Application: **Adopta – Web AppBuilder**

Version: **2.0**

Document: **Readme for setting up GP service**

Date: **May 27, 2016**

Author: **CyberTech Systems and Software Ltd.**

**Description**Adopta is a configurable ESRI Web AppBuilder widget which helps citizens create, adopt, maintain and review status of various utilities. This document describes the steps to publish GP service for sending email for Adopta.

**Publishing the geoprocessing service**

The outline of steps to be followed to publish the geoprocessing tool is as follows:

1. Create a table in an enterprise geodatabase to store users
2. Install ArcREST python module
3. Run the geoprocessing tool
4. Publish the result as geoprocessing service

**Create a table in an enterprise geodatabase to store users**

1. Create a table in an enterprise geodatabase with the following schema
   1. Email – text field type
   2. Usertoken – guid field type
   3. Tokendate – date field type
   4. Teamname – text field type
   5. Firstname – text field type
   6. Lastname – text field type
2. Enable globalids on this table
3. Register this database with ArcGIS Server

**Install ArcREST Python module**

1. Download the latest archive from <https://github.com/Esri/ArcREST/archive/master.zip> and extract it to a suitable directory
2. Open a command prompt and cd into the “src” directory inside the extract location
3. Install the module for ArcGIS Desktop by running the following command

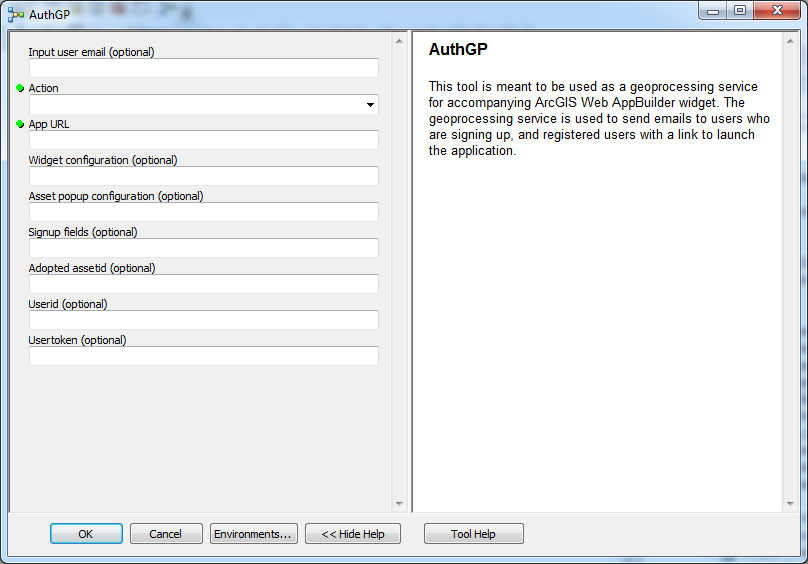
C:\Python27\ArcGIS10.4\python.exe setup.py install

1. Install the module for ArcGIS Server by running the following command

C:\Python27\ ArcGISx6410.4\python.exe setup.py install

**Run the “AuthGP” model by following these steps**

1. Right-click the AuthGP model and open it in edit mode
2. Double click the AuthGPScript tool inside the model and provide the following parameters
   1. **User table:** Browse to the table for storing users in an enterprise geodatabase
   2. **User email field:** Select field to store email address
   3. **Team field(optional):** Select field to store team names. If you do not select a team field, the sign up screen in widget does not provide an option to enter team name.
   4. **User token field:** Select field to store user tokens (guid)
   5. **Token date field:** Select field to store token generation dates
   6. **Token validity time:** Please enter time in minutes for expiry of tokens
   7. **Asset layer URL:** Full url to feature service (with layerid) containing assets
   8. **Asset layer username:** Credentials to access the feature service if secured
   9. **Asset layer password:** Credentials to access the feature service if secured
   10. **Asset layer portal url:** URL of the portal hosting the feature service
   11. **From email address:** The email address from which emails will be sent to users
   12. **Signup email subject line:** Subject line to be used for sign up emails
   13. **Signup email template:** HTML template to be used for sign up emails
   14. **Login email subject line:** Subject line to be used for sign up emails
   15. **Login email template:** HTML template to be used for sign up emails
   16. **SMTP Server:** Server hostname and port of smtp server
   17. **SMTP Username:** Required only if SMTP server is configured to authenticate
   18. **SMTP Password:** Required only if SMTP server is configured to authenticate
   19. **Use TLS:** Check this if your SMTP server requires TLS
3. Save the model and close it
4. Open the model to display its input parameters as shown in the graphic below



1. To test and review the signup email template, you need to add a user first by filling up the following parameters
   1. **Input user email**: Enter your email address so you can verify the receipt of sign up email
   2. **Action**: “Signup”
   3. **App URL**: Full url to the deployed location of the application
   4. Leave the other parameters empty
   5. Run the tool
   6. Review the sign up email you receive
2. To test and review the login email template, you need to run the tool again with action = “login” and provide the same email address you signed up with in the previous step
   1. **Input user email**: Enter the email address you signed up with
   2. **Action**: “Login”
   3. **App URL**: Full url to the deployed location of the application
   4. Leave the other parameters empty
   5. Run the tool
   6. Review the login email you receive

**Publish the geoprocessing service by following the steps below:**

1. Go to **Geo-Processing-> Result** window. Expand the **Current session** node (if it is collapsed) and right click on the result of the “Signup” action.
2. Go to **Share As -> Geo-processing Service.** “**Share As Service**” pop-up will appear. Select “**Publish a Service**” and click “**Next**”.
3. From “**Publish a Service**” pop up choose from drop down the **ArcGIS server connection** on which you want to publish the service. If you’re required server is not appearing in the drop down list click on “**Add ArcGIS Server**” icon to add.
4. Provide the **service name** to be displayed when it is published and click “**Next**”.
5. Select the **folder** in which the service should be shown. If necessary create a new one. Default is “root” folder of the server. Click “**Continue**”.
6. “**Service Editor**” window will be shown. Here you have to provide all the necessary properties of the service.
7. Go to “**Parameters**” tab. In “**Properties**” section, set the **Message Level** to **Info** and **Execution mode** to **Synchronous**. It is beneficial to get the status messages of the service while executing.
8. Click on “**Publish**” to publish it on specified server.
9. The login and signup template files will be copied to server.
10. Browse to the REST end point of the geoprocessing service. Ensure that the parameters are same as shown in the screenshot below:  
    