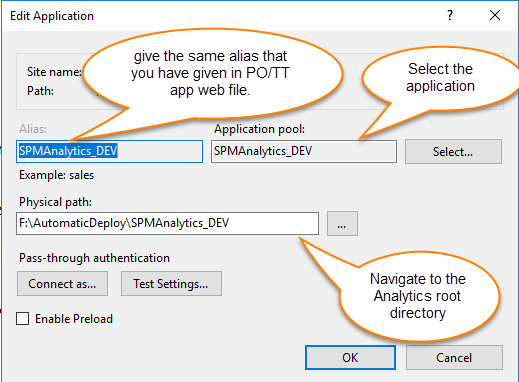
**Step 1.** Create an application pool by following the instructions.

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2.** Host the Analytics app on the IIS by following outlined instruction.

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.
* Click on OK and Deploy the Analytics app.



## Configure the Analytics API

**Step 1.** Navigate to the Analytics directory and create a new folder with the name "**Analytics API**".

**Step 2.** After cloning the Analytics API code, professionally publish it utilizing Visual Studio to a designated folder.

**Step 3.** Copy the published files and paste them to the newly created folder "**Analytics API**".

**Step 4.** Open the "**Appsettings.json**" file and meticulously update the database connection string by the provided code snippet.

"DefaultConnection": "Data Source=Server IP;Initial Catalog=Sasa database;User Id= Username; Password=KT22en/afHjhGdMaDJnJlw==; Encrypt=False;"

**Step 4.1** Activate SAAS functionality by setting its value to **1** as indicated in the provided code.

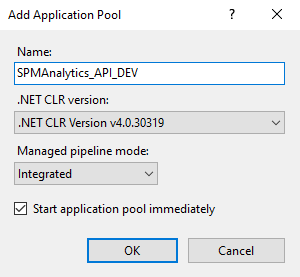
"SAAS\_ENABLED": "1"

**Step 4.2** Duplicate the code from the "**appsettings.json**" file and paste it into the "**appsettings.development.json**" file. Ensure to save the changes afterwards.

### 4.2.1 Deploy the Analytics API on the IIS

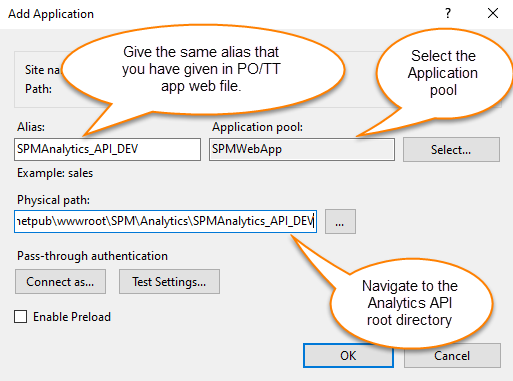
**Step 1.** Create an application pool by following the instructions.

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2.** Host the Analytics app on the IIS by following outlined instruction.

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.



## 4.3 Reporting API configuration

To set up the Reporting API, you must modify specific parameters in its "**web.config**" file. This entails updating details such as the **database** connection string, **PO application** path, and enabling **SAAS** functionality, ensuring the proper configuration for optimal functionality.

**Step 1.** Navigate to the Analytics directory and create a new folder with the name "**Reporting API**".

**Step 2.** After cloning the Analytics API code,

**Step 3**. Professionally publish the code utilizing Visual Studio to a designated folder.

**Step 4.** Copy the published files and paste them to the newly created folder "**Reporting API**".

**Step 5.** Open the "**web.config**" file and meticulously update the database connection string by the provided code snippet.

<add name="DefaultConnection" connectionString="Data Source=Server IP;Initial Catalog=Database Name;User=POIRE;Password=5335B5E31BD139DCA405" providerName="System.Data.SqlClient" />

**Step 4.1** Add the PO application URL in the web file seen in the below code.

<add key="poApplicationUrl" value="https://po-livetrial.com/SPM\_DEV" />

**Step 4.2** Activate SAAS functionality by setting its value to **1** as indicated in the provided code.

<add key="SAAS\_ENABLED" value="1" />

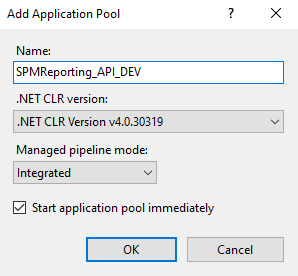
## 4.3.1 Deploy the Reporting API to IIS

To deploy the Reporting API on IIS, it is imperative to adhere to the following steps sequentially.

**Note:** Before moving forward with IIS deployment, make sure to download and install the SDK and hosting bundle versions as per the specified requirements. This step ensures that the application aligns seamlessly with the necessary runtime environment.

**Step 1.** Create an application pool by following the instructions.

* Open IIS Manager.
* Navigate to "Application Pools."
* Click "Add Application Pool..."
* Provide a name, select .NET CLR Version (e.g., v4.0), and choose "Integrated" for the managed pipeline mode.
* Click OK to create the application pool.



**Step 2.** Host the Analytics app on the IIS by following outlined instruction.

* Open IIS Manager.
* Right-click "Sites" and choose "Add Website..."
* Fill in site details, including the Physical Path pointing to your extracted code folder.
* Click OK to create the site.
* In the Actions pane, click "Basic Settings" for the new site.
* Choose the application pool.

