

How to develop a Not Permission

- 1. What is the problem?
- 2. What is your idea to solve the problem?
- 3. What is the input needed for your plugin?
- 4. What is the output and expected outcome of your plugin?
- 5. Is there any resources/API that can be reuse?
- 6. Prepare your development environment
- 7. Just code it!
 - a. Extending the abstract class of a plugin type
 - b. Implement all the abstract methods
 - c. Manage the dependency libraries of your plugin
 - d. Make your plugin internationalization (i18n) ready
 - e. Register your plugin to Felix Framework
 - f. Build it and testing
- 8. Take a step further, share it or sell it

In this tutorial, we will following the [guideline of developing a plugin](#) to develop our Not Permission plugin. Please also refer to the very first tutorial [How to develop a Bean Shell Hash Variable](#) for more details steps.

1. What is the problem?

I want to configure the userview permission or form permission to not in groups, department or etc.

2. What is your idea to solve the problem?

We can develop a [Userview Permission/ Form Permission Plugin](#) to reverse the result of other permission plugins.

3. What is the input needed for your plugin?

To develop a Not Permission plugin, we can consider to provide the following as input.

1. Permission: To configure a Permission plugin to get the result for reverse it.
2. Must be Logged In User: The result will only valid if and only if the user is also a logged in user.

4. What is the output and expected outcome of your plugin?

The reverse value of the configured permission plugin.

5. Is there any resources/API that can be reuse?

You can refer to [How to develop a Gantt Chart Userview Menu](#) on how to reuse a existing plugin.

6. Prepare your development environment

We need to always have our Joget Workflow Source Code ready and builded by following [this guideline](#).

The following of this tutorial is prepared with a Macbook Pro and Joget Source Code version 5.0.0. Please refer to [Guideline for Developing a Plugin](#) for other platform command.

Let said our folder directory as following.

```
- Home
- joget
  - plugins
  - jw-community
    -5.0.1
```

The "plugins" directory is the folder we will create and store all our plugins and the "jw-community" directory is where the Joget Workflow Source code stored.

Run the following command to create a maven project in "plugins" directory.

```
cd joget/plugins/  
~/joget/jw-community/5.0.1/wflow-plugin-archetype/create-plugin.sh org.joget.tutorial not_permission 5.0.1
```

Then, the shell script will ask us to key in a version for your plugin and ask us for confirmation before generate the maven project.

```
Define value for property 'version': 1.0-SNAPSHOT: : 5.0.0  
[INFO] Using property: package = org.joget.tutorial  
Confirm properties configuration:  
groupId: org.joget.tutorial  
artifactId: not_permission  
version: 5.0.0  
package: org.joget.tutorial  
Y: : y
```

We should get "BUILD SUCCESS" message shown in our terminal and a "not_permission" folder created in "plugins" folder.

Open the maven project with your favour IDE. I will be using [NetBeans](#).

7. Just code it!

a. Extending the abstract class of a plugin type

Create a "NotPermission" class under "org.joget.tutorial" package. Then, extend the class with `org.joget.apps.userview.model.UserviewPermission` abstract class and implement `org.joget.apps.form.model.FormPermission` interface. Please refer to [Userview Permission/ Form Permission Plugin](#).

b. Implement all the abstract methods

As usual, we have to implement all the abstract methods. We will using `AppPluginUtil.getMessage` method to support i18n and using constant variable `MESSAGE_PATH` for message resource bundle directory.

Implementation of all basic abstract methods

```
package org.joget.tutorial;

import org.joget.apps.app.service.AppPluginUtil;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.form.model.FormPermission;
import org.joget.apps.userview.model.UserviewPermission;

public class NotPermission extends UserviewPermission implements FormPermission {

    private final static String MESSAGE_PATH = "messages/NotPermission";

    public String getName() {
        return "Not Permission";
    }

    public String getVersion() {
        return "5.0.0";
    }

    public String getDescription() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.tutorial.NotPermission.pluginDesc", getClassName(),
MESSAGE_PATH);
    }

    public String getLabel() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.tutorial.NotPermission.pluginLabel", getClassName(),
MESSAGE_PATH);
    }

    public String getClassName() {
        return getClass().getName();
    }

    public String getPropertyOptions() {
        return AppUtil.readPluginResource(getClassName(), "/properties/notPermission.json", null, true,
MESSAGE_PATH);
    }

    @Override
    public boolean isAuthorize() {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

Then, we have to do a UI for admin user to provide inputs for our plugin. In `getPropertyOptions` method, we already specify our [Plugin Properties Options](#) definition file is locate at `/properties/notPermission.json`. Let us create a directory `"resources/properties"` under `"not_permission/src/main"` directory. After create the directory, create a file named `"notPermission.json"` in the `"properties"` folder.

In the properties definition options file, we will need to provide options as below. Please note that we can use `"@@message.key@@"` syntax to support i18n in our properties options.

```
[{
  title : '@@userview.notpermission.config@@',
  properties : [{
    name : 'permission',
    label : '@@userview.notpermission.permission@@',
    type : 'elementselect',
    options_ajax : '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.userview.model.
UserviewPermission',
    url : '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
  },
  {
    name : 'loggedIn',
    label : '@@userview.notpermission.loggedIn@@',
    type : 'checkbox',
    value : 'true',
    options : [{
      value : 'true',
      label : ''
    }]
  }
]}]
```

After done the properties option to collect input, we can work on the main method of the plugin which is isAuthorize method.

```
@Override
public boolean isAuthorize() {
    boolean isAuthorize = false;
    try {
        if ("true".equals(getPropertyString("loggedIn")) && WorkflowUtil.isCurrentUserAnonymous()) {
            return false;
        }

        //get the binder
        Object permissionData = getProperty("permission");
        if (permissionData != null && permissionData instanceof Map) {
            Map pMap = (Map) permissionData;
            if (pMap != null && pMap.containsKey("className") && !pMap.get("className").toString().
isEmpty()) {
                PluginManager pluginManager = (PluginManager) AppUtil.getApplicationContext().getBean
("pluginManager");
                UserviewPermission permission = (UserviewPermission) pluginManager.getPlugin(pMap.get
("className").toString());
                if (permission != null) {
                    Map pProps = (Map) pMap.get("properties");
                    permission.setProperties(pProps);
                    permission.setCurrentUser(getCurrentUser());
                    permission.setRequestParameters(getRequestParameters());

                    isAuthorize = !permission.isAuthorize();
                }
            }
        }
    } catch (Exception e) {
        LogUtil.error(getClassName(), e, "");
    }
    return isAuthorize;
}
```

c. Manage the dependency libraries of your plugin

There are no extra dependency for this plugin.

d. Make your plugin internationalization (i18n) ready

We are using i18n message key in getLabel and getDescription method. We also used i18n message key in our properties options definition as well. So, we will need to create a message resource bundle properties file for our plugin. Create directory "resources/messages" under "not_permission/src/main" directory. Then, create a "NotPermission.properties" file in the folder. In the properties file, let add all the message keys and its label as below.

```
org.joget.tutorial.NotPermission.pluginLabel=Not Permission
org.joget.tutorial.NotPermission.pluginDesc=Used to reverse the result of other permission plugin
userview.notpermission.config=Configure Not Permission
userview.notpermission.permission=Permission
userview.notpermission.loggedIn=Must be Logged In User
```

e. Register your plugin to Felix Framework

We will have to register our plugin class in Activator class (Auto generated in the same class package) to tell Felix Framework that this is a plugin.

```
public void start(BundleContext context) {
    registrationList = new ArrayList<ServiceRegistration>();
    //Register plugin here
    registrationList.add(context.registerService(NotPermission.class.getName(), new NotPermission(), null));
}
```

f. Build it and testing

Let build our plugin. Once the building process is done, we will found a "not_permission-5.0.0.jar" file is created under "not_permission/target" directory.

Then, let upload the plugin jar to [Manage Plugins](#). After upload the jar file, double check the plugin is uploaded and activated correctly.

Filter by Type Userview Permission

<input type="checkbox"/>	Plugin Name	Plugin Description	Plugin Version
<input type="checkbox"/>	Is Admin		5.0.0
<input type="checkbox"/>	Is Anonymous		5.0.0
<input type="checkbox"/>	Bean Shell Script		5.0.0
<input type="checkbox"/>	Department		5.0.0
<input type="checkbox"/>	Group		5.0.0
<input type="checkbox"/>	Logged In User		5.0.0
<input type="checkbox"/>	Not Permission	Used to reverse the result of other permission plugin	5.0.0
<input type="checkbox"/>	Organization		5.0.0
<input type="checkbox"/>	Is Sub Level Userview	Disallow access to sub level	5.0.0
<input type="checkbox"/>	User		5.0.0

10
Page 1 of 1
Displaying 1 to 10 of 10 items

Uninstall Plugin

Let us open an userview and change one of the category permission to Not Permission. We will want the current user not in a "Managers" group.

Settings

Preview

Save

Set Permission

Set Permission > Permission Type (Not Permission)

Hide From Menu

☐

Permission Type

Not Permission

✕

▼

< Prev

Next >

OK

Cancel

Settings

Preview

Save

Configure Not Permission

Set Permission > Configure Not Permission > Configure Userview Permission (Group)

Permission

Group

✕

▼

Must be Logged In User

☒

< Prev

Next >

OK

Cancel

Settings

Privatay

Save

Configure Userview Permission (Group)

Set Permission > [Configure Not Permission](#) > [Configure Userview Permission \(Group\)](#)

Select Organization

Select Groups

Managers ✕

< Prev

Next >

OK

Cancel

After done the configuration and save the userview. Let us test it. First, check the admin user is not in "Managers" group.

Edit Group

Delete Group

Group Detail

Group ID

G-001

Group Name

Managers

Group Description

Organization

User List

Search

<input type="checkbox"/>	Username	First Name	Last Name	Email	
<input type="checkbox"/>	clark	Clark	Kent		
<input type="checkbox"/>	david	David	Cain		
<input type="checkbox"/>	jack	Jack	Drake		
<input type="checkbox"/>	roy	Roy	Harper		
<input type="checkbox"/>	sasha	Sasha	Bordeaux		
<input type="checkbox"/>	tina	Tina	Magee		

10

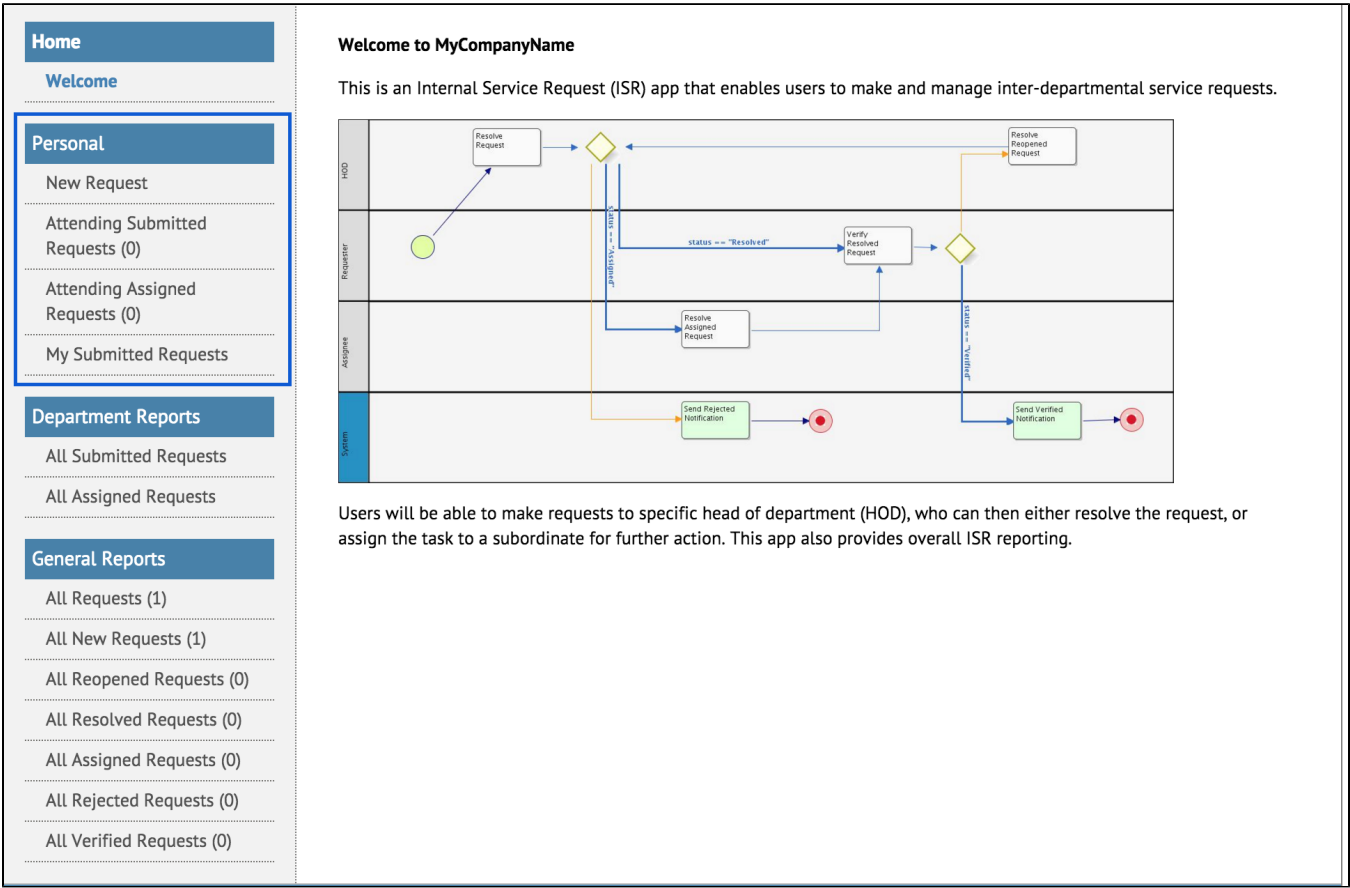
Page 1 of 1

Displaying 1 to 6 of 6 items

Assign Users

Unassign Users

Check the useview, the "Personal" category configured with not in "Managers" group is shown correctly.



Now, assign the "admin" user to "Managers" group.

Setup Users

1 Setup Organization

2 Setup Groups

3 Setup Users

Edit Group

Delete Group

Group Detail

Group ID

G-001

Group Name

Managers

Group Description

Organization

User List

Search

<input type="checkbox"/>	Username	First Name	Last Name	Email
<input type="checkbox"/>	admin	Admin	Admin	owen@joget.org
<input type="checkbox"/>	clark	Clark	Kent	
<input type="checkbox"/>	david	David	Cain	
<input type="checkbox"/>	jack	Jack	Drake	
<input type="checkbox"/>	roy	Roy	Harper	
<input type="checkbox"/>	sasha	Sasha	Bordeaux	
<input type="checkbox"/>	tina	Tina	Magee	

10

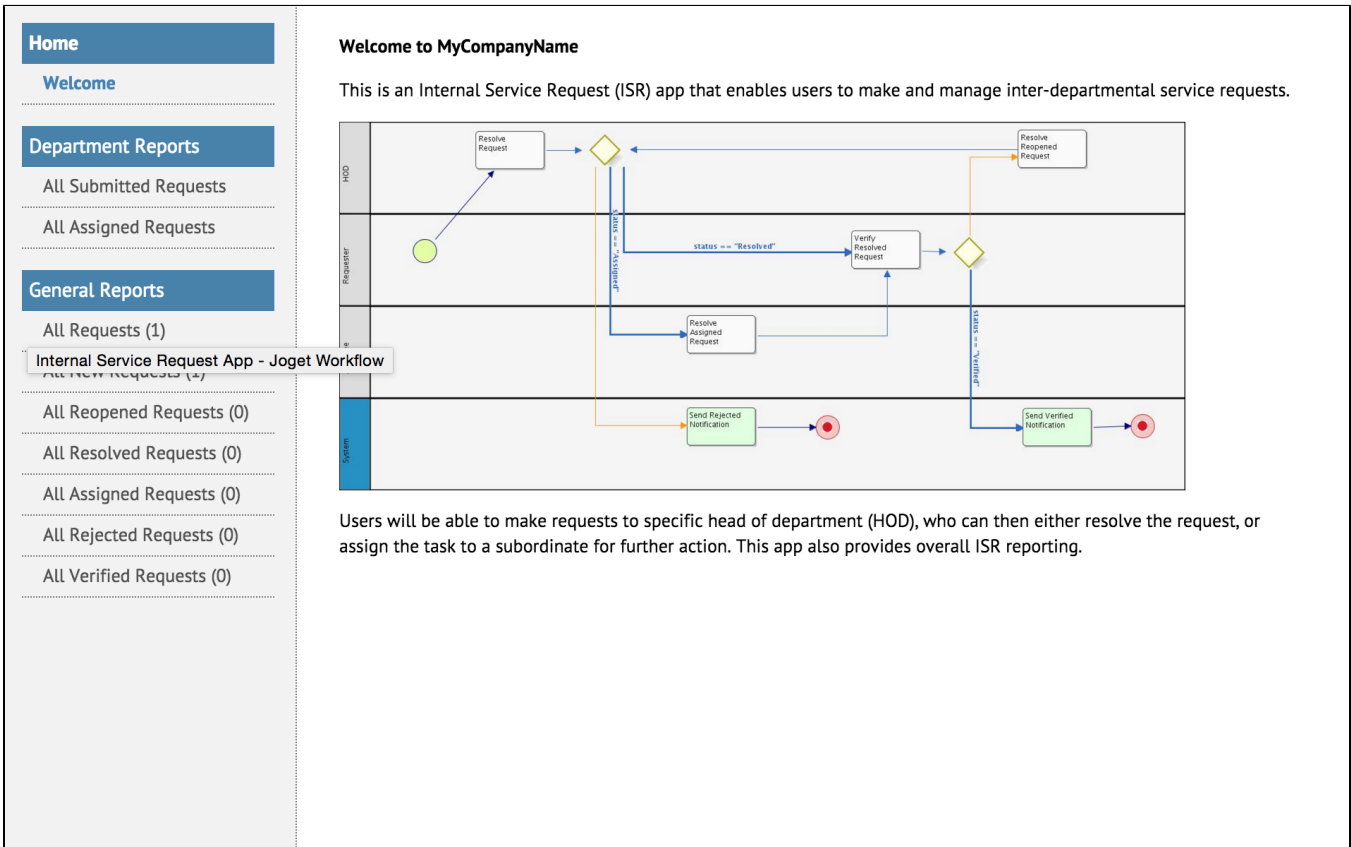
Page 1 of 1

Displaying 1 to 7 of 7 items

Assign Users

Unassign Users

The "Personal" category is now disappeared.



8. Take a step further, share it or sell it

You can download the source code from [not_permission.zip](#).

To download the ready-to-use plugin jar, please find it in <http://marketplace.joget.org/>.