Control Report Execution based on user roles
By: Ashish Harbhajanka

Introduction

While I assume most of us are already aware and are using this feature to control report execution based on roles assigned to a user but I would still like to share the step by step of accomplishing the same as it could be helpful for some newbies or other professionals who are not aware of this feature.

This kind of requirement can be met by making use of two concepts :

 a) Executing Report using a Springboard Icon
 b) Use Expression Language to control accessibility of report.

In this example we will demonstrate the following:

 1) All Employees who have the seeded Employee Roles will be able to access a report (PersonAbsenceAccrualDetail)
 2) Only Employees who have Line Manager role assigned will have access to (AccrualDetail)

For this example, we will use a pre-existing report. The high-level steps to perform this activity include:
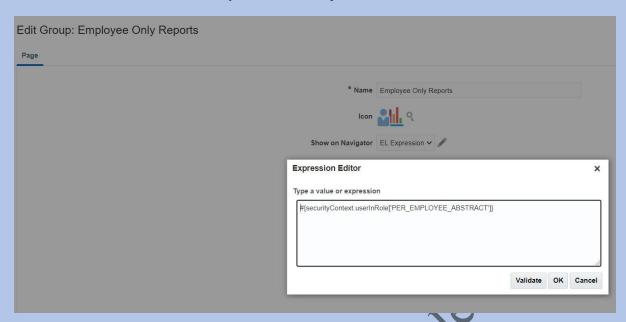
 1) Create a New Group Under Tools -> Configuration and use a EL Expression on the Visible property to ensure the Group is only visible to users having "PER_EMPLOYEE_ABSTRACT"
 2) Create a New Page Entry under the newly created group and use a EL Expression on the Visible property to ensure the Group is only visible to users having "PER_EMPLOYEE_ABSTRACT"
 3) Create a New Page Entry under the newly created group and use a EL Expression on the Visible property to ensure the Group is only visible to users having "PER_LINE_MANAGER_ABSTRACT"

Creating "Employee Only Reports" Group

One would need to enable a Sandbox and then use below navigation:

Navigator->Configuration->Structure->Create->Create Group

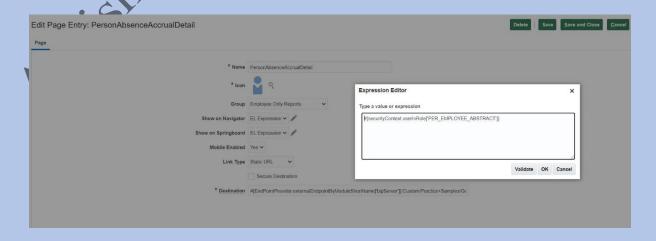| Attribute Name	Attribute Value
* Name	Employee Only Reports
Show on Navigator	EL Expression #{securityContext.userInRole['PER_LINE_MANAGER_ABSTRACT']}

Creating "PersonAbsenceAccrualDetail" Page

We will create a page entry which will be used to call "PersonAccrualDetailReport" which should be visible to all employees.

Properties of the Page will be as follows:

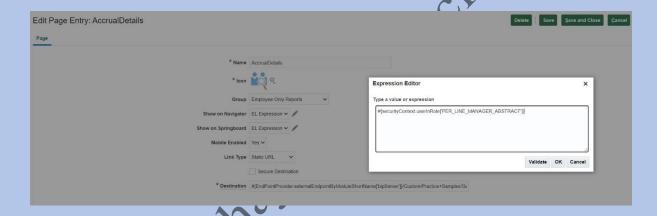| Attribute Name	Attribute Value
*Name	PersonAbsenceAccrualDetails
*Group	Employee Only Reports
Show on Navigator	EL Expression #{securityContext.userInRole['PER_EMPLOYEE_ABSTRACT']}
Show on Springboard	EL Expression #{securityContext.userInRole['PER_EMPLOYEE_ABSTRACT']}
Mobile Enabled	Yes
Link Type	Static URL
*Destination	#{EndPointProvider.externalEndpointByModuleShortName['bipServer']}/Custom/Practice+Samples/GetPersonAccrualData.xdo?_xmode=2

Creating "AccrualDetail" Page

We will create a page entry which will be used to call "AccrualDetail" which should be visible to line managers.

Properties of the Page will be as follows:

Attribute Name	Attribute Value
*Name	AccrualDetails
*Group	Employee Only Reports
Show on Navigator	EL Expression #{securityContext.userInRole['PER_LINE_MANAGER_ABSTRACT']}
Show on Springboard	EL Expression #{securityContext.userInRole['PER_LINE_MANAGER_ABSTRACT']}
Mobile Enabled	Yes
Link Type	Static URL
*Destination	#{EndPointProvider.externalEndpointByModuleShortName['bipServer']}/Custom/Practice+Samples/GetPersonAccrualData.xdo?_xmode=2

Once the setup is complete we can publish the sandbox.

Verification: Employee Login

We will login to application as SANDY.KIM who is an employee and check results

Verification: Line Manager Login

We will login to application as AMY.MARLIN who is both an employee and a line manager and check results

Conclusion

So this is how we can control accessibility and execution of reports based on roles assigned to user. Here, Amy.Marlin is both a Line Manager and Employee having both the roles and hence she is able to access and execute both the reports while Sandy.Kim is an employee and not a manager and so she can only access the report designed to be used by employees.

The same concept can be extended to specific users too such that a report will be accessible to only one user and not to another.