



Sync Installation Guide

Contents

Contents.....	2
Preface	3
Prerequisites	3
Installation	4
Prepare File System.....	4
Setup SQL Server.....	4
Setup IIS web server.....	4
Creating the Sync Application Pool	4
Add the Sync Web Application.....	5
Configuration	7
Web.config.....	7
Application Host Config.....	8
Securing your Platform.Sync site.....	9
Synchronizing Form Entries	9
Option 1: Push Form Entries to Sync (recommended)	10
Option 2: Poll Form Entries from Platform	10
Synchronizing Data Sources	11
App GET rows from Sync tables (recommended).....	11
Filtering Rows By User	12
Filtering Rows By Timestamp (Incremental Row Updates)	12

Preface

This guide assumes you have a working knowledge of the following:

- Forms On Fire Platform
- IIS web server
- SQL Server / SQL Server Management Studio

This guide contains the information you need to understand and use the Sync application to perform the following tasks:

- Synchronize Form Entries into your database on a polling schedule
- Push Form Entry submissions directly into your database
- Synchronize Data Sources on your mobile client directly from your database
- Push data into your Platform Data Sources

Prerequisites

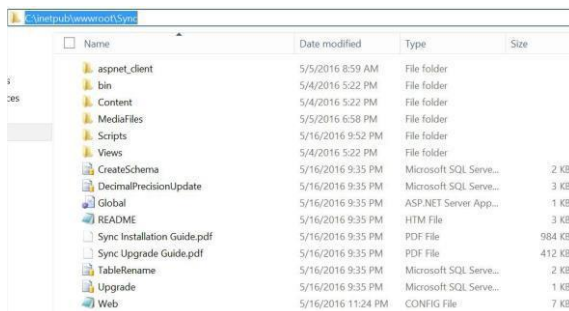
This guide assumes you have the following already setup/configured:

- A server machine running Windows Server 2008 or higher
- SQL Server 2008 R2 or higher
- .NET Framework 4.5.2 or higher
- IIS 7.0 or higher
- The IIS server must be Internet accessible via web domain or IP address on port 80 (HTTP) or port 443 (HTTPS).
If using HTTPS - which is recommended – then an appropriate SSL certificate must be loaded on the IIS server, with the certificate provided by a recognized root Certificate Authority. Internally generated SSL certificates are NOT supported.
- A Windows account for use by the Sync application software.
This user account must have administrator rights on the IIS web server machine
- Enabled ASP.NET 4.5 development features

Installation

Prepare File System

1. Log on to the web server computer as an administrator, preferably using the Sync windows account you should already have created.
2. Create a root directory into which you will install the various aspects of the Sync software.
Ensure that the Sync windows user account has Full Control security rights on this folder and all subfolders.
3. Inside this root directory, create a subdirectory by unzipping the Platform.Sync zip archive. Make a note of the Platform.Sync folder path - we will refer to this later on.
e.g. C:\inetpub\wwwroot\Sync



4. Create another subdirectory inside the root directory where you would like media files to be downloaded to. Note this folder path - we will refer to this later on.
e.g. C:\inetpub\wwwroot\Sync\MediaFiles

Setup SQL Server

1. Click **Start**, and type **SQL Server Management Studio**. Select it from the search results and open the program.
2. Log into your chosen SQL Server as an administrator.
3. Create two new databases in your SQL Server – name these **SyncControl** and **SyncData**.
4. Connect to the SyncControl database and open the CreateSchema.sql file found in the Sync directory.
5. Run the CreateSchema.sql file and check that tables have been created in the SyncControl database.
6. Ensure that you have granted the designated Sync user full admin access to the two databases created above.
7. Log out of SQL Server and make sure you can successfully login as the Sync user.

Setup IIS web server

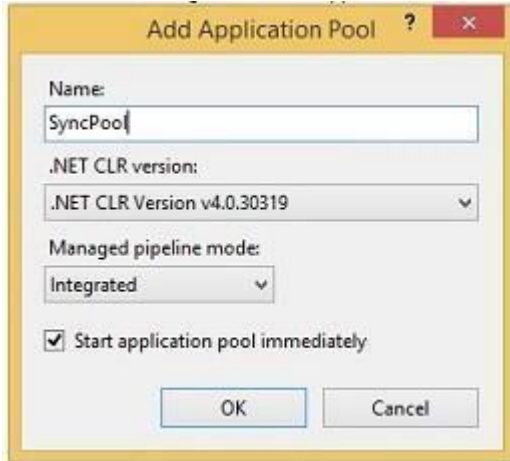
1. Click **Start**, and type **Internet Services Manager**. Select it from the search results and open the program.

Creating the Sync Application Pool

2. Right-click **Application Pools** from the left hand side menu and click **Add Application Pool...**

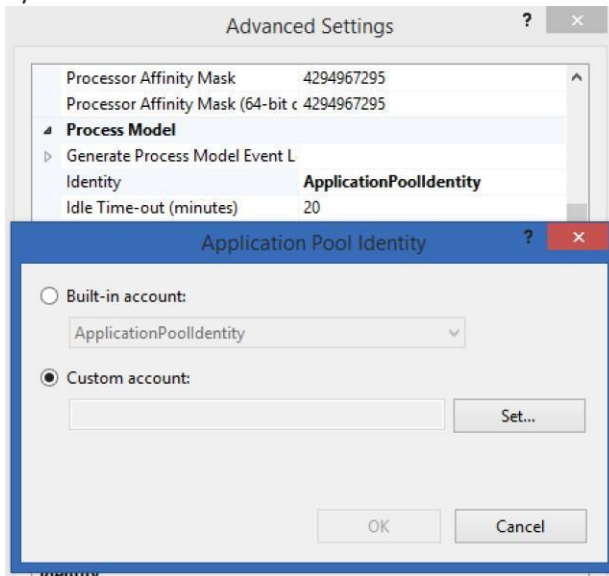


3. Enter the **Application Pool** info as per the screenshot below and click **OK**:



4. Right click on the newly created SyncPool and choose Advanced Settings
5. In the Advanced Settings dialog, click on the Identity field and change this to use a custom account.

Set this to be the designated Windows account that you previously created for the Sync software.



Add the Sync Web Application

1. Right-click on **Sites** and click **Add Website...**



6. Enter the **Website** info as per the screenshot below and click

The screenshot shows the 'Add Website' dialog box with the following configuration:

- Site name:** Sync
- Application pool:** SyncPool
- Content Directory:** Physical path: C:\deploy\sync
- Binding:** Type: http, IP address: (empty), Port: 80, Host name: platform.domain.com
- Start Website immediately

- **Application Pool** must be set to the SyncPool you created earlier
 - **Content Directory**
 1. **Physical Path** must refer to the Sync directory created earlier.
 - **Binding**
 1. **Type**
Select HTTP if you want the Web site to have an HTTP binding, or select HTTPS if you want the Web site to have a Secure Sockets Layer (SSL) binding.
 2. **IP Address**
Enter an IP address that users can use to access this site.
Select **All Unassigned** if you wish this to respond to requests for all IP addresses on the port and optional host name that you specify.
 3. **Port**
If you select HTTP from the **Type** drop-down list, the default port is 80; if you select HTTPS from the drop-down list, the default port is 443. If you specify a port different from the default ports, clients must specify the port number in requests to the server.
 4. **Host name**
Specify a domain name instead of an IP Address that clients must use to access the Sync website. e.g. sync.mydomain.com
You must ensure the relevant DNS routing is in place to direct host name requests to this web server.
2. Now click on Authentication. For Anonymous Authentication, make sure to select "Application Pool Identity".



Authentication

Group by: No Grouping

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Forms Authentication	Enabled	HTTP 302 Login/Redirect

Edit Anonymous Authentication Credentials

Anonymous user identity:

Specific user:

Application pool identity

OK Cancel

3. Now that the website has been created successfully in IIS, we can begin the Sync specific configuration.

Configuration *Web.config*

1. Navigate to the Sync folder created earlier and open the **Web.config** file in your favorite text editor (e.g. Notepad or Notepad++).
2. Navigate to the **<connectionStrings>** element:

```
<connectionStrings>
  <add name="SyncControl"
    connectionString="Server=yourServer;Database=syncontrol;User ID=yourId;Password=yourPassword;Trusted_Connection=False;Encrypt=True;MultipleActiveResultSets=False"
    providerName="System.Data.SqlClient"/>
  <add name="SyncData"
    connectionString="Server=yourServer;Database=syncdata;User ID=yourId;Password=yourPassword;Trusted_Connection=False;Encrypt=True;MultipleActiveResultSets=False"
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

1. SyncControl

This database contains the various configuration tables used by the Sync application.

2. SyncData

This connection refers to the database that will house the actual Data Source data and Form Entry submissions.

3. For each connection string replace the following:
 1. **yourServer** with the name or network address of the instance of SQL Server to connect e.g. localhost\sqlexpress
 2. **Database=syncontrol/syncdata**, with the name of the actual database to connect to. We recommend these be unchanged if you are able to create the databases named as such.
 3. **yourId** with the SQL Server login account.
 4. **yourPassword** with the password for the SQL Server account logging on.
3. Navigate to the **<appSettings>** element:
 1. For the purpose of this guide we are only interested in the **Sync.*** keys
 1. Basic
 1. **Sync.ProviderId**
enter your company's unique identifier from our cloud platform found here: <https://secure.formsonfire.com/Provider>
 2. **Sync.IntegrationKey**
enter your company's unique integration key from our platform found here: <https://secure.formsonfire.com/Provider>
 3. **Sync.Url**
enter your platform's root url – e.g. <https://secure.formsonfire.com>
 4. **Sync.BillingURL**
enter the Sync Billing Url provided to you by our Sales team. If none was provided, leave as default.
 5. **Sync.SelfUrl**
enter the Internet accessible url by which the sync site can be reached – e.g. <https://sync.yourdomain.com>
This would be the host name you have configured in IIS for the sync site above.
 2. Polling
 1. **Sync.Poll** to enable form entry polling set this key's value to true, else false if you do not wish to enable polling.
We recommend against using polling – rather use REST connectors to push form entry data to the Sync installation for far more real time data access. Additionally, polling will consume your daily API usage allocation on our platform, whereas REST connector pushes do not.

2. **Sync.Poll.Minutes** if polling is enabled you need to specify the timeframe on which to schedule the actual polling. Enter this value in minutes e.g. 15.
 3. **Sync.Form.ExternalIds** this field accepts a comma separated list of form screen ExternalId's. Any new form submissions against these forms will be downloaded and stored in the SyncData database.
3. Media
 1. **Sync.Download.Media** set this value to true to automatically download any related media files when a new form submission is retrieved either via polling or accepted POST request.
 2. **Sync.Download.Media.Path** this key specifies the path to which to download any medial files. Set this to be the path of the files subdirectory created earlier – e.g. D:\Sync\MediaFiles
 4. General Config
 1. **Sync.ServerId** insert your customer key here provided to you by our sales team.
 2. **Sync.LogPath** this key specifies the path in which log files will be stored. Please note that these logs will only be enabled remotely under certain circumstances and are for use specifically by our development team.

Application Host Config

NB: ENSURE THESE CHANGES ARE SAVED ONLY AFTER ALL FOLLOWING MODIFICATIONS HAVE BEEN MADE. Be sure to save a copy of the current Application Host file for backup.

The Sync application must be always running to guarantee always-on background job processing. In order to achieve this, you need to edit the global **applicationHost.config** file found at:

%WINDIR%\System32\inetsrv\config\applicationHost.config

1. First, you need to add/edit the **applicationPools** tag to include AlwaysRunning **startMode**.

```
<applicationPools>
  <add name="SyncPool" managedRuntimeVersion="v4.0" startMode="AlwaysRunning"
/> </applicationPools>
```

2. Second, add/edit the Sync **site** tag to include these **serviceAutoStart** properties:

```
<sites>
  <site name="Sync" id="1">
    <application path="/"
      serviceAutoStartEnabled="true"
      serviceAutoStartProvider="ApplicationPreload" />
  </site>
</sites>
```

3. Next add/edit a **ServiceAutoStartProviders** entry for the Platform.Sync web application. Note this must appear just AFTER closing the </sites> AND AFTER any <webLimits> tag.

```
<serviceAutoStartProviders>
  <add name="ApplicationPreload"
type="Platform.Sync.ApplicationPreload, Platform.Sync" />
```

```
</serviceAutoStartProviders>
```

Securing your Sync site

In order to secure your Sync site you must specify a username and password as well as the password hash function which will ensure that your Sync's public domain requires authentication.

1. Navigate to the Sync folder created earlier and open the Web.config file in your favorite text editor (e.g. Notepad or Notepad++).
2. Under the <authentication element you will see a <credentials element which allows you to specify the username and password as well as a passwordFormat which can be either "Plain", "MD5" or "SHA1".
3. The following credentials shows a username of "Sync" and an MD5 password hash value of "5yncU53r"

```
<credentials passwordFormat="MD5">
  <user name="Sync" password="1ed6699406c8853ef525826ea5dbe38f"/>
</credentials>
```

The Sync installation is now complete, to verify this open a browser and navigate to the IP address or domain name and port specified in the Binding section above.

You should see a Sync welcome page.

Synchronizing Form Entries

The Sync application allows each Form Entry to be saved into a table in the SyncData database, within a table named by the Form screen's External Id, and each column is being the data name of the answered fields.

Along with the form entry data the following metadata is also saved in columns:

- Id – form entry unique identifier.
- ItemNo – the row number within this form entry incremented for each repeated group answer.
- FormVersion – The form screens version
- Latitude – Where the form entry was saved
- Longitude – Where the form entry was saved
- StartTime – Datetime when the form entry was started
- ReceivedTime – Datetime when the form entry was initially received
- CompleteTime – Datetime when the form entry was completely received
- UserFirstName – First name of the mobile app user
- UserLastName – Last name of the mobile app user

Media files will be downloaded and saved to the MediaFiles folder you set up earlier, assuming Sync.Download.Media is set to true.

Option 1: Push Form Entries to Sync (recommended)

Within the Sync application, a form entry REST api is exposed with a single on POST function which accepts form entries and saves them into the SyncData database.

This api is accessible on the configured <https://{IP Address/Domain Name}/api/v1/formentries>.

In order to take advantage of this and have the Platform automatically POST each form entry received to your Platform.Sync project, you must create a REST connector within the Platform for each form screen as follows:

- a. Login to the Platform
- b. Navigate to App Builder > Screens and click the settings link on the specific form screen.
- c. Ensure that you have set an **External Id** value – this will be the name of the table created in the SyncData database.
- d. Next go to the Connectors page of your Form screen.
Mouse over Add Connector and select REST

The screenshot shows a configuration form for a REST connector. On the left is a globe icon. The form includes:

- A dropdown menu set to 'POST'.
- A 'Data as' dropdown menu set to 'JSON'.
- A 'Destination URL' text input field containing 'http://sync.mydomain.com/api/v1/formen'.
- A 'Use Global Forms REST URL' checkbox, which is currently unchecked.
- An 'Entry Format' dropdown menu set to 'Rich'.
- Two links on the right: 'customise headers' and 'customise body'.
- A link 'add run condition' is visible below the 'POST' dropdown.

- e. Enter the connector details as per the screenshot (using your own Sync installation's formentries url) and click Save.
- f. Be sure to choose "Rich" for the Entry Format.

When a Form entry is submitted from the mobile app the Platform will automatically post the JSON data to the Destination URL specified in the REST connector.

On receipt, Sync will automatically create the required Form entry table (if needed) and will save the form entry. If the **Sync.Download.Media** web.config setting is true, then Sync will automatically download any related media.

Note:

If you are setting up many Form screens with REST connectors, then you may want to set the Global Forms REST URL found in the Menu -> Organization Setup page.

Set this to be the Sync API endpoint and then on all REST connectors, instead of filling in the "Destination URL" rather just tick the "Use Global Forms REST URL" option.

Option 2: Poll Form Entries from Platform

In order to synchronize form entry submissions from the Platform into your own database, we need to specify the forms we wish to poll for.

This is done by setting the **Sync.Form.ExternalIds** to a comma separated list of Form screen External Id's. You can also specify the poll interval in **Sync.Poll.Minutes** the platform will request new form entries for each external id set from the Platform.

Synchronizing Data Sources

App GET rows from Sync tables (recommended)

If you have data source rows that you wish to make continuously available to the app, the best way is to make use of Sync's GET support.

You need to create your desired data source tables in the SyncData database, naming the tables by the External Id you have assigned to the matching Data Source in our platform.

Once set up, the app will directly request rows from the SyncData database table, bypassing the need to have rows on our platform altogether.

Within the Sync application, a data source REST API is exposed with a single on GET function which accepts a data source request and generates a response containing the specified SyncData database tables, where external id is the table name.

This API is accessible on the configured `https://{IP Address/Domain Name}/api/v1/datasources`.

In order to take advantage of this, add a Hosted GET connector within the Platform for each data source as follows:

1. Login to the Platform
2. Navigate to App Builder > Data Sources and click the settings link on the specific data source.
3. Ensure that you have set the **External Id** field with a value – this **MUST** match the name of the table/view you've already created in the **SyncData** database.
4. Click "Save".
5. **Important:** From the Rows page, manually add the columns that match the columns in your SQL table/view. Columns must be in the same order as they appear in SQL. Click Save.
6. Navigate back to the data source settings page. Mouse over **Add Connector** and select Hosted GET



7. Enter the connector details as per the screenshot (enter your own Sync installation's datasources target url) and click **Save**.

If you are setting up many data sources with Hosted GET connectors, then you may want to set the Global Hosted GET URL found in the Menu -> Organization Setup page.

Set this to be the Sync API endpoint and then on all Hosted GET connectors, instead of filling in the "Target GET URL" rather just to the "Use Global Hosted GET URL" option.

IMPORTANT NOTES:

- Each platform Data Source External Id must match that of the SyncData table's name.
- Columns in the data source must follow the same order as the columns in the related table.

Filtering Rows By User

By default, all rows in each Sync data source table is returned to the app. However, you can choose to filter the rows sent to the app by user if you wish.

To enable, add the following column onto your SyncData tables (column name must match exactly and should be added to the end of the table):

1. ***_userIds.***

This column must contain a comma separated list of Platform user emails when a row is only applicable to those users. When data source data is requested from the mobile app this column (if it exists) will be compared against the requesting mobile user's email address and only records containing their email will be returned.

Filtering Rows By Timestamp (Incremental Row Updates)

If your data set is very large, then instead of having the app download all rows on each synchronization, you can implement incremental row updates based on a timestamp and deletion status. This will result in only the changed/deleted rows being downloaded by the app after its first initial synchronization.

To enable, add the following columns onto your SyncData tables (column names must match exactly and should be added to the end of the table):

1. ***_lastUpdated.***

This column contains the date time value that the record was last updated and is used as a check when retrieving data to prevent retrieving old rows.

2. ***_deleted.***

Boolean column that represents whether or not this row has been deleted.