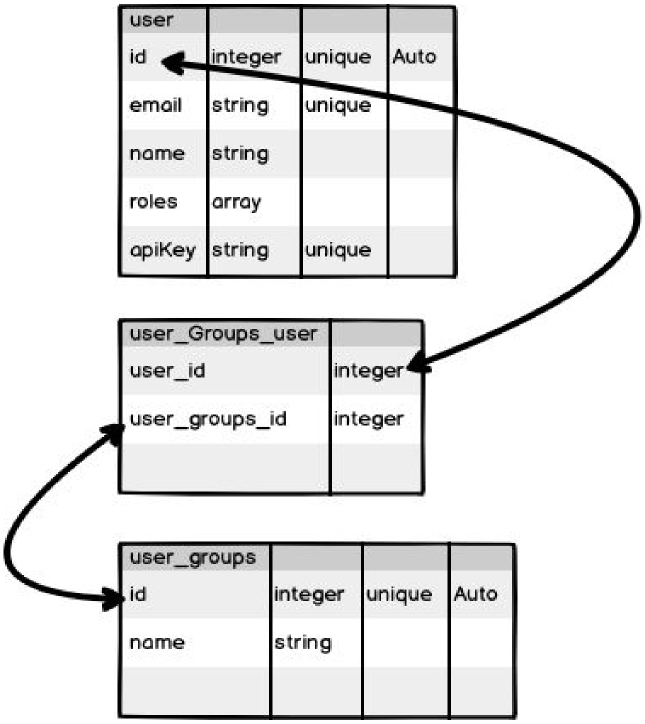
Test Project - user management system

https://github.com/tanvir1674/test-internation.git

Overview

In suggested 4 hours, I focused on building api for all the stories proposed. Built them in symfony 4 and tried to showcase the following in the system.



user			
id	integer	unique	Auto
email	string	unique	
name	string		
roles	array		
apiKey	string	unique	

user_Groups_user	
user_id	integer
user_groups_id	integer

user_groups			
id	integer	unique	Auto
name	string		

1.Symfony Security Component

2.Symfony validation constraint

3. REST Api design

4. The Serializer Component

5.API Authentication using Guard Authenticator

6.basic functional test using phpUnit

Major files worked on

/src/Controller/UserApiController.php
/src/Controller/UserGroupsController.php
/tests/Controller/UserApiControllerTest.php

/src/Entity/User.php and
/src/Entity/UserGroups.php

In the simple process, created two entity User and UserGroups with ManyToMany relation as one user can be assigned to multiple groups. Then created a controller with methods supporting REST api calls. Used Assert for basic property validation, isGranted to controll Admin access and Guard to facilitate token based api authentication.

API Documentaion

GET	/api/users	Retrieves the collection of user resources.

Sample response

```
{
   "users": [
      {
         "email": "ddddddd@test.com",
         "roles": [
            "ROLE_ADMIN",
            "ROLE_USER"
         ],
         "name": "tanvir alam",
         "userGroups": [
            {
               "name": "players"
            },
            {
               "name": "swimmers"
            }
         ]
      },
   ]
}
```

POST	/api/users	Creates a user resource.

Sample json input

```
{
      "email": "ddddddd@test.com",
      "roles": [
         "ROLE_ADMIN",
         "ROLE_USER"
      ],
      "name": "tanvir alam",
      "userGroups": [
         {
            "name": "players"
         },
         {
            "name": "swimmers"
         }
      ]
   }
```

GET	/api/users/{id} Retrieves a user resource.

	Sample Output response
	{
	  "user": {
	    "email": "b@test.com",
	    "roles": [
	      "ROLE_ADMIN",
	      "ROLE_USER"
	    ],
	    "name": "jewel",
	    "userGroups": []
	  }
	}

DELETE	/api/users/{id} Remove a user resource.
PUT	/api/users/{id} Updates the user resource.

	Sample json input
	{
	    "email": "ddddddd@test.com",
	    "roles": [
	      "ROLE_ADMIN",
	      "ROLE_USER"
	    ],
	    "name": "tanvir alam",
	    "userGroups": [
	      {
	        "name": "players"
	      },
	      {
	        "name": "swimmers"
	      }
	    ]
	  }

GET	/api/groups Retrieves the collection of group resources.

	Sample response
	{
	  "groups": [
	    {
	      "name": "players"
	    },
	    {
	      "name": "swimmers"
	    }
	  ]
	}

POST	/api/groups Creates a group resource.

	Sample json input
	{
	  "name": "swimmers"
	}

GET	/api/groups/{id} Retrieves a group resource.

	Sample Output response
	{
	  "name": "swimmers"
	}

DELETE	/api/groups/{id} Remove a group resource.
PUT	/api/groups/{id} Updates the group resource.