**Permission Steps**

**1) Define static permission in Solution**  For creating permission we have to create static permission variable in permission file.  
Declaration of variable name should be in proper format.

**Path**: File(Core=>Authorization=>PermissionNames)

**Example**:

public const string Pages\_Users = "Pages.Users";  
  
Here in above example Pages.Users is the name of Permission   
and Pages\_Users is variable name.

**2) Create Parent Permission.**Permission can be create by using static permission variable. **Path:**

**(**Core=>Authorization=> AbpProjectsAuthorizationProvider)

**Example**:  
context.CreatePermission(PermissionNames.Pages\_Users, L("Users"));

Here In above Example CreatePermission is method to create permission   
CreatePermission has to parameter first parameter is for name of permission i.e static permission variable and another is for text for that permission and text is define in localization file.

**3) If User wants to create child permission for User then**

If user wants to create a child permission for any module:  
Example   
If user wants to create parent and child both permission for User Module then parent permission is Users and child permission are create, edit and delete operation.  
  
To Create child Permission please refer below example

**Example**:

var User = context.CreatePermission(PermissionNames.Pages\_Users, L("Users"));

User.CreateChildPermission(PermissionNames.Pages\_Userschild, L("userchild"));

The User Variable is used for creating Parent Permission and by using user we have to create child Permission.

**4) To hide link from menu based on permission.**

For hiding link in menu we have to just add Permission name in parameter as shown in below example

**Path:** (Web=>App=>main=>Views=>layout=>sidebar-nav.js)   
Example:

createMenuItem(App.localize("Employee"), "**Pages.Users**", "people", "users"),  
  
Here in above Line CreateMenuItem is Method that is used for creating a link in Menu.  
In Creating a menu first parameter is name of link: link name ,  
2nd Parameter is for Defining the permission, 3rd is icon menu and 4 parameter is for state name for that module.  
  
  
**5)To Restrict the URL**

**Path** (Web=>App=>main=>app.js)

For restricting url we have to check in routing file that current has permission or not with of **abp.auth.hasPermission('Pages.Users'))**

Abp.auth.Has persmission check that current login user has that permission or not.

Example:  
if (abp.auth.hasPermission('Pages.Users')) {

$stateProvider

.state('users', {

url: '/employee',

templateUrl: '/App/Main/views/users/index.cshtml',

menu: 'Employee' //Matches to name of 'Users' menu in AbpProjectsNavigationProvider

});

$urlRouterProvider.otherwise('/');

}

If user has that permission then it will redirect to that page otherwise it will redirect to home page.

**6) Add Permission in Service File.  
  
For securing the API we have to add Permission tag in Service file.  
If we want to to authorized the service File without permission then we have to just add [**AbpAuthorize] above service File and if User wants to authorized with specific permission then add [AbpAuthorize(PermissionNames)]  
 **Path** (Application solution )  
Example

**[AbpAuthorize(PermissionNames.Pages\_Users)]**

**7) Show/Hide div based on Permission in View Page**.  
Define variable in js file that returns true/false if current user has right for that permission as given below:  
 Define Permission in .js file.  
 Example: vm.userpermission = abp.auth.isGranted('Pages.Users);  
  
 <div ng-if=" vm.userpermission ==true">

<div class="header">

<h2>

@L("Hosting")

</h2></div>

**8) If we want to add Permission base where clause in service File**

**Declare interface for using permission**private readonly PermissionChecker \_permissionChecker  
  
Add Parameter in Construction of that service method  
and Assign that interface in Construction method as given below:

\_permissionChecker = permissionChecker;  
  
With the help of \_permissionChecker we check if current user has that permission or not as given below

var result=’’;  
if (\_permissionChecker.IsGranted("Pages.Projects.ProjectManager"))

{

result = result.Where(x => x.projectmanagerId == UserId);

}