



PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

Debugging server-side scripts

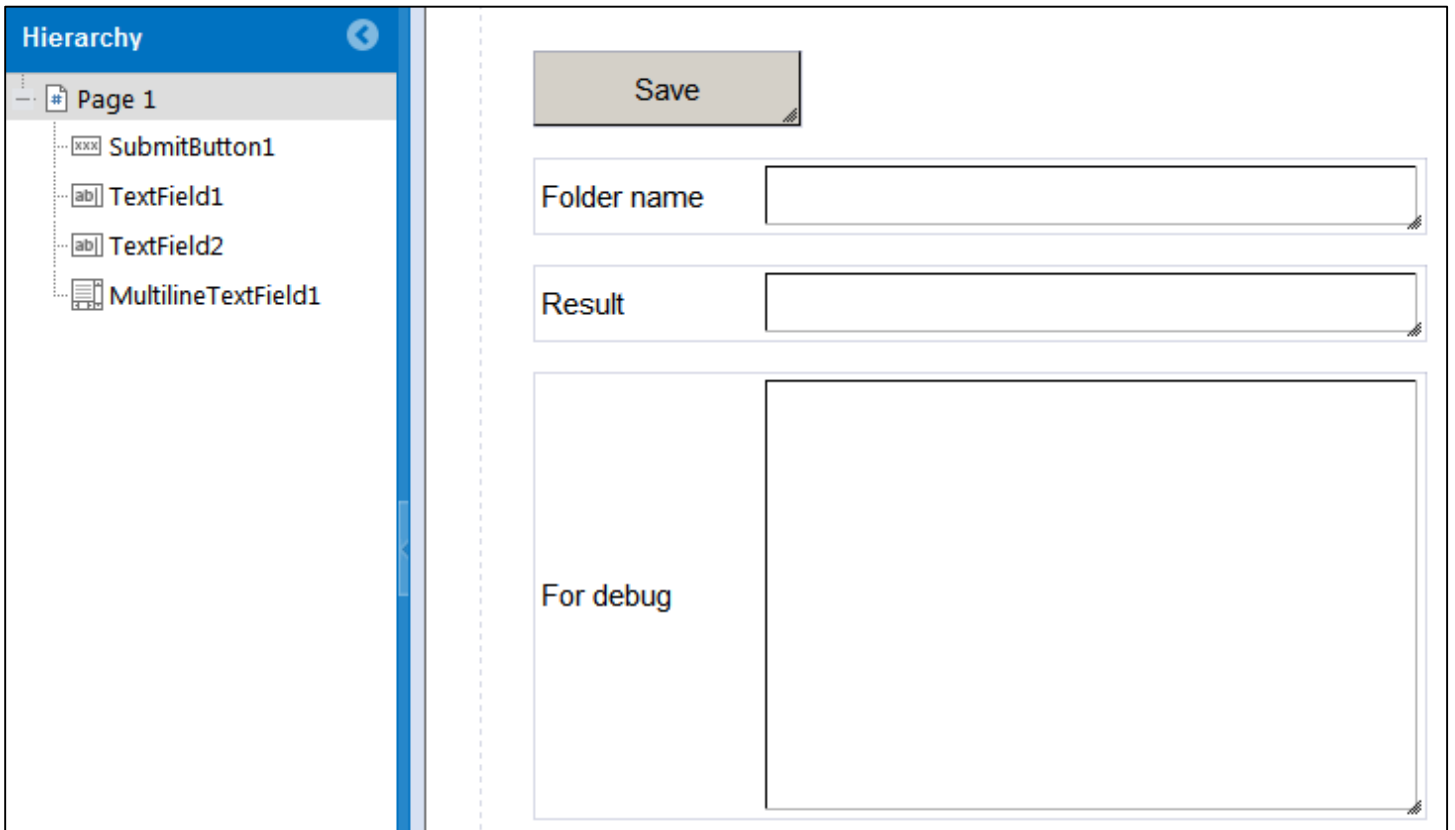
Contents

Preparing function	Error! Bookmark not defined.
Calling function from buttons	Error! Bookmark not defined.
Runtime results	Error! Bookmark not defined.

This guide will show how to debug server-side (C#) scripts in PDF templates. Following scenario based on the script, which will create folder in SharePoint 2013 library.

Template

Step 1. Prepare template



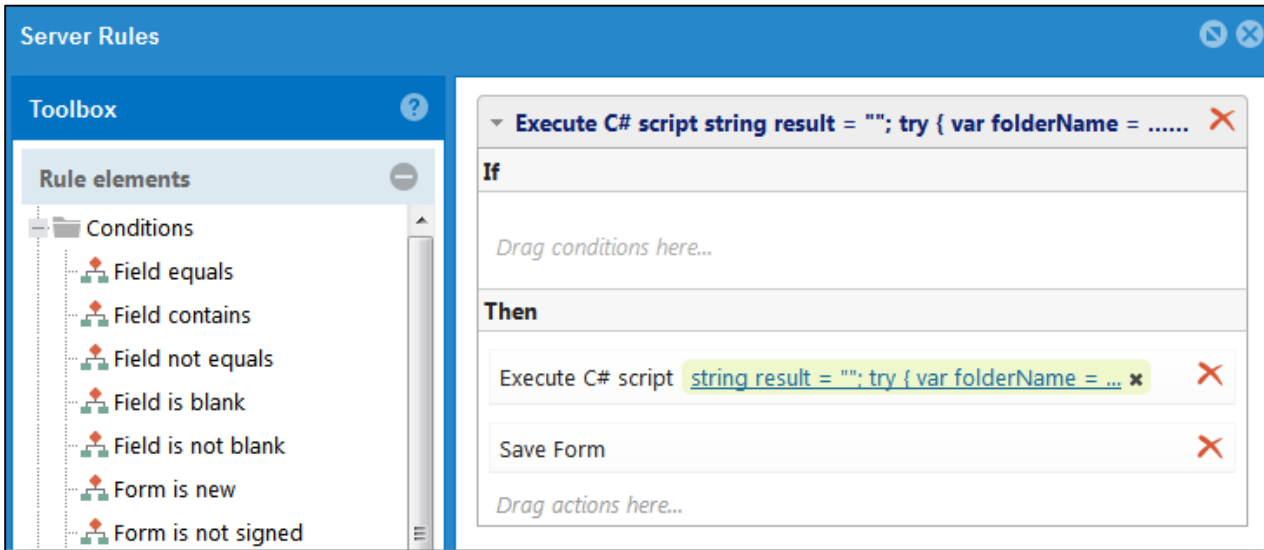
The screenshot shows a PDF form template editor. On the left, a 'Hierarchy' pane lists the controls: Page 1, SubmitButton1, TextField1, TextField2, and MultilineTextField1. On the right, the form layout includes a 'Save' button, a 'Folder name' text field, a 'Result' text field, and a large 'For debug' area.

This template has two text fields, multiline text field and submit button:

- ▣ TextField1 – “Folder name” used to give name for new folder;
- ▣ TextField2 – “Result” used to display result of folder creation script work;
- ▣ MultilineTextField1 – used to display errors in scripts.

Script with a bug

Step 1. Add script to Form Submit (**PDF Form tools** → **Developer** → **Form Submit**)



Add **“Execute script”** action before **“Save Form”** action and place inside it following code:

```
string result = "";
try
{
    var folderName = data.resolveNode("TextField1").value;
    SPList list = currentWeb.Lists["LibraryName"]; //this is name which is not assigned to any
    library on the site so error could be caught
    SPLListItem folder = list.AddItem(list.RootFolder.ServerRelativeUrl,
    SPFileSystemObjectType.Folder, folderName);
    folder.Update();
    result = "Folder has been created successfully";
}
catch (Exception ex)
{
    data.resolveNode("MultilineTextField1").value = ex.ToString();
    result = "Failed to create folder";
}
finally
{
    data.resolveNode("TextField2").value = result;
}
```

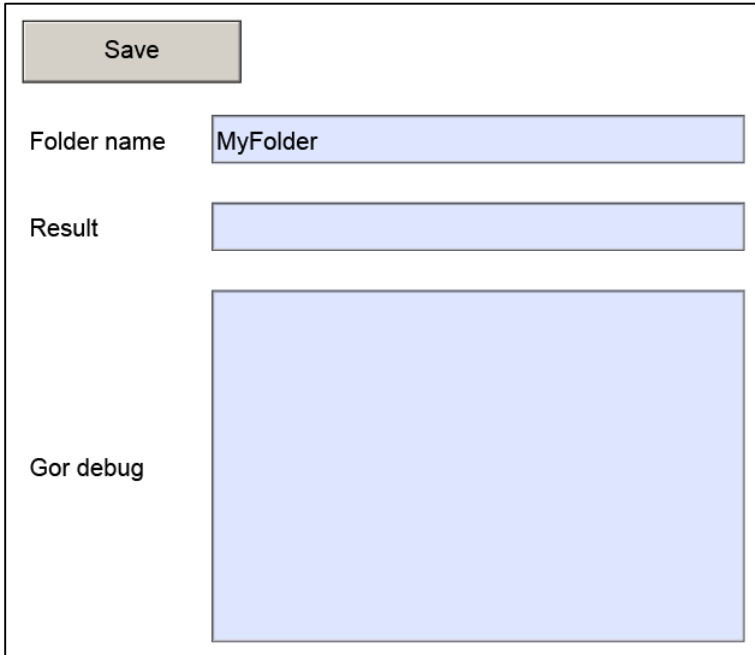
This code will take value from **“TextField1”** and use it as name for folder created in **“LibraryName”**. In order to catch errors in code **“try-catch-finally”** construction was added. It will catch any error in the code in try {}. This construction allows catching bugs, logical mistakes, not the misprints. Caught error can be used in catch {} section, where (Exception ex) is declaration of **“ex”** variable, which will be used to store error. Finally {} – section of code which will be ran no matter code have any error or not.

Step 2. Deploy template

In this guide for deployment was used library “Manuals”.

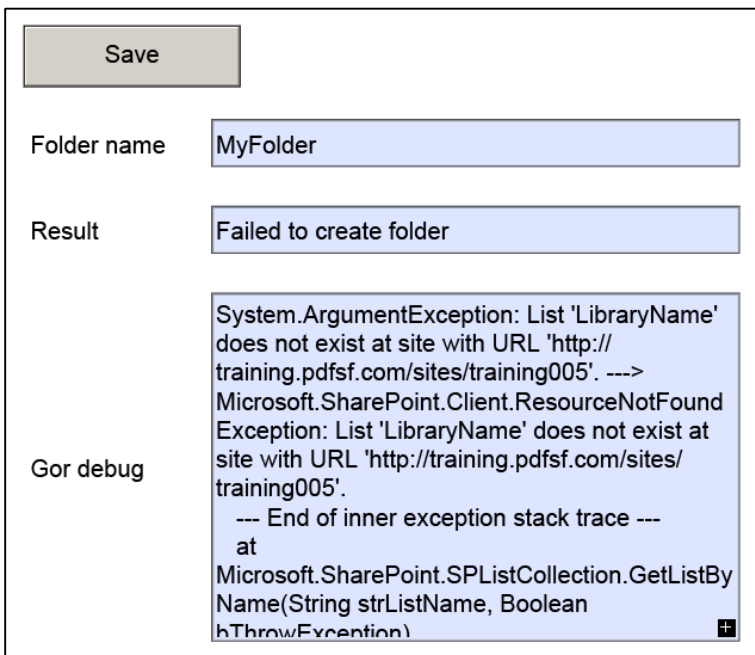
Step 3. Create new form and fill it

Fill “Folder name” and click on save button.



The screenshot shows a form with a 'Save' button at the top left. Below it, there is a 'Folder name' field containing the text 'MyFolder'. Underneath that is a 'Result' field, which is currently empty. At the bottom is a 'Gor debug' field, also empty.

Step 4. Open form again



The screenshot shows the same form after clicking the 'Save' button. The 'Result' field now contains the text 'Failed to create folder'. The 'Gor debug' field contains a detailed error message: 'System.ArgumentException: List 'LibraryName' does not exist at site with URL 'http://training.pdfsf.com/sites/training005'. ---> Microsoft.SharePoint.Client.ResourceNotFoundException: List 'LibraryName' does not exist at site with URL 'http://training.pdfsf.com/sites/training005'. --- End of inner exception stack trace --- at Microsoft.SharePoint.SPListCollection.GetListByName(String strListName, Boolean bThrowException)'. There is a small '+' icon at the bottom right of the debug field.

As the result you will see, that in multiline text field “**For debug**” appeared error message, which says, that there is no list with name “LibraryName”. In addition, text field “**Result**” is showing data which was added in finally {} section.

Script without bug

Step 1. Change library name in script

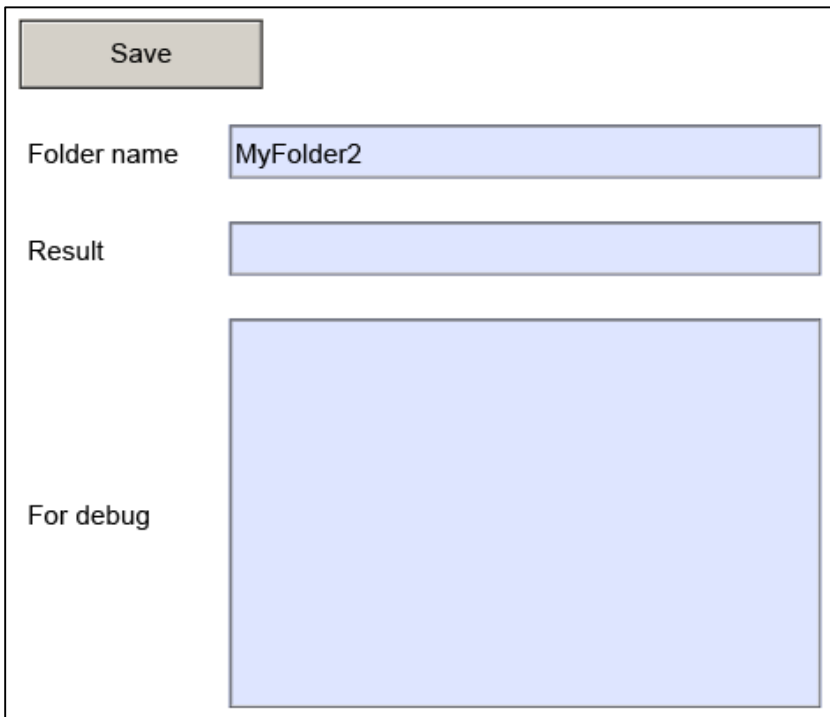
Change library name from script in step 2 for any existing library (in this example – same as for template deployment: “Manuals”):

```
SPList list = currentWeb.Lists["Manuals"]; //here must be existing library
```

Step 2. Deploy template again

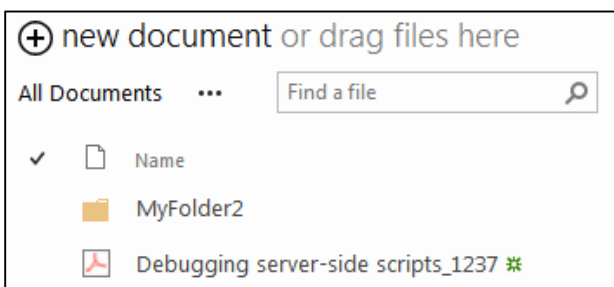
Step 3. Create new form

Fill “Folder name” and click on save button.



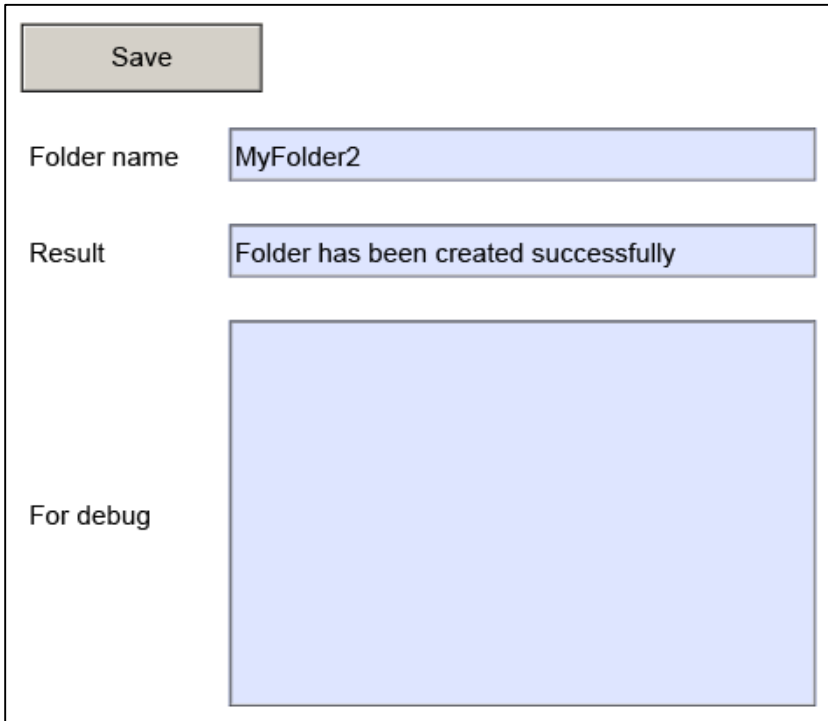
The screenshot shows a form with a 'Save' button at the top left. Below it is a 'Folder name' field containing 'MyFolder2'. Underneath is a 'Result' field, and at the bottom is a large 'For debug' area.

Step 4. Check library



Since in the code it was selected to create new folder in the same library where form is stored, this library has both items – newly created folder “MyFolder 2” and previously saved PDF form.

Step 5. Open form again



The screenshot shows a form interface with the following elements:

- A grey button labeled "Save" at the top left.
- A label "Folder name" followed by a text input field containing "MyFolder2".
- A label "Result" followed by a text output field containing "Folder has been created successfully".
- A label "For debug" followed by a large, empty light blue rectangular area.

Script worked without any errors, "Result" has confirmation, that folder was created.