# Development Environment Setup

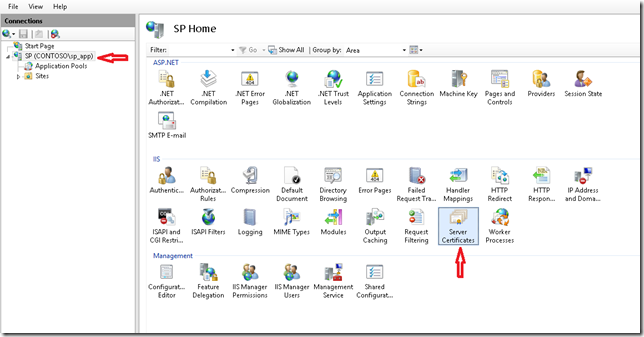
# Create a new SharePoint team site

1. Create a new managed path for PIW systems: /piw: explicit inclusion
2. Create new content database name D2\_Content\_PIW
3. Execute this scripts to create new site collection:



# Create S2S certificate

Create a self-signed certificate from the server that is going to host the remote web



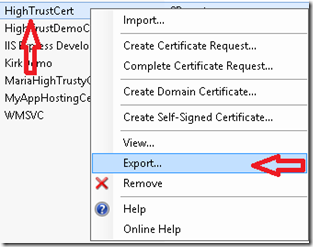
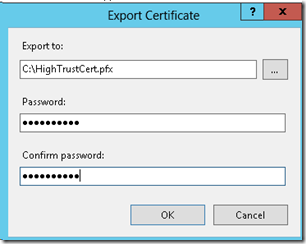
2. Click on Create Self Signed Certificate

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/1351.image_5F00_3A2B60A9.png)

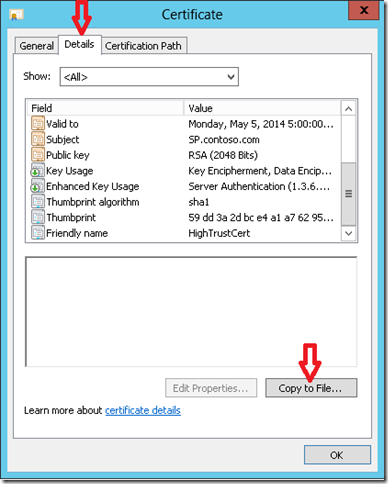
3. Enter some meaningful name like **S2SDEVRev2**and Click on **Ok.**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/7608.image_5F00_692D9C80.png)

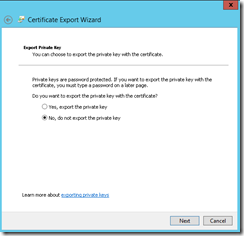
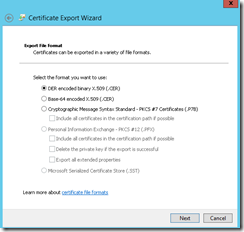
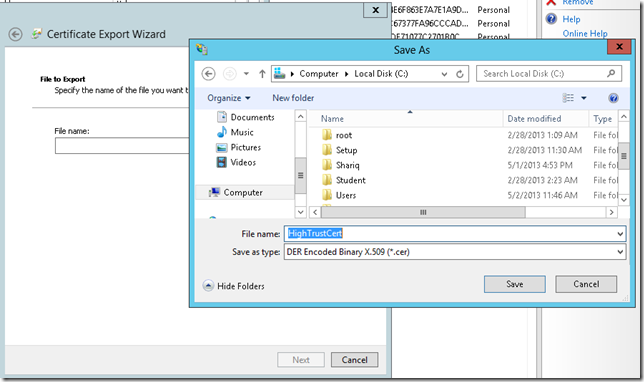
4. Now we need to export the **personal exchange format (.pfx)**file. Right Click on the Certificate in IIS click on **Export**and provide an accessible location. Also enter the password that you want to use and Click on **Ok**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/2133.image_5F00_56A9DF43.png) [](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/0844.image_5F00_2C7A315B.png)

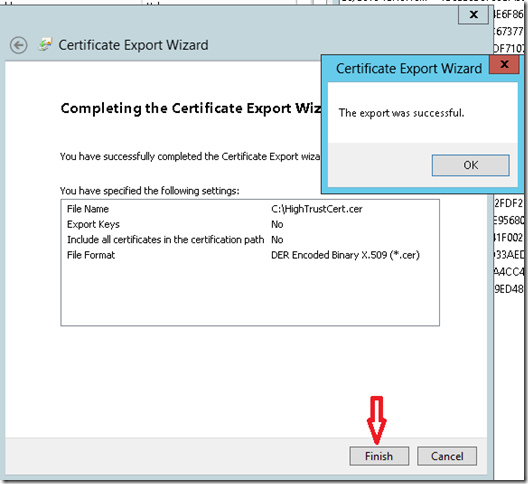
5. Next, double click on the Certificate in IIS. Click on **Details**tab and click on **Copy to File.**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/4786.image_5F00_6692EC38.png)

6. Now you should see the **Certificate Export Wizard** (remember earlier we exported the .pfx file). The first screen elucidate the significance of what we are doing. Keep Clicking **Next**across the three screens. Below screenshots demonstrate the same. I keep all the default options. Just one thing to mark that now we are now exporting the **.cer**file. I choose the same location. Click on**Save.**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/5811.image_5F00_056F641A.png)[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/3833.image_5F00_6D887CF3.png)[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/3312.image_5F00_6A931840.png)[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/0116.image_5F00_4F896943.png)

and finally, click on **Finish.**You should see you the message “**The export was successful**”.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/3225.image_5F00_7DB33F30.png)

To verify, go to your **Trusted Root Certification Authorities Store** and you should see your Certificate there. If you don’t, or if you re-use the certificate from other computer, you need to add the certificate. https://technet.microsoft.com/en-us/library/cc754841.aspx

# Run Windows PowerShell cmdlets to set up trusted security token service

Run SharePoint 2013 Management Shell as administrator. First thing first, you need an Issuer ID. Use the below PowerShell cmdlets to get one. Remember to make note of this, as we will use it later. An important point, it has to be Lowercase only.



Check if the SecurityTokenIssuer is not expired



The next step is register clientID. It will be helpful so we can deploy the app the local IIS instead of using IIS Express and F5 for debug (slow) from Visual Studio

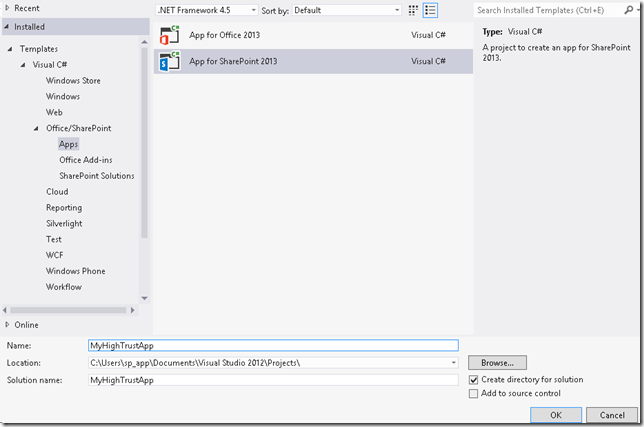
Go to site: <https://fdc1s-sp23wfed2.ferc.gov/piw/_layouts/15/appregnew.aspx>

Click on Generate Client ID. Copy the GUID to below script (or use powerhell as above)

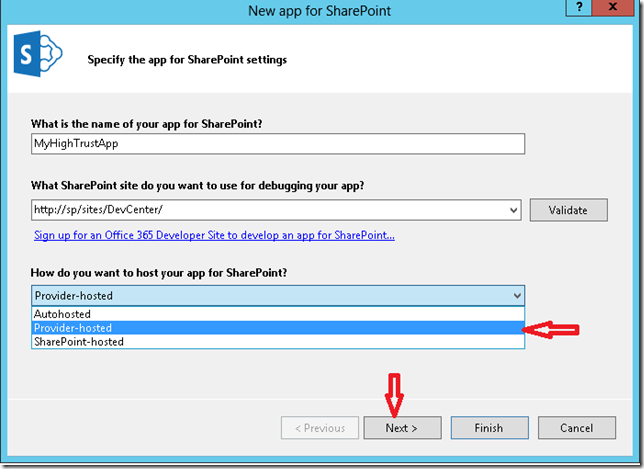


# Create a Simple “High Trust” Provider Hosted App using Visual Studio 2012

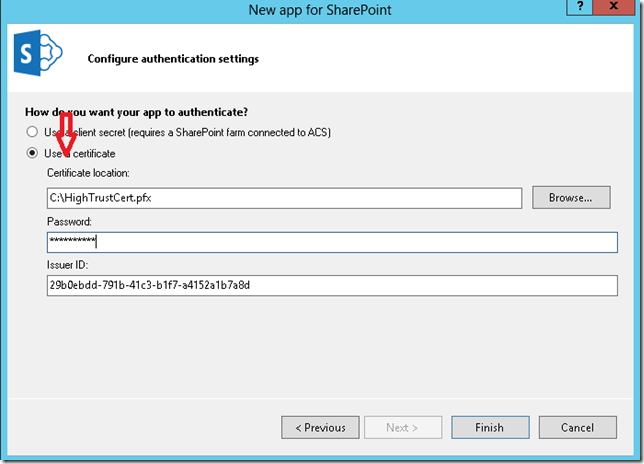
1.  Open Visual Studio 2012, under Office/SharePoint section select **App for SharePoint 2013**. Enter a name and click **OK**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/8400.image_5F00_283EFA41.png)

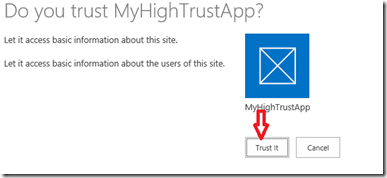
2. Select**Provider-hosted** and don’t forget to Click **Next**(and not Finish)

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/0647.image_5F00_2F9AB9EE.png)

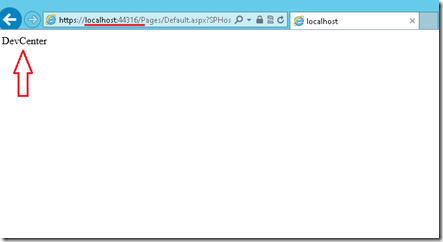
3. Then select **Use a Certificate**and provide the location of your certificate **.pfx**file. Provide the same password as provided above**.**The Issuer ID should be same as created above**.**Then Click **Finish.**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/5822.image_5F00_32EC2BC9.png)

4. The App Solution should be set up with its two projects.  Since we don’t intend to do anything extravagant, if you go ahead and Hit F5. After a few seconds you should see the below screen asking whether you ‘Trust’ this App. Click **Trust it.**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/0513.image_5F00_2F8A9421.png)

5. And we are done!! You should see the title of your Developer site displayed as shown below. Yay Smile

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-57-92-metablogapi/6523.image_5F00_3A5F9F3A.png)

6. Note that the App is right now hosted on the **IIS Express**.

# **ClientID generated above: put them in web.config of the appweb project and AppManifest.xml file of the app project. When we do it, visual studio will not have to generate the clientID everytime we debug, and it will be possible for us to register clientID(appID) to the sharepoint app permission page (which we have done in the powershell script above).**

# In the next step, we will deploy the app web to the local IIS for a better debug.

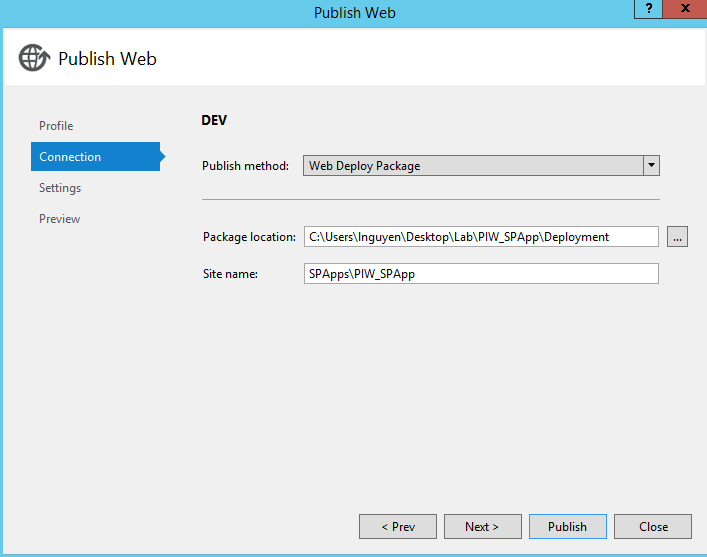
# Deployment

## Publish the remote web

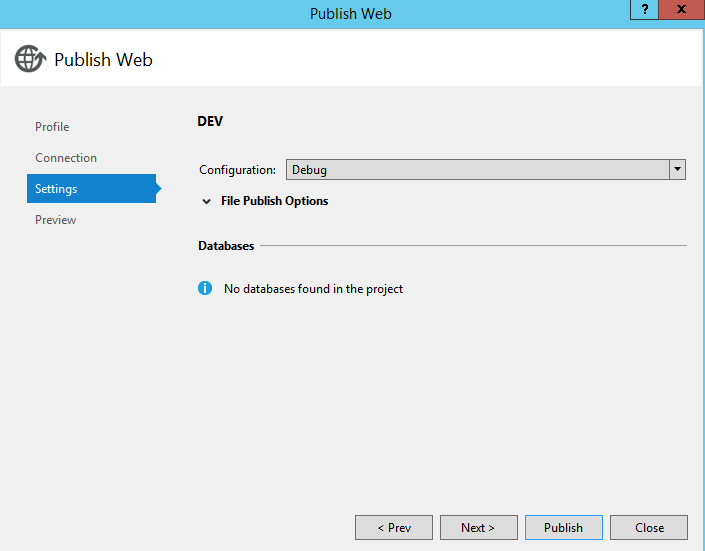
Modify the appManifest.xml, set the Start page to your new IIS location (in my dev machine, it is <https://dev.spapps.ferc.gov/PIW_SPApp/pages/default.aspx>)

Right click in app web project, select Publish.

Create new Profile (Dev, Test, Prod)



Change the configuration if deploy to Test or Prod



The next time need to deploy, just select the pre-defined profile to deploy.

Hit Publish, it will create a folder and files under specified location

Copy the folder to server need to deploy the remote web, Open command prompt, navigate to the folder and execute the command: [ProjectName].deploy.cmd /y

The command will copy all necessary file to the local IIS pre-defined in the Publishing process.

Open IIS to double check the virtual directory and files.

Open IE, browse to the default.aspx page (in my dev machine, it is <https://dev.spapps.ferc.gov/PIW_SPApp/pages/default.aspx>). It should say: error while processing, because the request is not initiated from SharePoint.

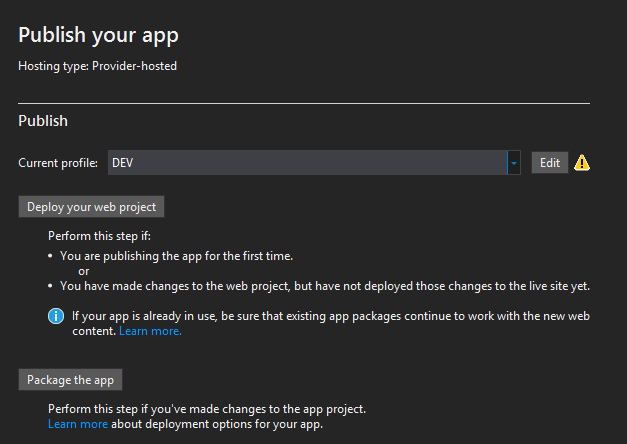
## Publish the app

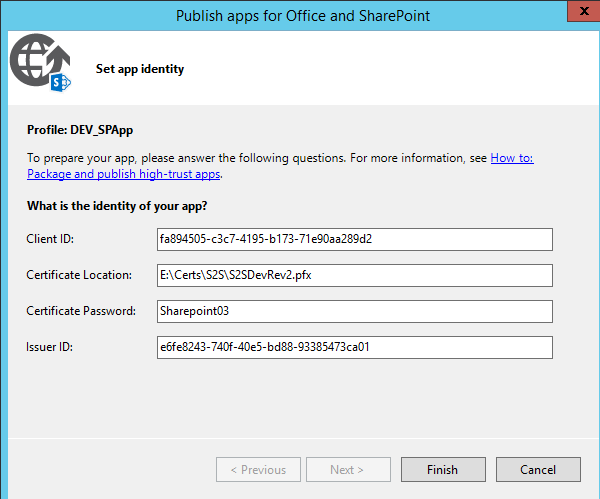
After remote web is published successfully, the next step is publish the app to Sharepoint.

Make sure team site is created.

From visual studio, Right click the app project, select Publish

Create new profile if you don’t have one.





If deploy to Test or Prod,

ReGenrate the new clientID and register it along with the issuerID

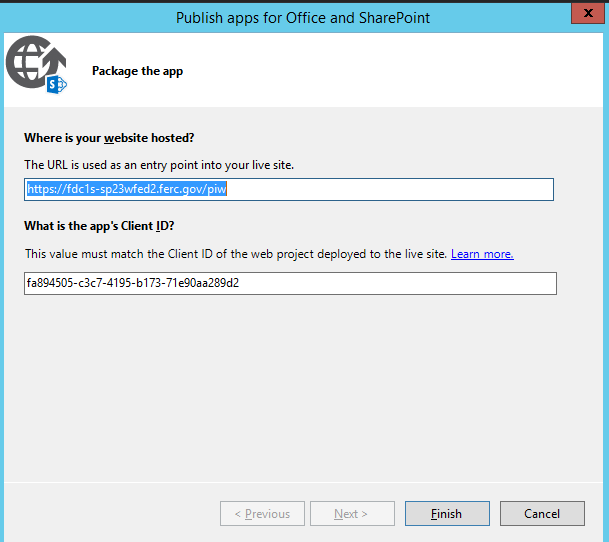
Update the Certificate Location and Password

Update the IssuerID

Update the web.config

Update the appmanifest.xml

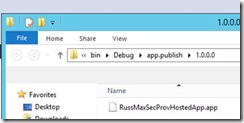
Hit OK, then hit package the app. Specify the sharepoint teamsite url and clientID



4. Click Finish and it builds the app package automatically:

[clip_image039](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-97-52-metablogapi/5483.clip_5F00_image039_5F00_6E56182A.jpg)

Note: It automatically opens windows explorer to the app.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-97-52-metablogapi/6406.clip_5F00_image041_5F00_56BE6AB7.jpg)

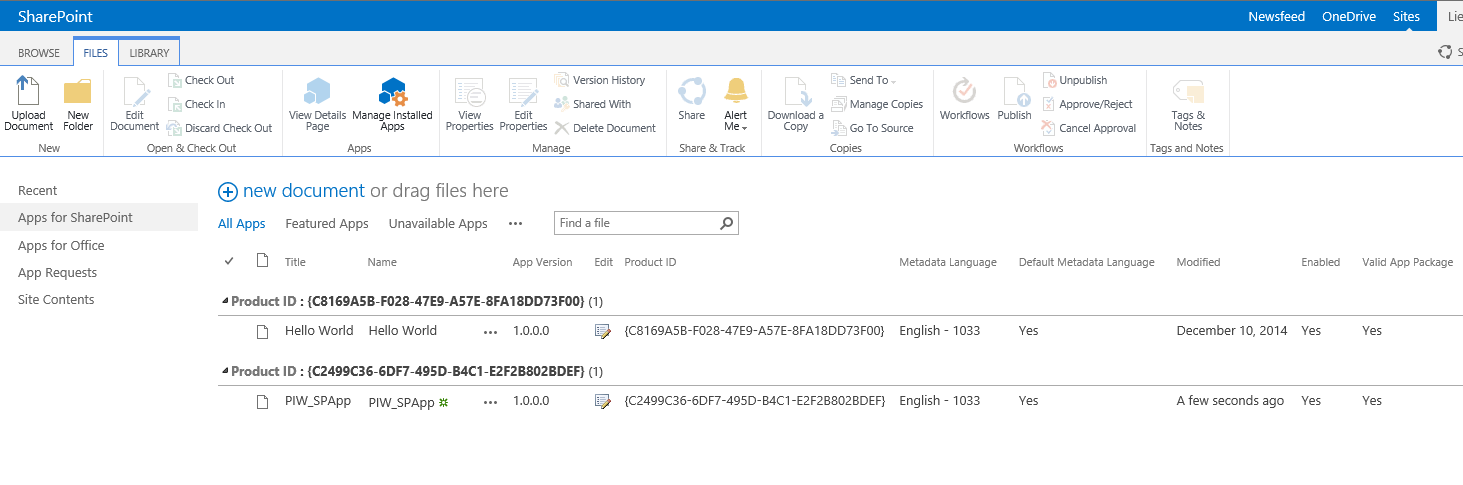
5. Copy this file over to a server that can access the app catalog.

# Add app to app catalog

For an app to be consumed, it must be added to an app catalog.

1. Navigate to the app catalog and select Apps for SharePoint   
2. Select New App and upload the .app file produced from the last set of steps

In my case, it looks like the following after upload:

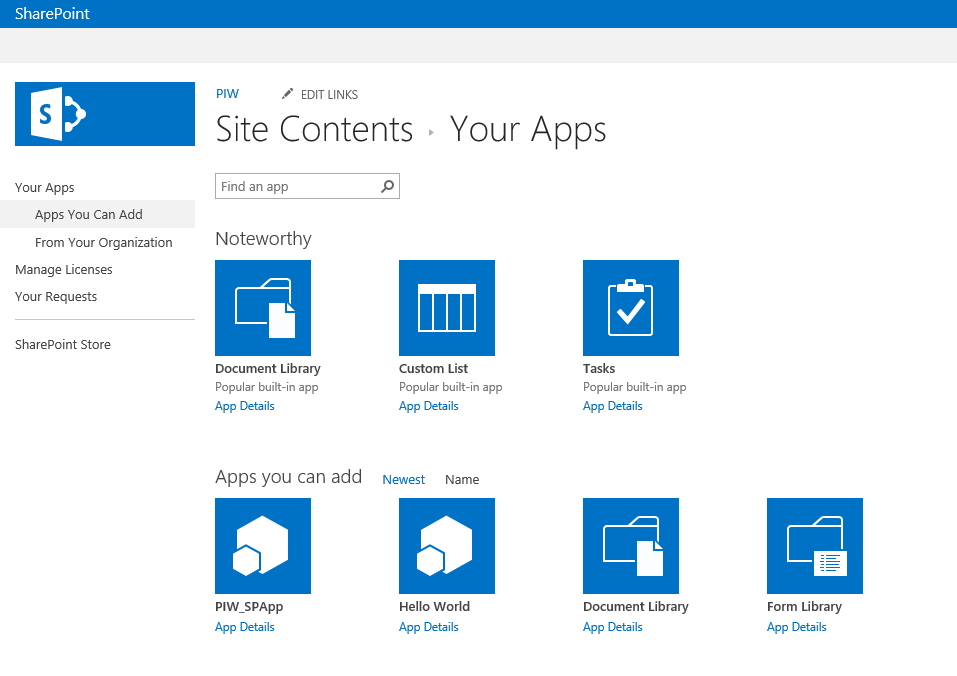


Note: I added PIW\_SPApp so I have two apps available now.

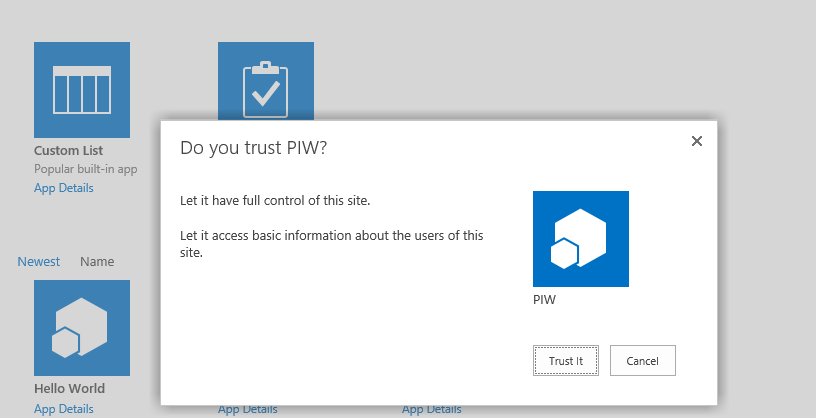
**Install App to host site**

1. Access a team site and selected site contents and clicked Add App.

Note: I see it here



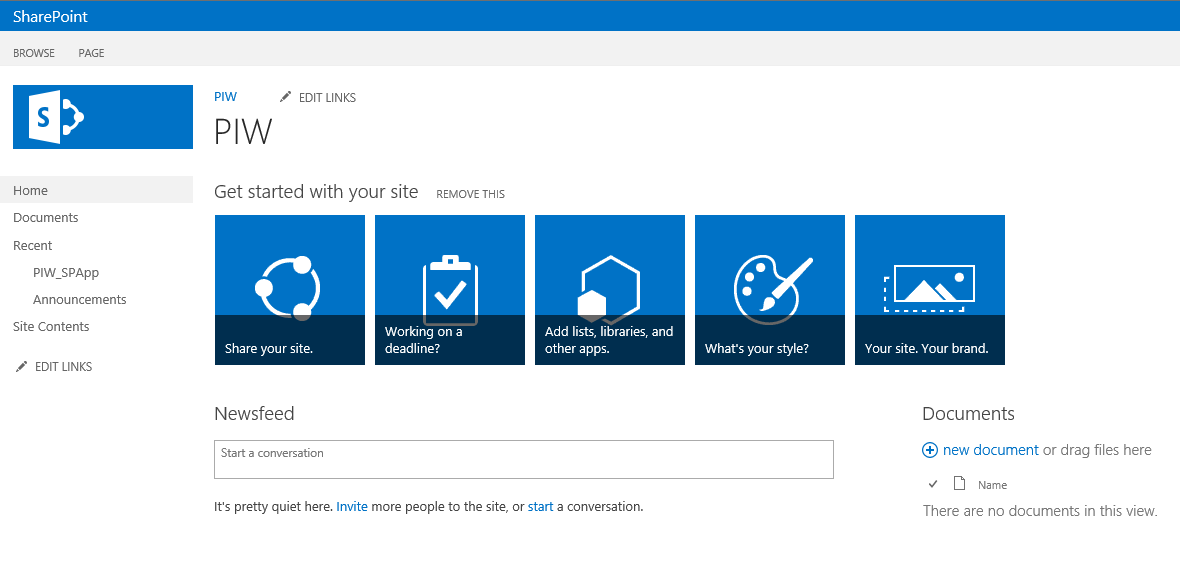
2. Click on it and click Trust It



Note: If it errors on this step and you’re logged in as the system account, try again using a non-system account.

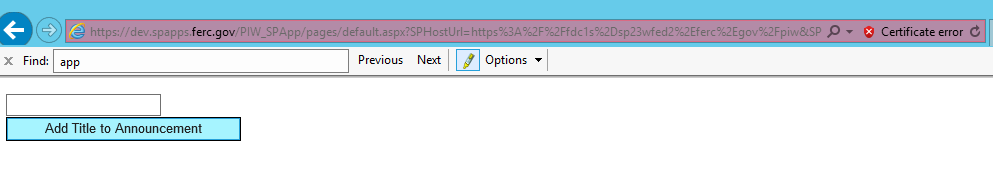
3. After install, test by clicking on the app.

For Example:  Clicked on the following (Before):



And the redirect works!

After:



Remote Event Receiver Debug – ERROR

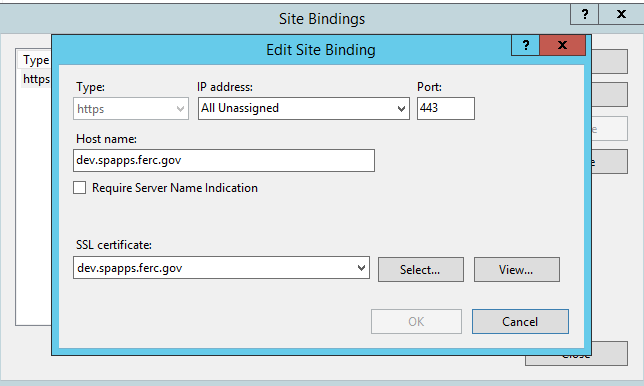
During Event receiver development , if ULS report error : Calling remote event receiver failed …. Could not establish trust relationship for the SSL/TLS secure channel with authority 'spapps.rchen.com'.

It is because the certificate using in DEV is not trusted in SharePoint and Local machine. It also generate the “certificate error” from the app web like above picture

Step 1: Generate a self-signed certificate and register it to the IIS. Open Visual Studio command prompt and execute the scripts. Note: the Name CANNOT have wildcard in it, like \*.dev.spapps.ferc.gov. It must match to the header.



Step 2: Open IIS Manaager, verify the certificate is added. Bind it to your SPApps virtual directory, specify the SSL



Step 3: Open MMC.exe and add Certificates snap-in, Copy the new certificate from “Personal/Certificates” folder to “Trusted Root Certification Authorities/Certificates” folder.

Or Open IE, browse to the site, you will see “certificate error” message, click on it, select Install Certificate.

Now you should not receive the “certificate error” when open the site from your dev server.

Step 4: Add to SharePoint Trust Root Authority through Central Admin (May not necessary, My dev does not need it)

* 1. Security -> Managed Trust -> New
  2. Enter name and browse to the certificate. Click OK