



User Management

When we click Administration/Users menu, we enter the user management page:

Users

Manage users and permissions.

Export to excel

Create new user

Search...

Show advanced filters

| Actions | User name | Name | Surname | Roles | Email address | Email confirm | Active | Last login time | Creation time |
|---------|-------------------|-------------|----------|-------|----------------------------------|---------------|--------|-----------------|---------------|
| Actions | admin | admin | admin | Admin | admin@acme.com | Yes | Yes | 11/22/2017 | 11/22/2017 |
| Actions | amin.maalouf | Amin | Maalouf | User | amin.maalouf@yahoo.com | Yes | Yes | | 11/22/2017 |
| Actions | andrew.andrewus | Andrew | Andrewus | User | andrew.andrewus@outlook.com | Yes | Yes | | 11/22/2017 |
| Actions | audrey.naulin | Audrey | Naulin | User | audrey.naulin@live.com | Yes | No | | 11/22/2017 |
| Actions | biff.tannen | Biff | Tannen | User | biff.tannen@hotmail.com | Yes | Yes | | 11/22/2017 |
| Actions | christophe.grange | Christophe | Grange | User | christophe.grange@aspnetzero.com | Yes | Yes | | 11/22/2017 |
| Actions | christopher.nolan | Christopher | Nolan | User | christopher.nolan@aspnetzero.com | Yes | Yes | | 11/22/2017 |
| Actions | clarice.starling | Clarice | Starling | User | clarice.starling@live.com | Yes | Yes | | 11/22/2017 |
| Actions | david.wells | David | Wells | User | david.wells@yahoo.com | Yes | Yes | | 11/22/2017 |
| Actions | forrest.gump | Forrest | Gump | User | forrest.gump@hotmail.com | Yes | Yes | | 11/22/2017 |

Showing 1 to 10 of 23 entries | Show 10 entries

« < 1 2 3 > »

Users are people who can **login** to the application and perform some operations based on their **permissions**.

User class represents a user. User class **can be extended** by adding new properties.

UserManager is used to perform domain logic, **UserAppService** is used to perform application logic for users.

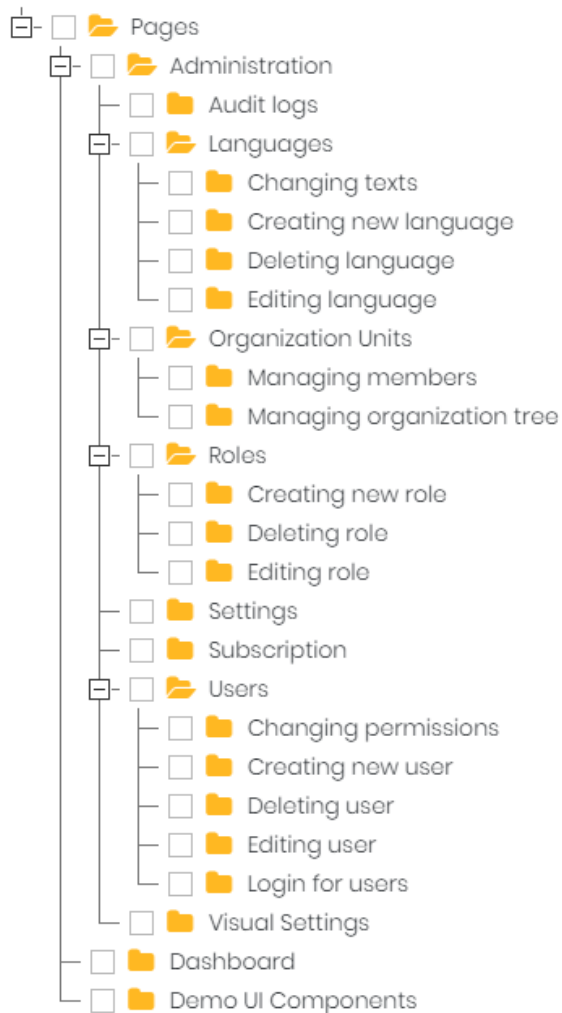
A user can have zero or more **roles**. If a user has more than one role, the user inherits union of permissions of all these roles. Also, we can set **user-specific permission**. A user specific permission setting overrides role settings for this permission. For example if a user has Accountant role and Accountant role has "editing invoice" permission, we can take away "editing invoice" permission from a specific user if we want to. The opposite is also true, so we can grant invoice editing permission for a specific user even if none of the roles of that user don't have "editing invoice" permission.

A screenshot of user permission dialog:



ASP.NET CORE ANGULAR

DOCUMENTS



If you are changing your own permissions, you may need to refresh page (F5) to take effect of permission changes on your own screen!

Reset special permissions

Cancel

 Save

Feedback



Note that not all permissions shown in the figure above.

A dialog is used to create/edit a user:



ASP.NET CORE ANGULAR

DOCUMENTS

User informations

Roles **1**

Organization Units



Name

Christophe

Surname

Grange

Email address

christophe.grange@aspnetzero.com

Phone number

User name

christophe.grange

☐ Set random password.

Password

Password (repeat)

☐ Should change password on next login.

☐ Send activation email.

☒ Active

☐ Lockout enabled

Cancel

 Save

Feedback



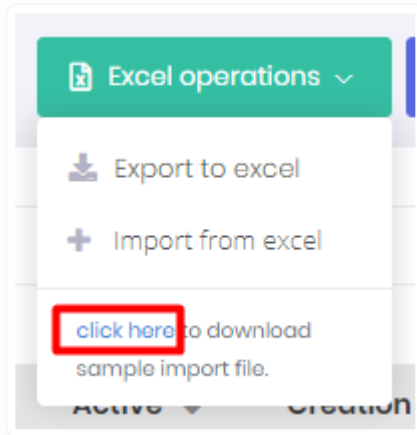
We can change user's **password**, make the user **active/passive** and so on... A user can have a **profile picture**. It can be changed by the user (See User Menu section). **Admin** user can not be deleted as a business rule. If you don't want to use admin, just make it inactive.



ASP.NET CORE ANGULAR

DOCUMENTS

User list can be downloaded as an Excel file. Also, new users can be imported from an Excel file. In order to import users from an Excel file, ASP.NET Zero requires a specific excel file format. You can download a sample import template on the user list by clicking the "click here" link under the "Excel operations" dropdown button.



When you import users from Excel, if there are errors while importing users to ASP.NET Zero's database, invalid users will be saved to an Excel file with the reason (validation error message or exception message) and the user who made the import operation will be notified via ASP.NET Zero's notification system. So, the user can click the notification and see the result Excel file, fix the validation errors and import the failed users again.

Feedback



User Unlock

If a user is locked out (had a specific count of failed access), user can be unlocked on the user list page. You can click "Actions" button on the user page and click "**Unlock**" item on the opened dropdown list.

Lockout options can be configured in the **IdentityRegistrar.cs** class under the "Identity" folder in the ***.Core** project. In the sample below, MaxFailedAccessAttempts is configured as 10, so if a user makes 10 failed login attempts in a row, the user will be locked out for a period of time.

```
return services.AddAbpIdentity<Tenant, User, Role>(options =>
{
    // other configurations
    options.Lockout.MaxFailedAccessAttempts = 10;
})
```

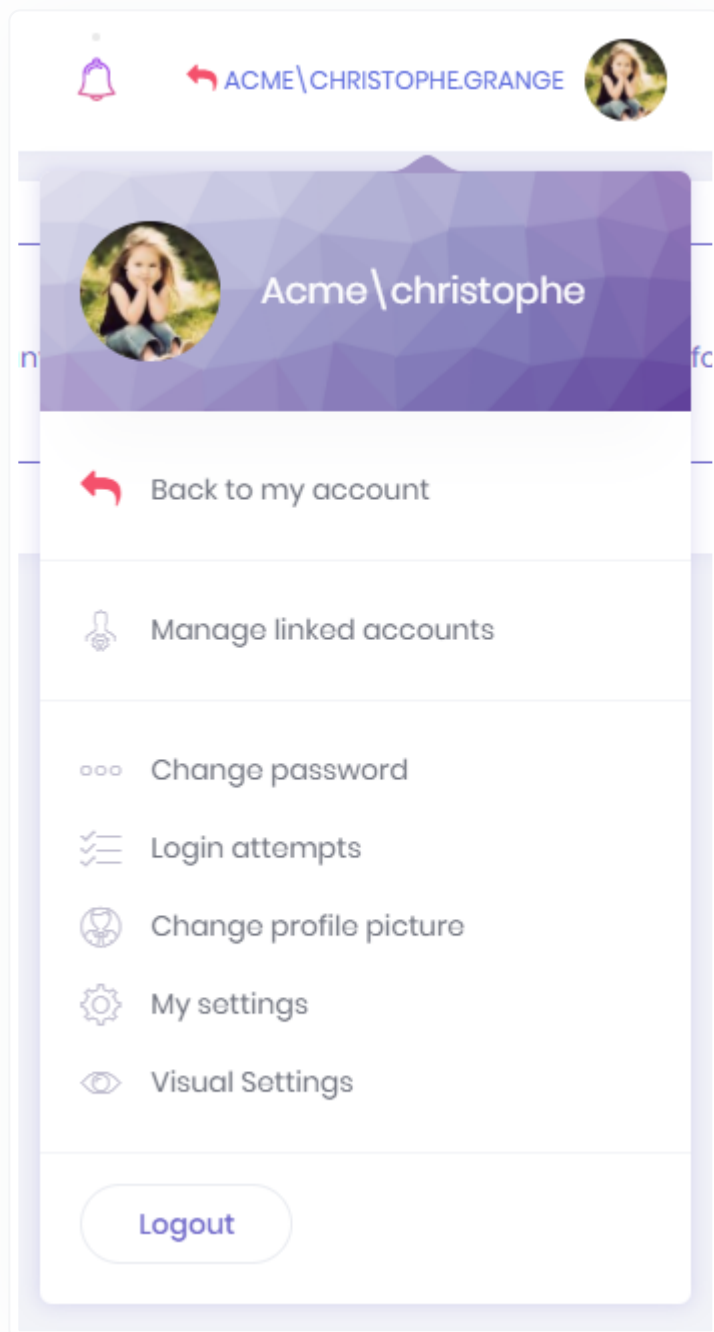
User Impersonation



ASP.NET CORE ANGULAR

DOCUMENTS

user, we are automatically redirected and logged in as this user. This is called **user impersonation**. When we impersonate a user, a **"back to my account"** option is added to the user profile menu:



Feedback



In an impersonated account, we can only perform operations allowed to that user. It means, everything **exactly** works as same as this user is logged in. The only difference is shown in audit logs which indicates that operations are performed by somebody else on behalf of the user.

Notice that also a **red 'back' icon** shown near to the user name to indicate that you are in an impersonated account.



ASP.NET CORE ANGULAR

DOCUMENTS

- [Language Management](#)

