HOW TO DEVELOP FULL PAGE APPS

COUNTERSOFT

How to Develop Full Page Apps

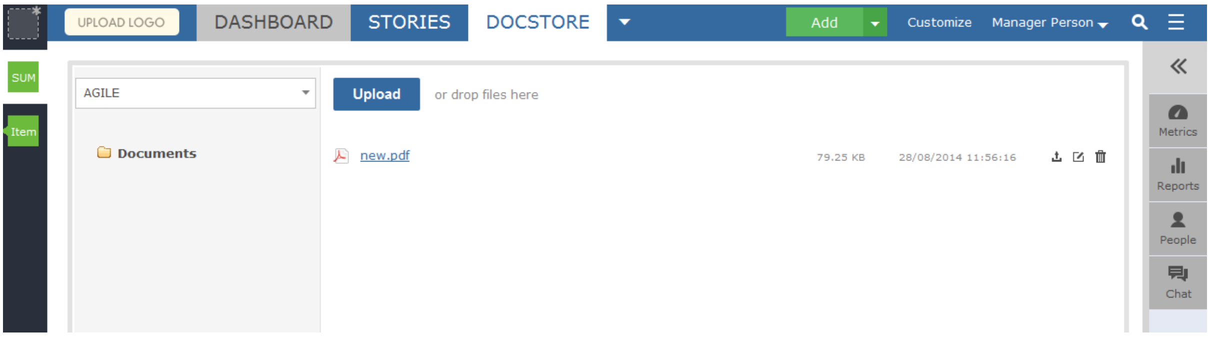
Contents

1. Introduction

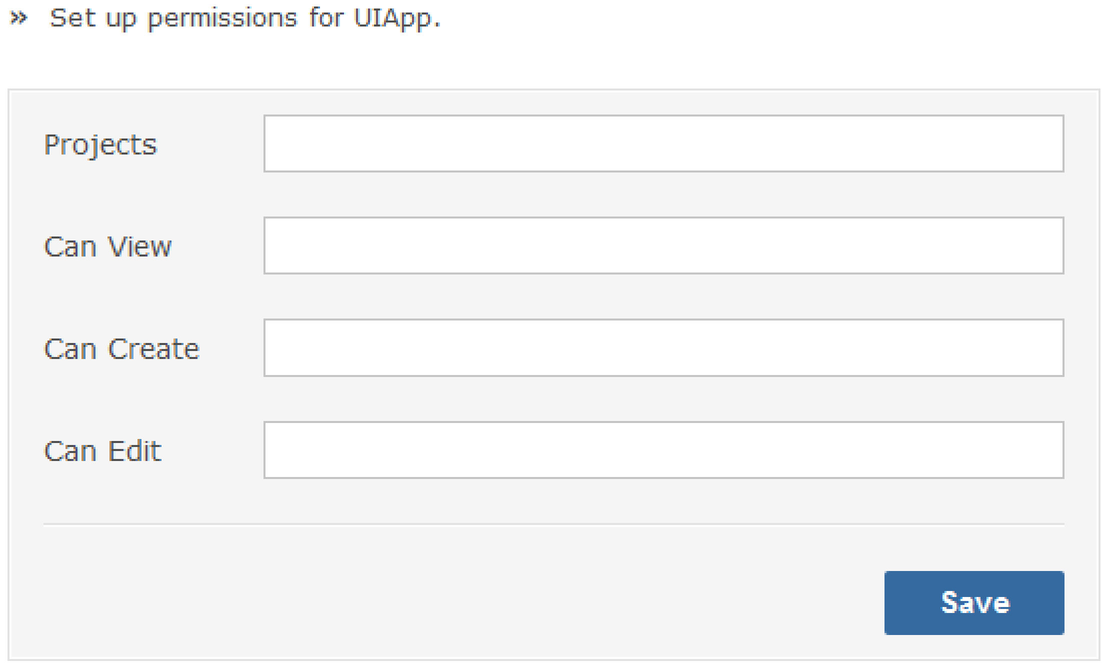
Gemini has lots of features built in, which you can explore in our docs. However there are times when you need something to work very specifically to your requirements in which case you can create your own apps using the Gemini App Framework.

This guide will show you how to create and deploy a Gemini full page app. A full page app allows you to easily extend the Gemini UI with your specific content. Your app will become accessible from the top menu once it has been deployed and the permissions are set.

We will be using our open-source DocStore app as reference.



You can have an optional configuration screen in Customize -> Apps -> AppName. You can set permissions for specific projects and then use this permission set to validate user access in your app.


» Set up permissions for UIApp.

Projects

Can View

Can Create

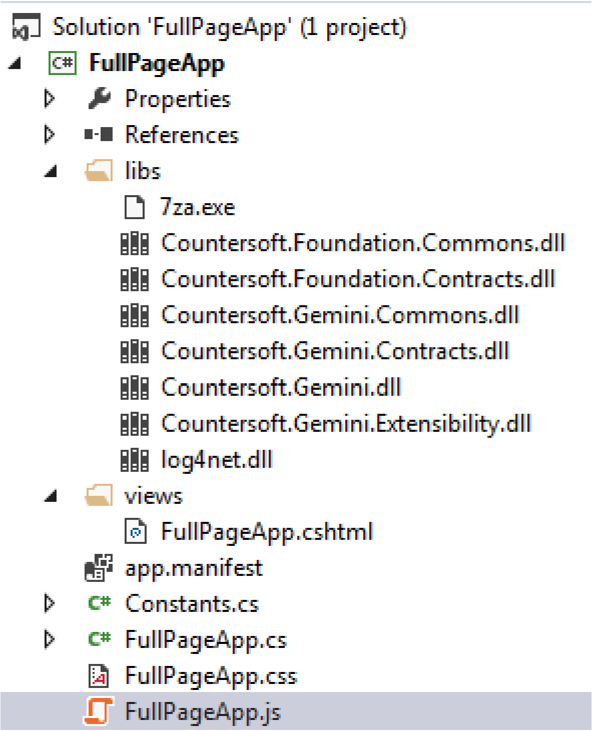
Can Edit

2. Create Project

Open Visual Studio and download our project template Gemini Full Page App 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 FullPageApp.cshtml, FullPageApp.css and FullPageApp.js to the Solution name.

3. Structure

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



```
Solution 'FullPageApp' (1 project)
  FullPageApp
    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
    views
      FullPageApp.cshtml
    app.manifest
    Constants.cs
    FullPageApp.cs
    FullPageApp.css
    FullPageApp.js
```

Libs

Contains all relevant DLL's the app uses.

Views

Views folder consists of Razor views – the user interface for your app if applicable.

Manifest

The app.manifest file describes your app.

CSS and Javascript

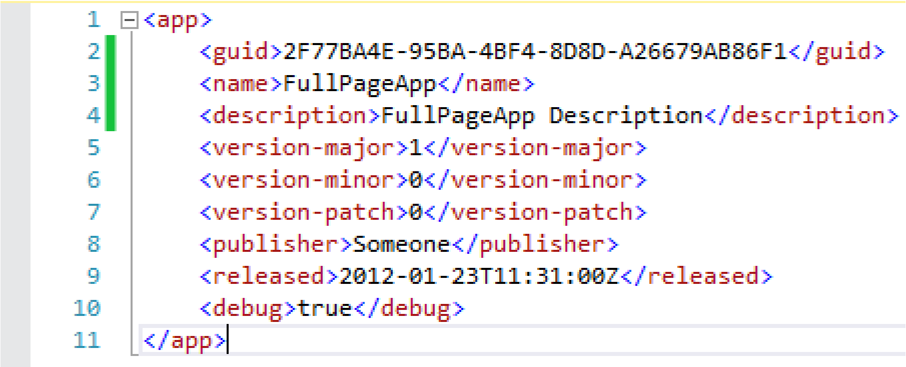
FullPageApp.css and FullPageApp.js are optional files. If you want to use them make sure they are referenced in the FullPageApp.cshtml file.

Constants

Constants.cs contains the AppGuid and AppControlGuid.

4. App.Manifest

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

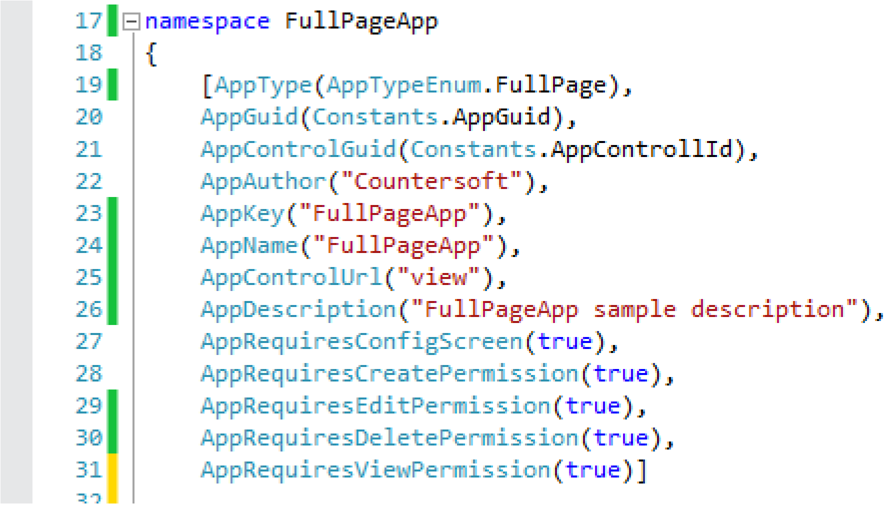


```
<app>
    <guid>2F77BA4E-95BA-4BF4-8D8D-A26679AB86F1</guid>
    <name>FullPageApp</name>
    <description>FullPageApp Description</description>
    <version-major>1</version-major>
    <version-minor>0</version-minor>
    <version-patch>0</version-patch>
    <publisher>Someone</publisher>
    <released>2012-01-23T11:31:00Z</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 attributes to describe your app and configuration screen.



```
namespace FullPageApp
{
    [AppType(AppTypeEnum.FullPage),
    AppGuid(Constants.AppGuid),
    AppControlGuid(Constants.AppControlId),
    AppAuthor("Countersoft"),
    AppKey("FullPageApp"),
    AppName("FullPageApp"),
    AppControlUrl("view"),
    AppDescription("FullPageApp sample description"),
    AppRequiresConfigScreen(true),
    AppRequiresCreatePermission(true),
    AppRequiresEditPermission(true),
    AppRequiresDeletePermission(true),
    AppRequiresViewPermission(true)]
```

Attributes

Option	Description
AppType	The type of app i.e. Widget, Event, Timer etc.
AppGuid	Globally unique identifier (GUID) for your app
AppControlGuid	Global unique identifier for the control. One app can have multiple controllers
AppAuthor	Your name
AppKey	URL identifier for the app http://gemini/workspace/2/app/**FullPageApp**/view
AppUrl	URL identifier for the method http://gemini/workspace/2/app/FullPageApp/**view**
AppName	Application title displayed in screen setup for Administrators to identify your app
AppDescription	Brief description of the application
AppRequiresConfigScreen	Can be *true* or *false*. Show a configuration page in Customize -> Apps

AppRequiresCreatePermission AppRequiresEditPermission AppRequiresViewPermission AppRequiresDeletePermission	Can be set to *true* or *false*. Show create, edit, view and delete permission fields in configuration page. Control which user can do which action in the app

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 FullPageApp.cs and change the namespace, class name and all other references from "FullPageApp" to something unique. Use a name relevant to its function.

c) Replace the AppGuid and AppControlGuid in Constants.cs with two new GUID's.

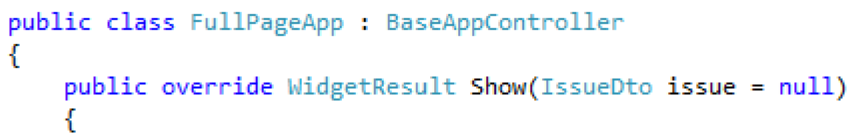
Open the app.manifest and make sure the Guid is changed to the AppGuid. You can change the name and description as well. Leave the debug property as "true" for as long as you are developing your app. Change it to "false" when the app is ready to go live.

7. Methods and Services

These are the available methods you can override.

Name	**Description**
Show(IssueDto)	Called when we need to show the content of the app. Requires an IssueDto as a parameter.
Caption(IssueDto)	Called every time we show the app's name. Requires an IssueDto as a parameter.
Configuration()	Override this method to show your own customized configuration view

The app's controller is derived from the main Gemini controller and that means you have access to most of Gemini's services.



```
public class FullPageApp : BaseAppController
{
    public override WidgetResult Show(IssueDto issue = null)
    {
```

List of some important services, which offer methods to retrieve, create, update and delete items.

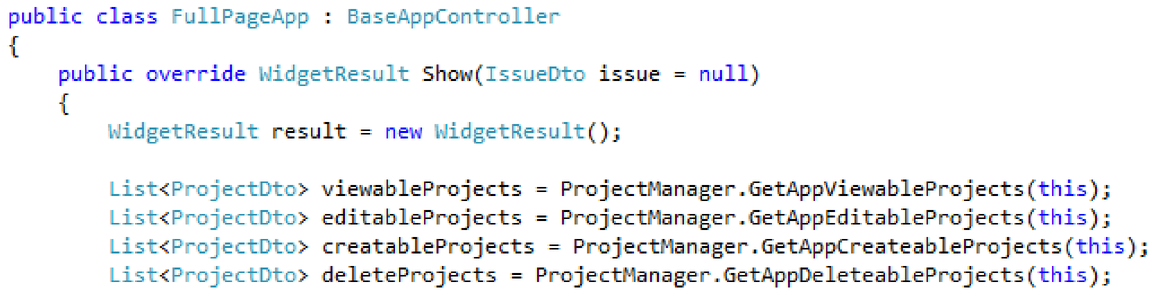
Name	**Description**
ProjectTemplateManager	Provides access to project templates
ProjectManager	Provides access to projects
IssueManager	Provides access to issues

UserManager	Provides access to users
CustomFieldManager	Provides access to custom fields
NavigationCardsManager	Provides access to workspaces
MetaManager	Provides access to meta data (status, priority etc.)

8. Permissions

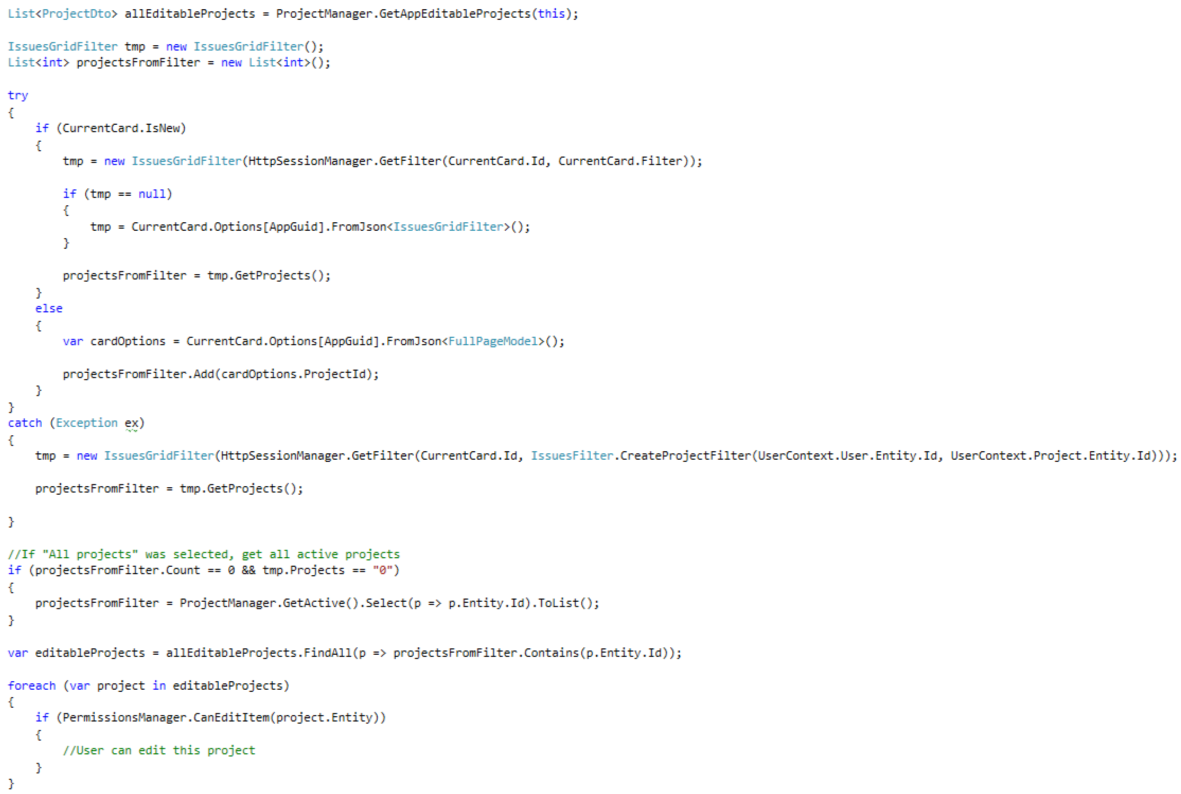
As mentioned earlier you can show an optional configuration screen in Customize -> Apps -> AppName. This is where you can define which projects the groups have access to.

Below you can see how to retrieve the projects for each action, set in the configuration screen.



```
public class FullPageApp : BaseAppController
{
    public override WidgetResult Show(IssueDto issue = null)
    {
        WidgetResult result = new WidgetResult();

        List<ProjectDto> viewableProjects = ProjectManager.GetAppViewableProjects(this);
        List<ProjectDto> editableProjects = ProjectManager.GetAppEditableProjects(this);
        List<ProjectDto> creatableProjects = ProjectManager.GetAppCreateableProjects(this);
        List<ProjectDto> deleteProjects = ProjectManager.GetAppDeleteableProjects(this);
```

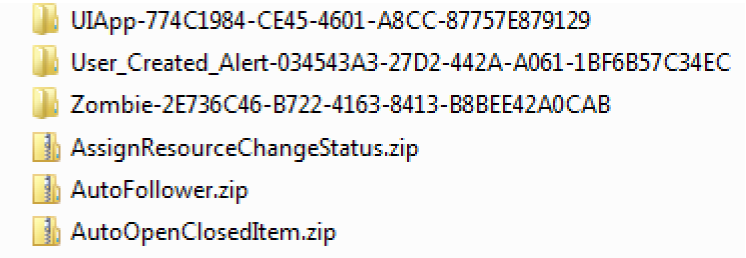
This shows you how to retrieve the projects from the current workspace and check if the user has permission to edit the project.



```
List<ProjectDto> allEditableProjects = ProjectManager.GetAppEditableProjects(this);

IssuesGridFilter tmp = new IssuesGridFilter();
List<int> projectsFromFilter = new List<int>();

try
{
    if (CurrentCard.IsNew)
    {
        tmp = new IssuesGridFilter(HttpSessionManager.GetFilter(CurrentCard.Id, CurrentCard.Filter));

        if (tmp == null)
        {
            tmp = CurrentCard.Options[AppGuid].FromJson<IssuesGridFilter>();
        }

        projectsFromFilter = tmp.GetProjects();
    }
    else
    {
        var cardOptions = CurrentCard.Options[AppGuid].FromJson<FullPageModel>();

        projectsFromFilter.Add(cardOptions.ProjectId);
    }
}
catch (Exception ex)
{
    tmp = new IssuesGridFilter(HttpSessionManager.GetFilter(CurrentCard.Id, IssuesFilter.CreateProjectFilter(UserContext.User.Entity.Id, UserContext.Project.Entity.Id)));

    projectsFromFilter = tmp.GetProjects();

}

//If "All projects" was selected, get all active projects
if (projectsFromFilter.Count == 0 && tmp.Projects == "0")
{
    projectsFromFilter = ProjectManager.GetActive().Select(p => p.Entity.Id).ToList();
}

var editableProjects = allEditableProjects.FindAll(p => projectsFromFilter.Contains(p.Entity.Id));

foreach (var project in editableProjects)
{
    if (PermissionsManager.CanEditItem(project.Entity))
    {
        //User can edit this project
    }
}
```

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.

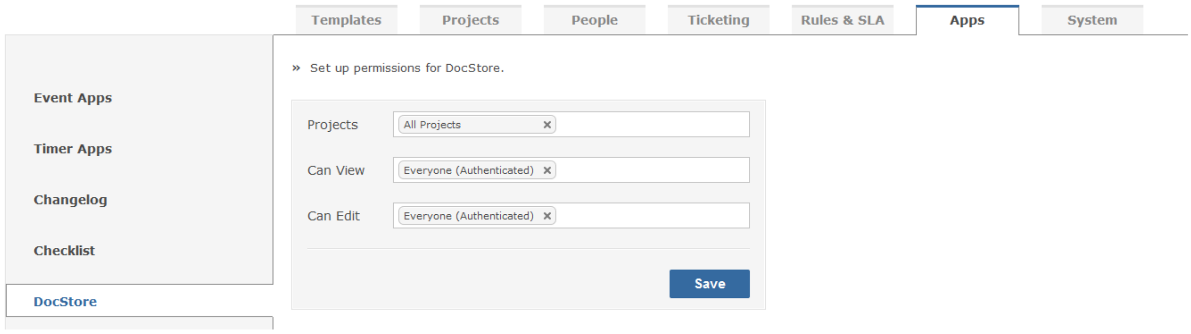
b) Stop your Gemini application pool and copy the ZIP into %Gemini%/App_Data /apps

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

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.

The app should now be available in Gemini. Now all you need to do is set the permissions and enable the app in the workspace settings.

Set app permissions.



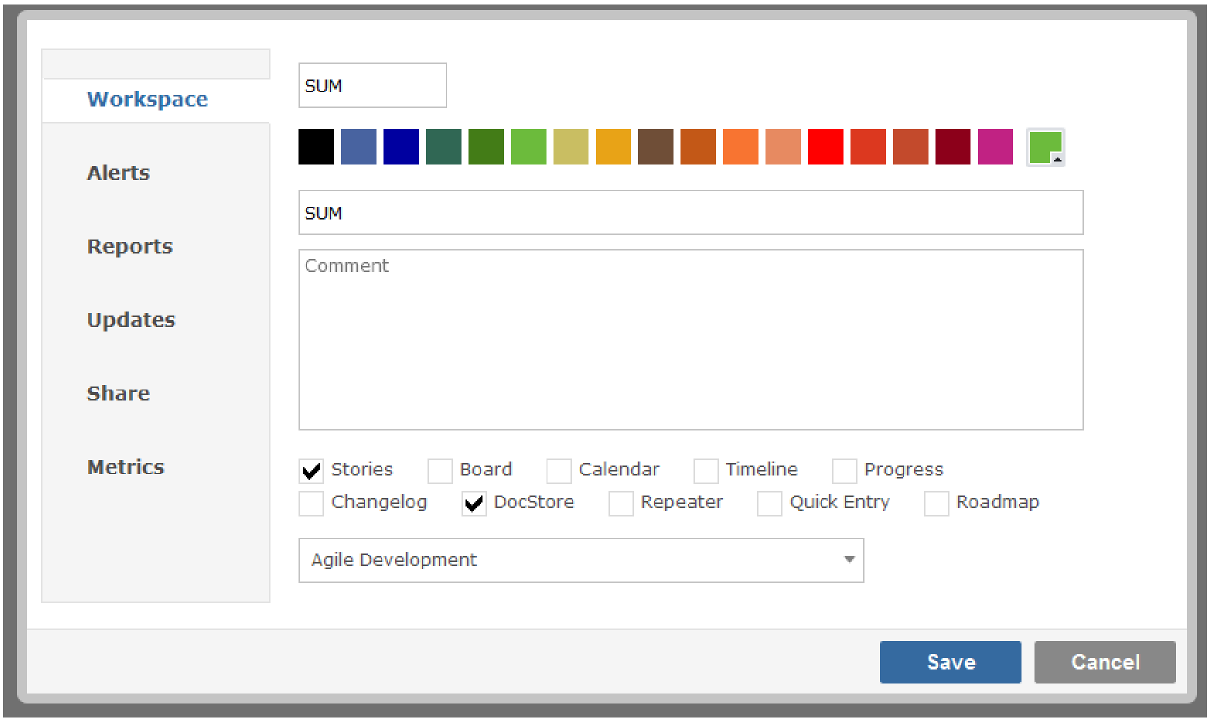
» Set up permissions for DocStore.

Projects: All Projects ×

Can View: Everyone (Authenticated) ×

Can Edit: Everyone (Authenticated) ×

Edit your workspace and enable the app.



SUM

SUM

Comment

☑ Stories □ Board □ Calendar □ Timeline □ Progress
□ Changelog ☑ DocStore □ Repeater □ Quick Entry □ Roadmap

Agile Development

10. References Examples

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

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