HOW TO DEVELOP EVENT APPS

COUNTERSOFT

How to Develop Event Apps

Contents

1. Introduction

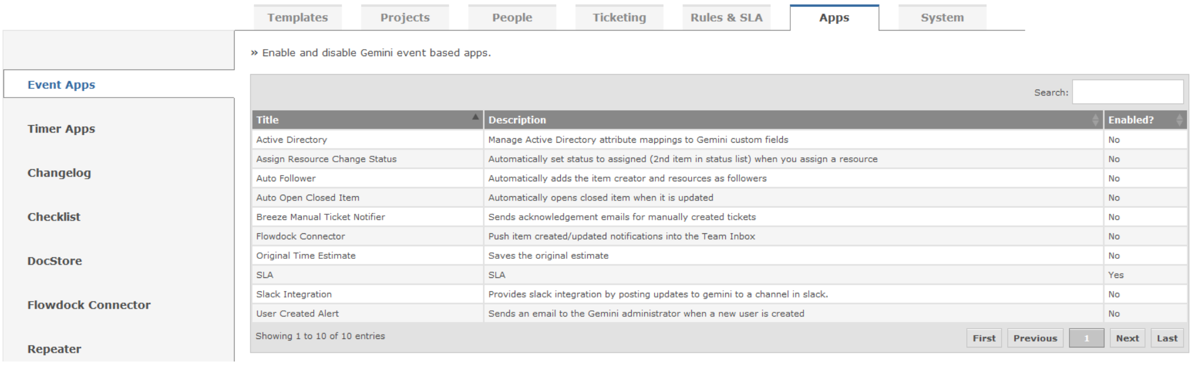
Gemini has lots of features built in, which you can explore in our docs. However there are times when you need something to work very specifically to your requirements in which case you can create your own apps using the Gemini App Framework.

This guide will show you how to create and deploy a basic Gemini event app.

Event apps allow you to subscribe to Event Listeners and have the event apps invoked when something happens. We will implement an *IIssueBeforeListener* allowing you to modify or work with the data **before** Gemini completes an action.

You can find a link to a list of event apps with source code in section References.

Gemini Administrators can enable or disable event apps.



» Enable and disable Gemini event based apps.

Title	Description	Enabled?
Active Directory	Manage Active Directory attribute mappings to Gemini custom fields	No
Assign Resource Change Status	Automatically set status to assigned (2nd item in status list) when you assign a resource	No
Auto Follower	Automatically adds the item creator and resources as followers	No
Auto Open Closed Item	Automatically opens closed item when it is updated	No
Breeze Manual Ticket Notifier	Sends acknowledgement emails for manually created tickets	No
Flowdock Connector	Push item created/updated notifications into the Team Inbox	No
Original Time Estimate	Saves the original estimate	No
SLA	SLA	Yes
Slack Integration	Provides slack integration by posting updates to gemini to a channel in slack.	No
User Created Alert	Sends an email to the Gemini administrator when a new user is created	No

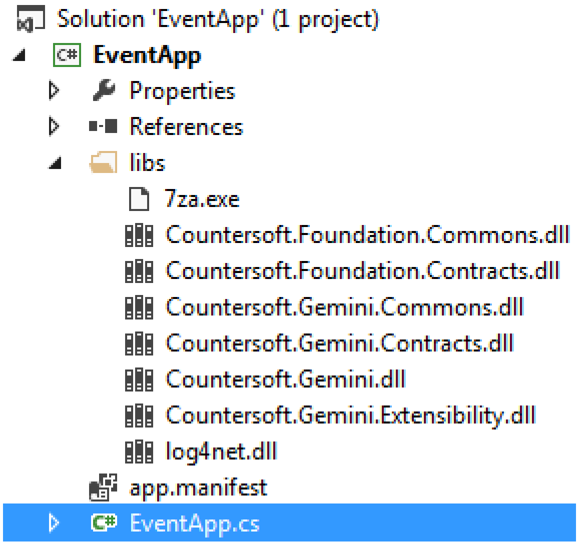
Showing 1 to 10 of 10 entries

2. Create Project

Open Visual Studio and download our project template *Gemini Event App Project Template* from Tools -> Extensions and Updates -> Online. Close and open Visual Studio again in order to select the new template from File -> New -> Project -> Visual C#. Enter a *Name* and *Solution name* and make sure you rename the files EventApp.cshtml, EventApp.css and EventApp.js to the Solution name.

3. Structure

The new template will create a project with the following structure.



```
Solution 'EventApp' (1 project)
◢ EventApp
    ▷ Properties
    ▷ References
    ◢ libs
          7za.exe
          Countersoft.Foundation.Commons.dll
          Countersoft.Foundation.Contracts.dll
          Countersoft.Gemini.Commons.dll
          Countersoft.Gemini.Contracts.dll
          Countersoft.Gemini.dll
          Countersoft.Gemini.Extensibility.dll
          log4net.dll
      app.manifest
    ▷ EventApp.cs
```

<u>Libs</u>

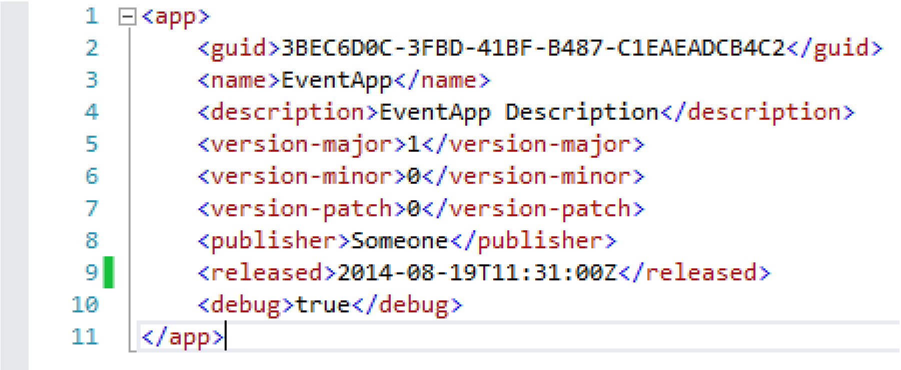
Contains all relevant DLL’s the app uses.

<u>Manifest</u>

The app.manifest file describes your app.

4. App.Manifest

The manifest file describes your app and is used during deployment.



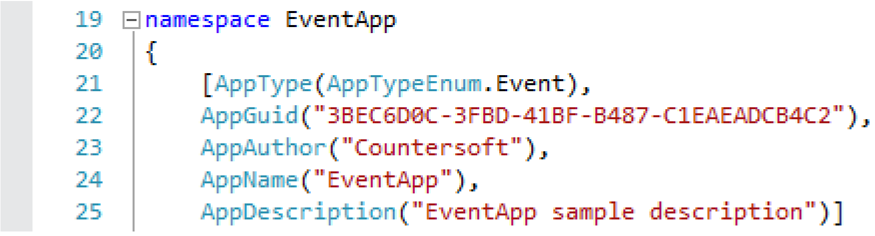
```xml
<app>
    <guid>3BEC6D0C-3FBD-41BF-B487-C1EAEADCB4C2</guid>
    <name>EventApp</name>
    <description>EventApp Description</description>
    <version-major>1</version-major>
    <version-minor>0</version-minor>
    <version-patch>0</version-patch>
    <publisher>Someone</publisher>
    <released>2014-08-19T11:31:00Z</released>
    <debug>true</debug>
</app>
```

Option	**Description**
GUID	Globally unique identifer (GUID) for your app
Name	Application title displayed in Screen setup for Administrators to identify your app

Description	Brief description of the application
Version Major	Semantic versioning for your app
Version Minor	
Version Patch	
Publisher	Name of the organization/individual who owns the app
Released	Date the app was released
Debug	Can be *true* or *false*. When *true* app is not cached and can be changed on disk (useful during development)

5. Attributes

Set attribute’s to describe your app.



```
namespace EventApp
{
    [AppType(AppTypeEnum.Event),
    AppGuid("3BEC6D0C-3FBD-41BF-B487-C1EAEADCB4C2"),
    AppAuthor("Countersoft"),
    AppName("EventApp"),
    AppDescription("EventApp sample description")]
```

Attributes

Option	Description
AppType	The type of app i.e. Widget, Event, Timer etc.
AppGuid	Globally unique identifier (GUID) for your app
AppAuthor	Your name
AppName	Application title displayed in Screen setup for Administrators to identify your app

AppDescription	Brief description of the application

6. Project Setup

a) Replace all Countersoft DLL’s in the libs folder with the latest Gemini DLL’s from your %Gemini%/bin installation folder.

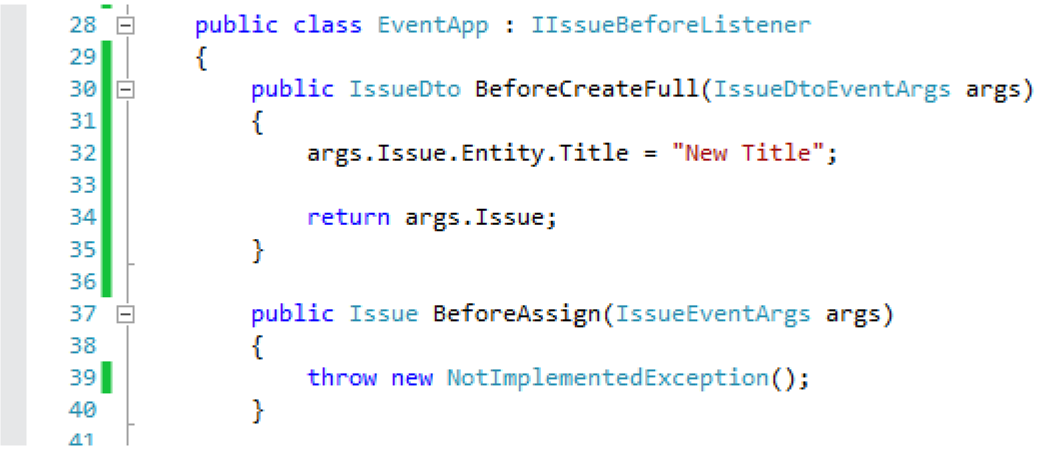
b) Open EventApp.cs and change the namespace, class name and all other references from *EventApp* to something unique. Use a name relevant to its function.

c) Replace the AppGuid with a new GUID. AppGuid is used in several places so do a Search and Replace.

Open the app.manifest and make sure the Guid is changed to the AppGuid. You can change the name and description as well. Leave the debug property as “true” for as long as you are developing your app. Change it to “false” when the app is ready to go live.

7. Event Listener

If you wanted to modify the issue before it was created, then you would add your logic to the method *BeforeCreateFull.*



```
public class EventApp : IIssueBeforeListener
{
    public IssueDto BeforeCreateFull(IssueDtoEventArgs args)
    {
        args.Issue.Entity.Title = "New Title";

        return args.Issue;
    }

    public Issue BeforeAssign(IssueEventArgs args)
    {
        throw new NotImplementedException();
    }
```

You can add more listeners by separating them with a comma. When you add a new listener, right click on it and select ‘implement interface’ to implement the methods. You can find a full list of listeners here.

8. Services

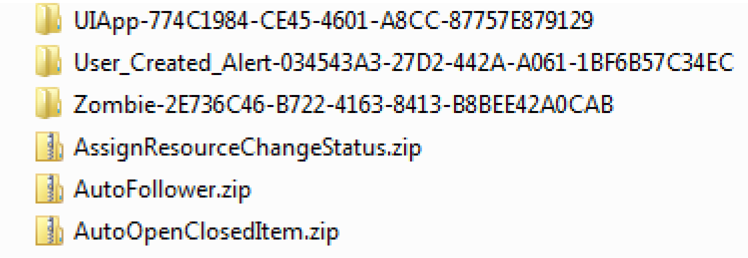
You can access the current issue via *args.Issue*.

Here are some important services the IssueEventArgs (*args.Context*) provides you with. The services offer methods to retrieve, create, update and delete items.

Name	**Description**
ProjectTemplates	Provides access project templates
Projects	Provides access projects
Issues	Provides access issues
Users	Provides access users
CustomFields	Provides access custom fields
NavigationCards	Provides access workspaces
Meta	Provides access meta data (status, priority etc.)

9. Deployment

Apps are deployed to the App_data/Apps folder which is located where you installed Gemini on your web server.



UIApp-774C1984-CE45-4601-A8CC-87757E879129
User_Created_Alert-034543A3-27D2-442A-A061-1BF6B57C34EC
Zombie-2E736C46-B722-4163-8413-B8BEE42A0CAB
AssignResourceChangeStatus.zip
AutoFollower.zip
AutoOpenClosedItem.zip

ZIP files are the packaged apps and the folders are deployed apps.

a) We use Post-Build Events in the project to generate the ZIP file. Build your solution and you should find your app's zip file in the bin/target directory.

b) Stop your Gemini application pool and copy the ZIP into %Gemini%/App_Data /apps

c) Start your Gemini application pool and this should extract the content of the ZIP into a folder

The app should now be available in Gemini. Make sure all relevant permissions are assigned for the app in Customize -> Apps -> AppName and also enable the app in Customize -> Template -> Process screens.

When you re-deploy the app then make sure the app's ZIP file and the folder are deleted from %Gemini%/App_Data/apps before you copy in the new ZIP file.

10. References Examples

Complete list of Event listeners. http://docs.countersoft.com/event-listeners/

Several Gemini app examples with source code http://countersoft.github.io/

Docs http://docs.countersoft.com/developing-custom-apps/