# Abstract

Identity and Access Management (*IAM*)

Here is short explanation of how we manage Users and their Permissions in Finmars Platform.

Precondition: Read a IAM Keywords explanations

So, when User joins to Space, Member record will be created, all the rules and policies will be applied to Member

By default Member has no Roles, Groups, Access Policies, so that means, by default Member **has no access to anything**

If Member has Admin flag (is_admin=True) he will access to everything. IAM Engine Access Policies will be ignored

## How its all works?

Finmars is Platform, it has a REST API interface, that works with HTTP protocol (you could read more about HTTP, REST API and Requests in public sources)

so when User open a Browser and Logs In into Finmars, Finmars Frontend App will start making Requests

So each Request will address some Endpoint e.g. *api/v1/portfolios/portfolio*

So, when Finmars receive incoming request, server will basically get (in simple terms):

member: user_a,
method: GET,
endpoint: *api/v1/portfolios/portfolio*

What happen next?

Finmars Fetch all Roles,Groups,Access Policies assigned to that Member

Then it will take all Access Policies from assigned Roles, Groups and make a one huge list with unique Access Policies

Then Evaluation of Policies will occur

it means we transform

method: GET,
endpoint: *api/v1/portfolios/portfolio*

to

finmars:Portfolio:list (each endpoint has own Viewset, eg PortfolioViewset serves *api/v1/portfolios/portfolio)* each action/method will tranfom to action (e.g. list)

And then we will find if any AccessPolicy allows me finmars:Portfolio:list

If nothing is found, then Permission Denied (403) will be raised (with explanation of why)

## Resource Evaluation

Next step, if User has permissions to make a certain action, but he requested specific resource, then we will check if he has access to that resource:

```
"Resource": "*",
```

This will grant access to all objects of that Entity

```
 "Resource": [
        "frn:finmars:iam:resourcegroup:portfolio_group_a"
],
```

This will grant access to objects that assigned to that **Resource Group**

```
 "Resource": [
        "frn:finmars:portfolios:portfolio:bonds-portfolio"
],
```

This will grant access only to certain Portfolio

so, even if you have access to Entity, you still able to get Permission Denied (403) because you

requested an object that you have no access to

## Ownership

Each objects in finmars has *owner*. It means when member creates an objects, he is now a owner of that object. Ownership grants a member full control over an object, even if IAM Access Policies are not granting it.

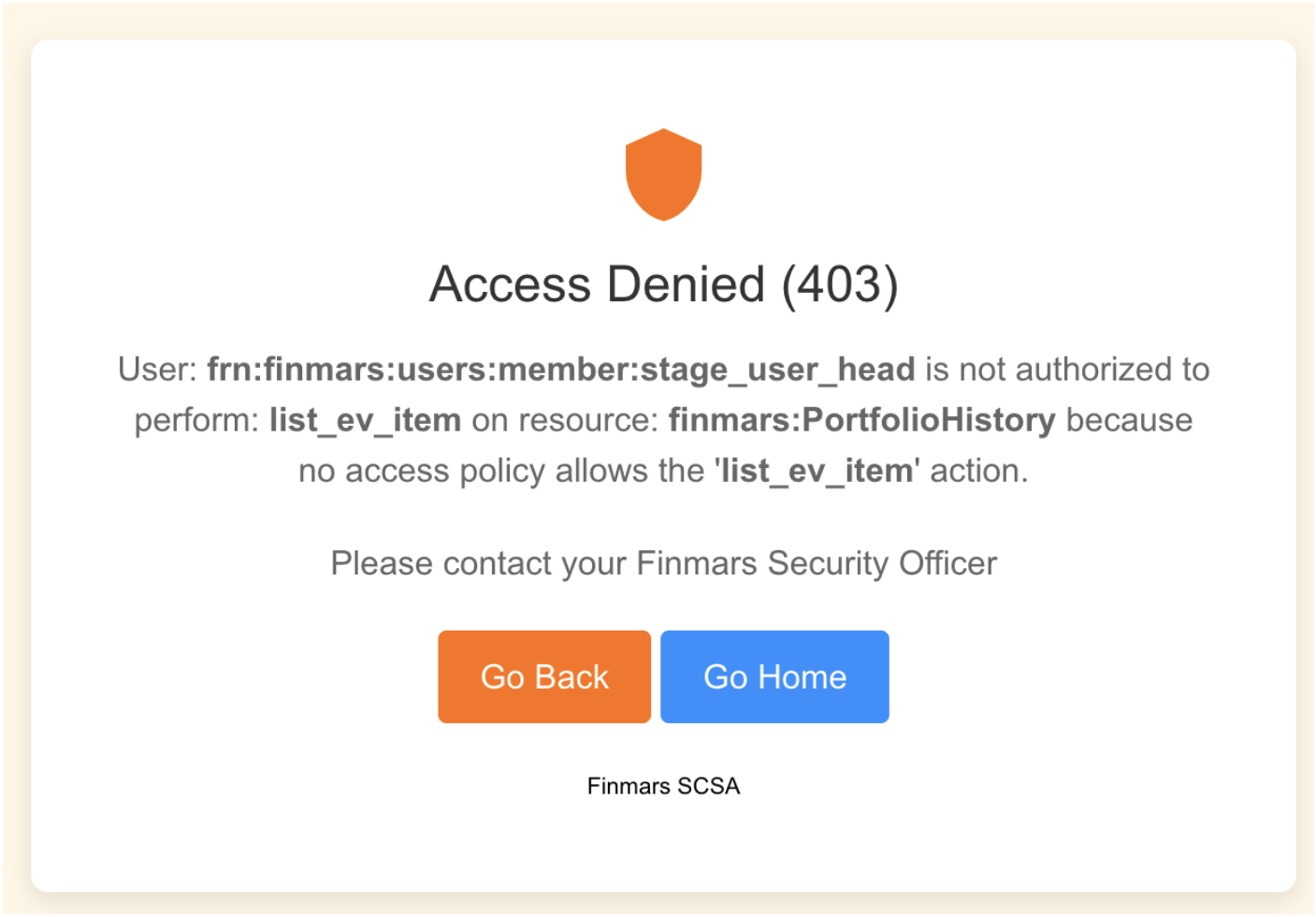
## Public Objects

So, in certain scenarios you normally has no access to objects, but some Transactions (e.g. Transfer) force you to see other objects. E.g. you have access to Portfolio 1 (From) but has no permissions to Portfolio 2 (To), in that case able to see Portfolio 2 but only its Public Name (public_name)

So, on any object you have always 3 public properties: **id, user_code, public_name**

General Recommendation is consult with your Client and ask how he want to see his permissions and public name, in that example instead of Portfolio 2, public name should be "Private Portfolio"

If User has no access to certain action or resource he will be redircted to 403 Page




Revision #8
Created 2024-10-31 12:09:20 UTC by Sergei Zhitenev
Updated 2024-10-31 14:48:34 UTC by Sergei Zhitenev