~~NUGET~~

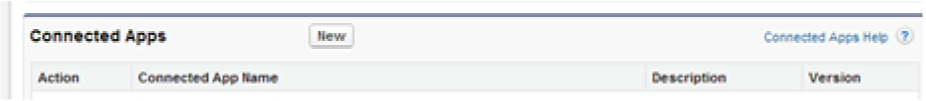
~~To conserve space in source control, I haven't checked in nuget packages for external dependencies. You may need to configure Visual Studio to automatically download these assemblies. This link explains how this works: http://docs.nuget.org/docs/workflows/using-nuget-without-committing-packages~~

HOW TO CREATE A "CONNECTED APP" FOR SALESFORCE CANVAS

This example uses an application on localhost (something a developer would do while developing an application), but if you are creating a connected app to deploy to a real server, use the real server endpoint instead (name, port, app path, etc).

From your SalesForce.com instance,

- Build > Create > Apps
- Connected Apps Section, click "New"

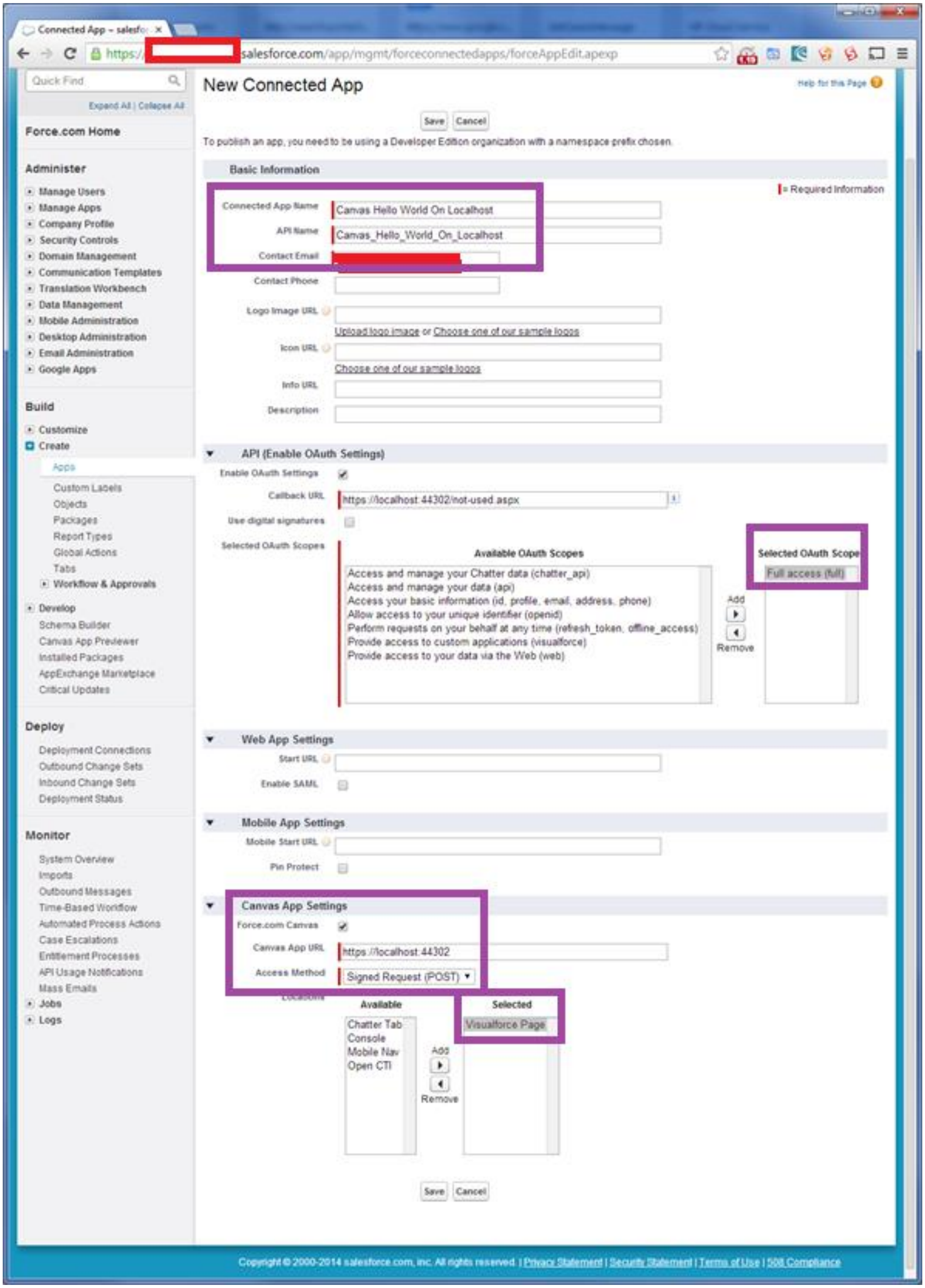


Connected Apps

Action	Connected App Name	Description	Version

Fill out the new Connected App with minimal information:

- Connected App Name
- API Name
- Contact Email
- API (Enable OAuth Settings):
  - Select "Enable OAuth Settings"
  - Selected OAuth Scope: "Full access (full)"
- Canvas App Settings
  - Select "Force.com Canvas"
  - Canvas App URL
  - Access Method: "Signed Request (POST)"
  - Locations: Select "Visualforce Page"



https://[redacted]salesforce.com/app/mgmt/forceconnectedapps/forceAppEdit.apexp

New Connected App

To publish an app, you need to be using a Developer Edition organization with a namespace prefix chosen.

Basic Information

| = Required Information

Connected App Name: Canvas Hello World On Localhost

API Name: Canvas_Hello_World_On_Localhost

Contact Email: [redacted]

Contact Phone:

Logo Image URL:

Upload logo image or Choose one of our sample logos

Icon URL:

Choose one of our sample logos

Info URL:

Description:

API (Enable OAuth Settings)

Enable OAuth Settings ☑

Callback URL: https://localhost:44302/not-used.aspx

Use digital signatures ☐

Selected OAuth Scopes

Available OAuth Scopes		Selected OAuth Scope
Access and manage your Chatter data (chatter_api) Access and manage your data (api) Access your basic information (id, profile, email, address, phone) Allow access to your unique identifier (openid) Perform requests on your behalf at any time (refresh_token, offline_access) Provide access to custom applications (visualforce) Provide access to your data via the Web (web)	Add Remove	Full access (full)

Web App Settings

Start URL:

Enable SAML ☐

Mobile App Settings

Mobile Start URL:

Pin Protect ☐

Canvas App Settings

Force.com Canvas ☑

Canvas App URL: https://localhost:44302

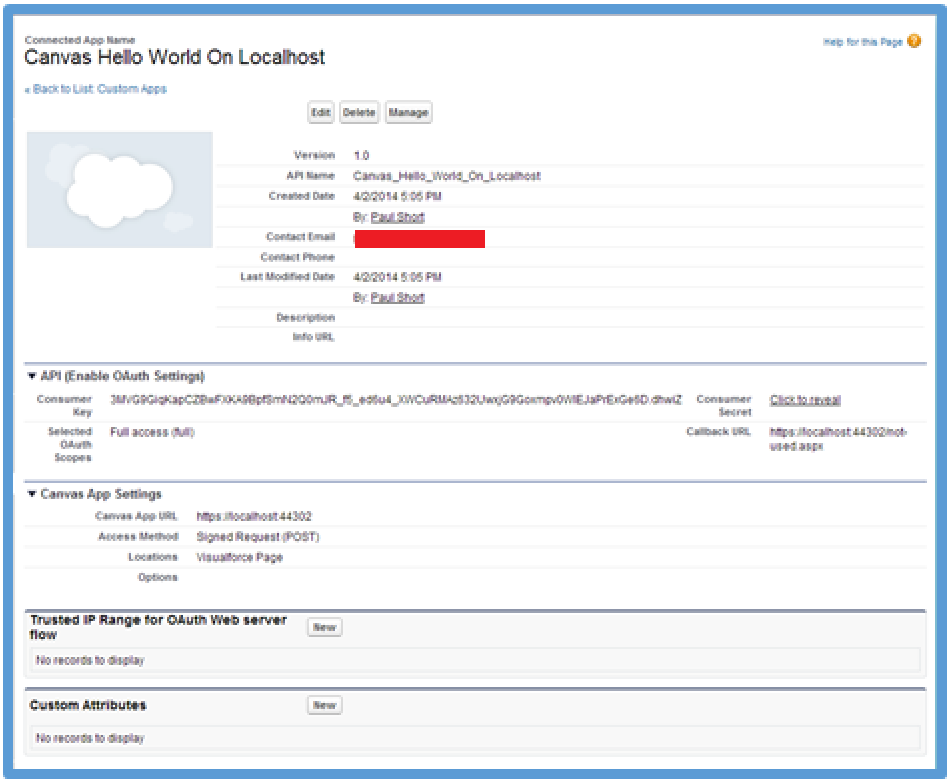
Access Method: Signed Request (POST)

Locations

Available		Selected
Chatter Tab Console Mobile Nav Open CTI	Add Remove	Visualforce Page

Copyright © 2000-2014 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance

After the App is created, click “Manage”:



Connected App Name

Canvas Hello World On Localhost

« Back to List: Custom Apps

Version	1.0
API Name	Canvas_Hello_World_On_Localhost
Created Date	4/2/2014 5:05 PM
	By: Paul Short
Contact Email	[redacted]
Contact Phone	
Last Modified Date	4/2/2014 5:05 PM
	By: Paul Short
Description	
Info URL	

▼ API (Enable OAuth Settings)

Consumer Key	3MVG9GiqKapCZBwFXKA9BpfSmN2Q0mJR_f5_ed6u4_XWCuRMAz632UwqG9Gompv0WIEJaPrExGe9D.dhwiZ	Consumer Secret	Click to reveal
Selected OAuth Scopes	Full access (full)	Callback URL	https://localhost:44302/not-used.aspx

▼ Canvas App Settings

Canvas App URL	https://localhost:44302
Access Method	Signed Request (POST)
Locations	Visualforce Page
Options	

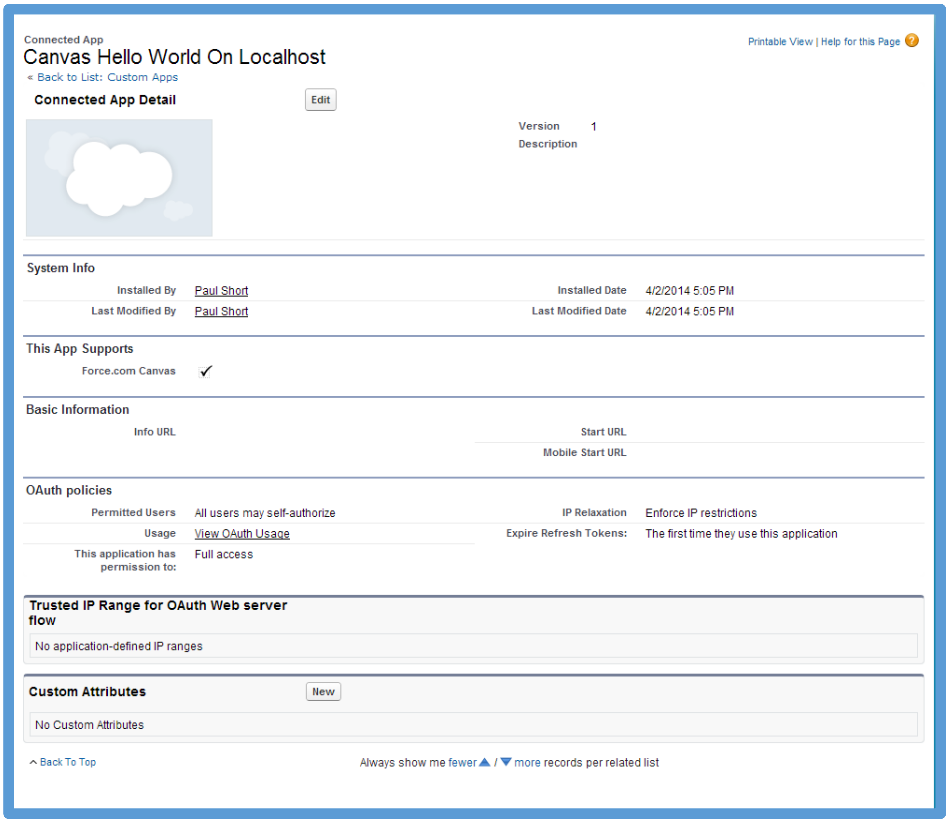
Trusted IP Range for OAuth Web server flow

No records to display

Custom Attributes

No records to display

…then “Edit”:



Connected App

Canvas Hello World On Localhost

« Back to List: Custom Apps

Connected App Detail

Version	1
Description	

System Info

Installed By	Paul Short	Installed Date	4/2/2014 5:05 PM
Last Modified By	Paul Short	Last Modified Date	4/2/2014 5:05 PM

This App Supports

Force.com Canvas	✓

Basic Information

Info URL		Start URL	
		Mobile Start URL	

OAuth policies

Permitted Users	All users may self-authorize	IP Relaxation	Enforce IP restrictions
Usage	View OAuth Usage	Expire Refresh Tokens:	The first time they use this application
This application has permission to:	Full access		

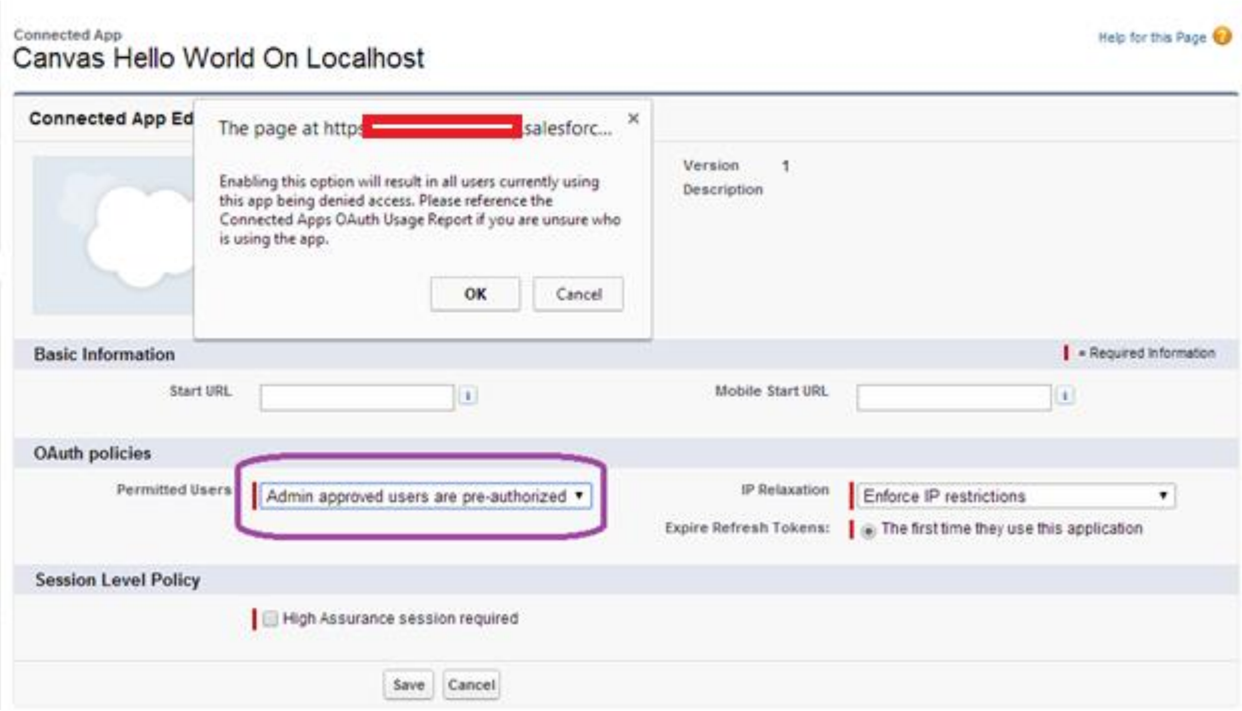
Trusted IP Range for OAuth Web server flow

No application-defined IP ranges

Custom Attributes

No Custom Attributes

Change OAuth policies for Permitted Users from “All users may self-authorize” to “Admin approved users are pre-authorized”; click OK on the warning; click Save:



Connected App

Canvas Hello World On Localhost

Connected App Ed

The page at https[redacted]salesforc...

Enabling this option will result in all users currently using this app being denied access. Please reference the Connected Apps OAuth Usage Report if you are unsure who is using the app.

Version 1

Description

Basic Information

▌ = Required Information

Start URL

Mobile Start URL

OAuth policies

Permitted Users: Admin approved users are pre-authorized ▾

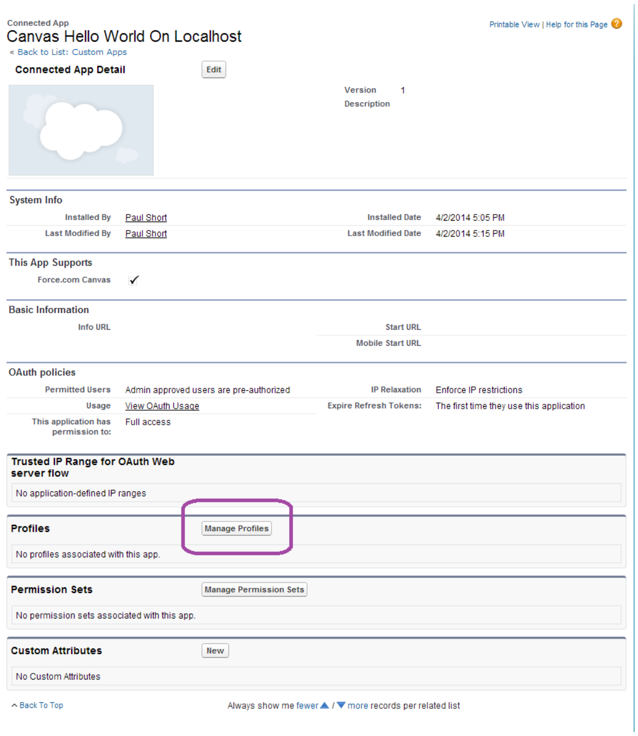
IP Relaxation: Enforce IP restrictions ▾

Expire Refresh Tokens: ● The first time they use this application

Session Level Policy

□ High Assurance session required

You’ll be taken back to the previous page, but now you’ll have two new Sections, Profiles and Permission Sets. If you application security is profile-based, click on “Manage Profiles” in the Profiles section:



Connected App

Canvas Hello World On Localhost

Printable View | Help for this Page
« Back to List: Custom Apps

Connected App Detail

Version 1

Description

System Info

Installed By	Paul Short	Installed Date	4/2/2014 5:05 PM
Last Modified By	Paul Short	Last Modified Date	4/2/2014 5:15 PM

This App Supports

Force.com Canvas ✓

Basic Information

Info URL		Start URL	
		Mobile Start URL	

OAuth policies

Permitted Users	Admin approved users are pre-authorized	IP Relaxation	Enforce IP restrictions
Usage	View OAuth Usage	Expire Refresh Tokens:	The first time they use this application
This application has permission to:	Full access		

Trusted IP Range for OAuth Web server flow

No application-defined IP ranges

Profiles

No profiles associated with this app.

Permission Sets

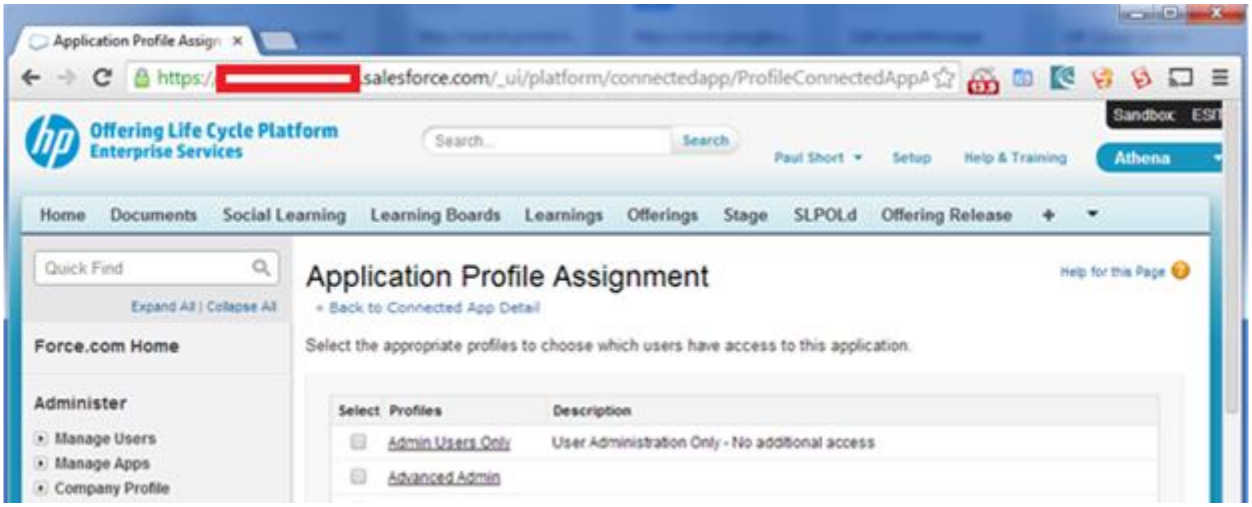
No permission sets associated with this app.

Custom Attributes

No Custom Attributes

Always show me fewer ▲ / ▼ more records per related list

The Application Profile Assignment page will probably have a lot of Profiles to choose from (too many to show here). These choices vary depending on the application and possibly the instance. After selecting the ones appropriate for your application, scroll to the bottom of the page and click Save:



https://[redacted]salesforce.com/_ui/platform/connectedapp/ProfileConnectedAppA

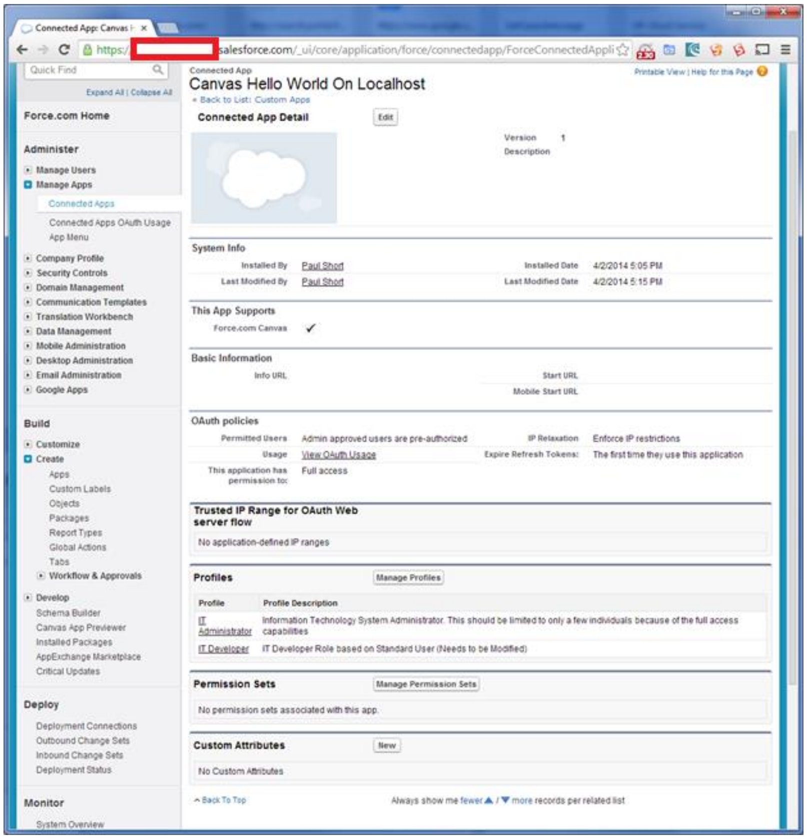
Application Profile Assignment

« Back to Connected App Detail

Select the appropriate profiles to choose which users have access to this application.

Select	Profiles	Description
□	Admin Users Only	User Administration Only - No additional access
□	Advanced Admin	

You'll be taken back to the previous page: the "Manage" Page for your Connected App.



https://[redacted]salesforce.com/_ui/core/application/force/connectedapp/ForceConnectedAppli

Connected App

Canvas Hello World On Localhost

« Back to List: Custom Apps

Connected App Detail [Edit]

Version 1

Description

System Info

Installed By	Paul Short	Installed Date	4/2/2014 5:05 PM
Last Modified By	Paul Short	Last Modified Date	4/2/2014 5:15 PM

This App Supports

Force.com Canvas ✓

Basic Information

Info URL

Start URL

Mobile Start URL

OAuth policies

Permitted Users	Admin approved users are pre-authorized	IP Relaxation	Enforce IP restrictions
Usage	View OAuth Usage	Expire Refresh Tokens:	The first time they use this application
This application has permission to:	Full access		

Trusted IP Range for OAuth Web server flow

No application-defined IP ranges

Profiles [Manage Profiles]

Profile	Profile Description
IT Administrator	Information Technology System Administrator. This should be limited to only a few individuals because of the full access capabilities
IT Developer	IT Developer Role based on Standard User (Needs to be Modified)

Permission Sets [Manage Permission Sets]

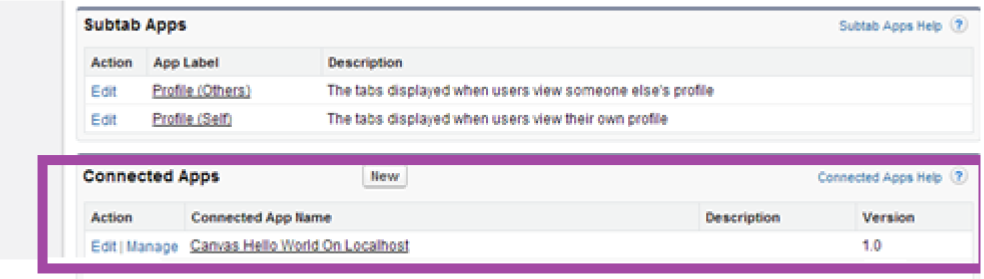
No permission sets associated with this app.

Custom Attributes [New]

No Custom Attributes

Repeat the process for ***Permission Sets*** *if your app uses them.*

Since we need to view the Connected App to see the Consumer Secret, go to Create > Apps, and scroll to the last section, Connected Apps. Click on Connected App Name “Canvas Hello World On Localhost”:



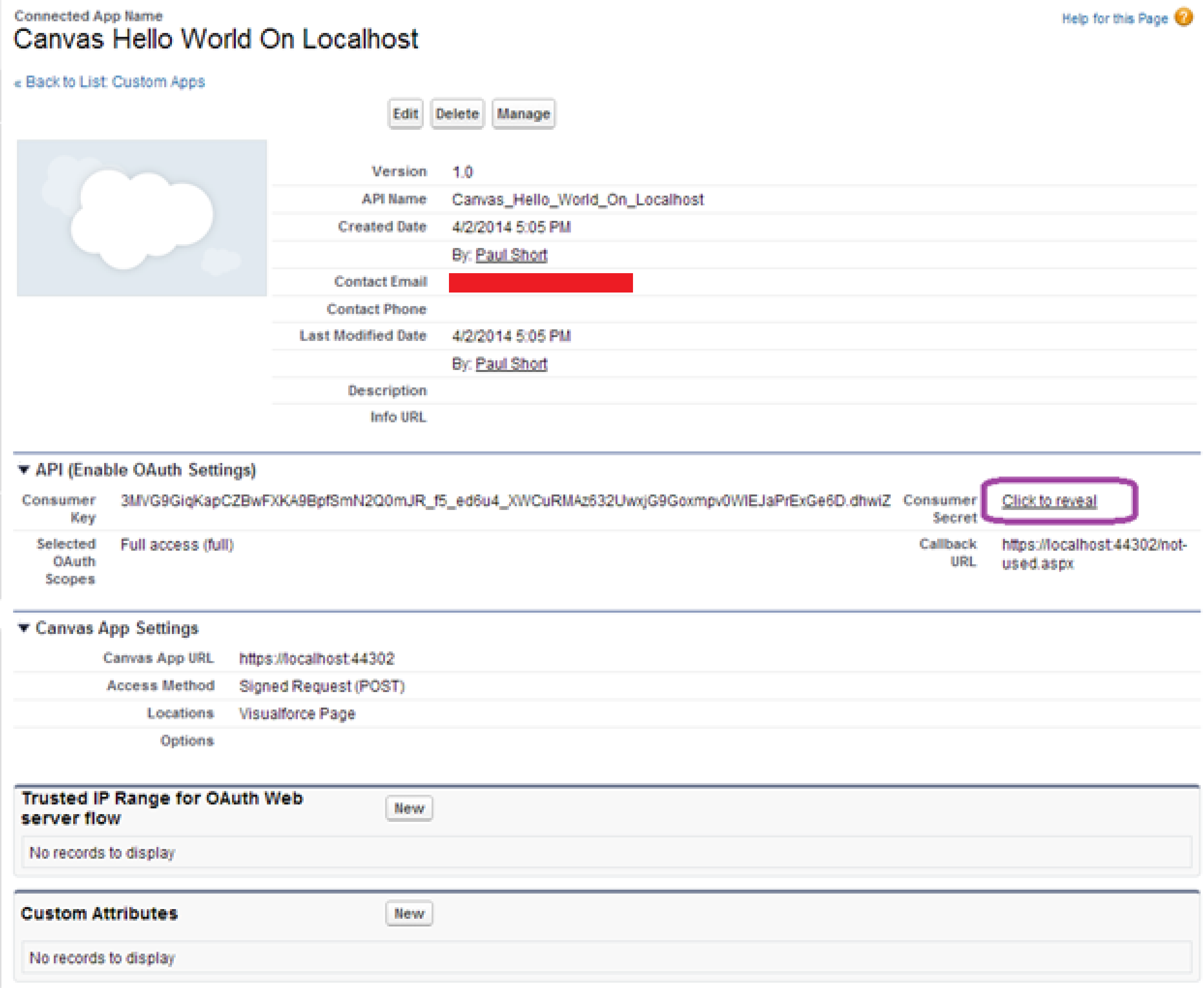
Subtab Apps

Action	App Label	Description
Edit	Profile (Others)	The tabs displayed when users view someone else's profile
Edit	Profile (Self)	The tabs displayed when users view their own profile

Connected Apps

Action	Connected App Name	Description	Version
Edit | Manage	Canvas Hello World On Localhost		1.0

Click on the link “click to reveal” to see the Consumer Secret:



Connected App Name

Canvas Hello World On Localhost

Version	1.0
API Name	Canvas_Hello_World_On_Localhost
Created Date	4/2/2014 5:05 PM
	By: Paul Short
Contact Email	[redacted]
Contact Phone	
Last Modified Date	4/2/2014 5:05 PM
	By: Paul Short
Description	
Info URL	

▼ API (Enable OAuth Settings)

Consumer Key	3MVG9GiqKapCZBwFXKA9BpfSmN2Q0mJR_f5_ed6u4_XWCuRMAz632UwxjG9Goxmpv0WIEJaPrExGe6D.dhwiZ	Consumer Secret	Click to reveal
Selected OAuth Scopes	Full access (full)	Callback URL	https://localhost:44302/not-used.aspx

▼ Canvas App Settings

Canvas App URL	https://localhost:44302
Access Method	Signed Request (POST)
Locations	Visualforce Page
Options	

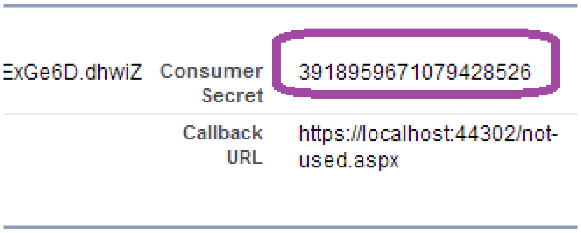
Trusted IP Range for OAuth Web server flow

No records to display

Custom Attributes

No records to display

Safeguard the Consumer Secret somewhere and give it to the developer or integrator who needs it.



ExGe6D.dhwiZ	Consumer Secret	3918959671079428526
	Callback URL	https://localhost:44302/not-used.aspx

Your application will need to know it to decode and verify signed requests using the Consumer Secret—it's an encrypted hash into the Consumer Key. Since a Connected App is unique, the corresponding Consumer Key and Consumer Secret are also unique--you cannot clone or copy it. If your application will be running on different instances it will need to use the consumer secret that corresponds to the unique connected app. If you ever come back to the connected app page, just remember to use "Edit" to view or modify the connected app and "Manage" to only modify its permissions.

Suggestion

One way to maintain consumer secrets is to keep them in a config file for your ASP.NET application:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appSettings>
. . .
    <add key="cs-key:localhost:44302/" value="9353398616531729939"/>             <!-- LOCAL -->
    <add key="cs-key:c0147892.foo.com:44302/Hello" value="2232324265999932928"/> <!-- DEV 2008 -->
    <add key="cs-key:c0143648.foo.com:44302/Hello" value="2616573287664899249"/> <!-- DEV 2012 -->
    <add key="cs-key:itg-is-not-yet-defined:44302/Hello" value="undefined"/>     <!-- ITG -->
    <add key="cs-key:pro-is-not-yet-defined:44302/Hello" value="undefined"/>     <!-- PRO -->
. . .
```

WARNING: *Depending on your security requirements, you may need to further obfuscate or encrypt your consumer secrets—safeguard them as if they were passwords. For this simple example we will just show them unencrypted.*

Fetch and cache the consumer secret for the lifetime of your app:

```
private string GetConsumerSecret()
{
    // Example for both ASP.Net MVC and ASP.Net Web Forms:
    // Since the consumer secret shouldn't change often, we'll put it in the Application Cache
    // for this simple example, but you may want cache it differently in a production application.
    string cachedConsumerSecret =
        (HttpContext.Current.Application["ConsumerSecret"] ?? String.Empty).ToString();
    if (!String.IsNullOrEmpty(cachedConsumerSecret))
    {
        return cachedConsumerSecret;
    }

    // We use key names in the format "cs-key:<server>:<port><app-path>" so that we
    // can maintain a consumer secret per server + port + app instance
    string key = String.Format("cs-key:{0}:{1}{2}",
        Request.ServerVariables["SERVER_NAME"],
        Request.Url.Port,
        Request.ApplicationPath);

    string secret = ConfigurationManager.AppSettings[key];
    if (!String.IsNullOrEmpty(secret))
    {
        HttpContext.Current.Application["ConsumerSecret"] = secret;
    }
    return secret;
}
```

If you have a lot of servers and/or don’t want to have them listed all in one file, you may want to do a web.config XML Document transform for deployment instead of the mapping technique I’m showing. It depends on how your organization or customer handles configuration management and deployment to servers.

END.