Keywords

IAM - Identity and Access Management (*IAM*)

User - user itself, when you sign up in FInmars, your User will be created

Space - Finmars Installation, like a workspace

Member - its a User inside a Space, like a membership, so one **User** can be a Member in different **Spaces**

Admin Member - member who has checkbox Is Admin (is_admin=True), it means, he will absolutely **ignores** all IAM Engine Access Policies

Owner - each object has *owner* attribute which links to Member, if you are owner of object you will Full Access to it (you able to see, modify it even if IAM is not allow it to you)

Access Policy - Document in JSON format that contains rules and conditions about Permission

Role - Entity that can be attached to Member, also **Access Policies** can be attached to Role

Group - Entity similar to Role, only difference that to Group you could attach other **Roles**

user_code - a property that has asci lowercase notation (like a variable in programming language), so its human readable ID

public_name - a property that contains a Verbose Name (Human Readable Name). So, this property is always visible / accessible even if Access Policies are restrict it.

FRN - Finmars Resource Name - basically a human-readable unique identifier for Finmars Platform

e.g. frn:finmars:iam:resourcegroup:portfolio_group_a

```
'''
    converts:
        frn:finmars:iam:resourcegroup:portfolio_group_a
    to
        {
         "type": "frn",
```

```
          "service": "finmars"
          "app_label": iam",
          "model": "resourcegroup",
          "user_code": "portfolio_group_a"
         }
'''
```

So in other words its even more advanced version of **user_code**

Access Policy Structure

```
{
  "Version": "2023-01-01",
  "Statement": [
    {
      "Action": [
        "finmars:Portfolio:create",
        "finmars:Portfolio:update",
        "finmars:Portfolio:destroy",
        "finmars:Portfolio:bulk_delete",
        "finmars:Portfolio:bulk_restore",
        "finmars:Portfolio:delete_preview",
        "finmars:Portfolio:list_ev_group",
        "finmars:Portfolio:list_ev_item",
        "finmars:Portfolio:list"
      ],
      "Effect": "Allow",
      "Resource": [
        "frn:finmars:iam:resourcegroup:portfolio_group_a"
      ],
      "Principal": "*"
    }
  ]
}
```

So, each policy can have a list of **Statement**

Statement - a document that has list of **Action, Effect, Resource, Principal**

Action - basically its a String FRN like format that identifiy an Action in Finmars Platform (e.g.

REST API Endpoint)

Effect - its either **Allow** or **Deny**. Deny will forbid access to that action/resource

Resource - could be *** (allows access to all objects)** or list of FRN identifiers

Principal - identifier to Actor of that Policy, normally its * any user,role,group. But if you need to be extra sure that this policy will work only for specific member, you could pass a member FRN

Resource Group - an entity who serves like a container of objects, so basically you could link any object in Finmars to some Resource Group. Idea is simple, if you have 1000 portfolios, and you need to grant access only to 100 portfolios, you need to declare all 100 portfolios FRN in Resource property in Access Policy. Alternative approach is just to put FRN of one Resource Group and link all 100 portfolios to that Resource Group via Finmars Web Interface

---

Revision #4
Created 2024-10-28 09:36:08 UTC by Sergei Zhitenev
Updated 2024-10-31 13:30:31 UTC by Sergei Zhitenev