Overview

The ‘Amb Licensing’ project is the project for generating cryptographically signed license files. License file contains product features and components (with features), optionally end user agreement file and cryptographic signature. The file is zipped as License.zip.

Technology&Architecture

The product is written in C# language. Running on .Net 9 framework. The product is comprised of 2 components: UI component and Api component. The components are installed and running as windows services. The Api component is implemented as Web Api service with minimal Api. Entity framework is used as database storage. Swagger is used as testing and documentation framework. Serilog is used for logging. Angular material is used as front-end part.

Installation

Before installation make backup copy of the services. Go to Amb.Licensing.Api and Aamb.Licensing.UI services, look at the properties, find the .exe path and go to that path in windows explorer. Copy all contents and paste it into some other newly created folder. Download installation zip from the site. Unzip it to some folder. Run install.cmd in admin mode.

If .net 9 is not installed you’ll be given a ;ink to download and install it. After the installation you’ll have to run installation script again.

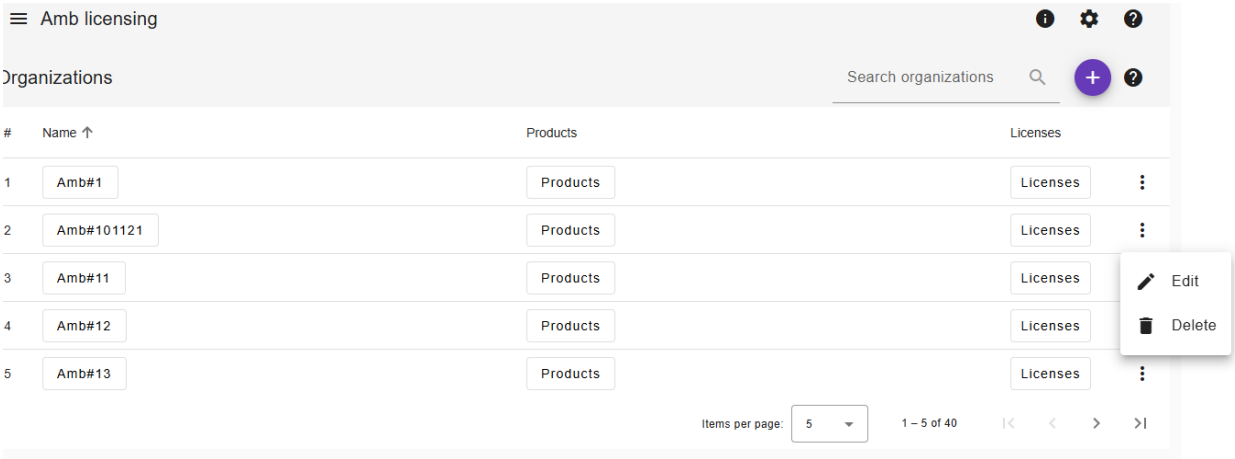
If amb.licensing.api service I installed, you’ll be given an option to reinstall it. If installing/reinstalling the service, you’ll be prompted to enter account/password for the account the service will be running under. Make sure the account has access to ms sql server database with db creator or admin rights. In case ms sql server is installed, you’ll be given an option to select sql instance to connect to. Also there will be an option to connect to sql server by custom connection string (for instance remote one). In case sql server is not installed, you’ll be given an option to connect by connection string or install sql server express by provided link. Then api service is installled and started.

Then it comes to UI service. If amb.licensing.ui service is installed, you’ll be given an option to reinstall it. If installing/reinstalling the service, you’ll be prompted to enter account/password for the account the service will be running under. Then UI service is installled and started. Since now the UI should be available at **http://localhost:8080** or http://[IP]:8080. If you use IP please create router port forwarding and firewall rules (speak to your devops engineer about how to do it)

The sample output is below.

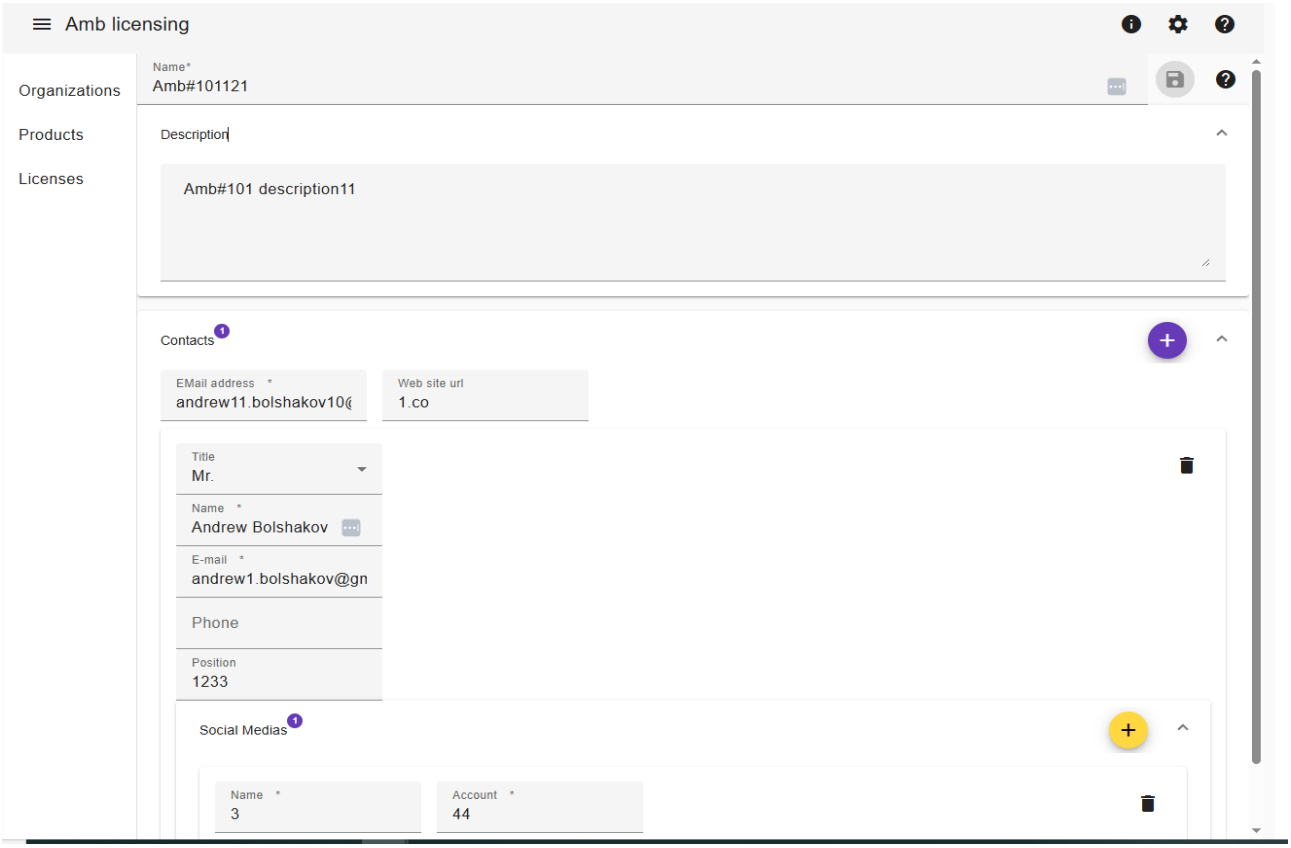
```
C:\WINDOWS\system32>powershell d:\Users\amb\Documents\source\repos\Amb\Licensing\install.ps1
No Api service found.
Install in C:\Program Files\Amb.Licensing.Api:
1) MSSQLSERVER
2) SQLEXPRESS
3) Connection string
Select Sql Server instance to use: 1
Data Source=localhost;Integrated Security=True;TrustServerCertificate=True;MultipleActiveResultSets=true:
Connection Successful!
Staring Api service...
Api service is updated and started.
No UI service found.
Install in C:\inetpub\wwwroot\Amb.Licensing.UI:
Staring UI service...
UI service is updated and started.
Press any key for exit...
```

Organizations



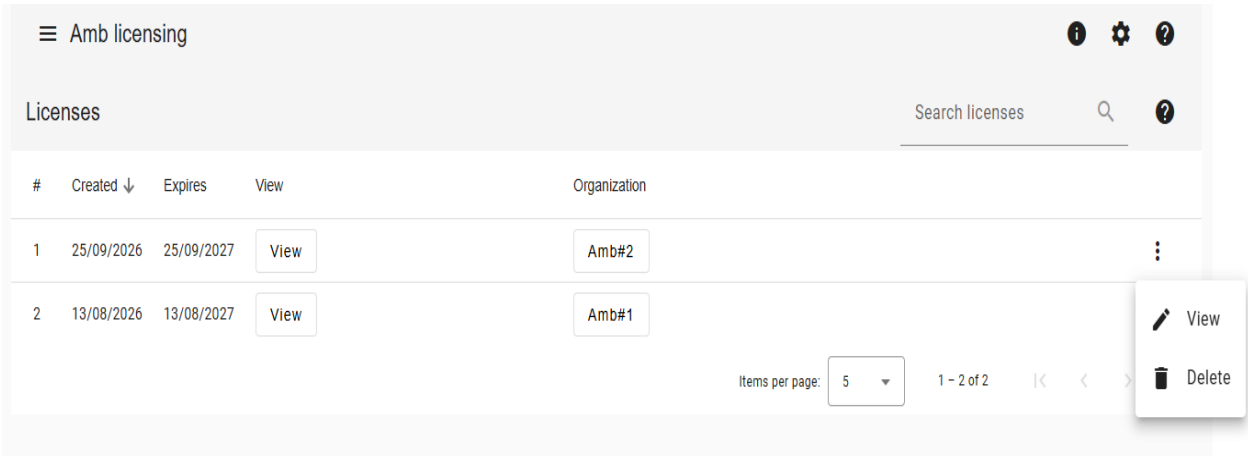
In this page we can see a list of available organizations. Using the paginator we can move between the pages and select how many rows to display in the page. Name column is clickable and lets us select the sort order (asc or desc). Column ‘#’ shows sequential numbers of rows. ‘Name’ link in each row points/navigates to particular organization. The link is also available in ‘three dots/more’ menu. The ‘Products’ column link in each row points/navigates to the product list. The ‘Licenses’ column link in each row points/navigates to the organization license list. In ‘three dots/more’ menu there is also an option to delete organization. Search box gives one an option to search/filter organizations by search value contained in the ‘Name’ , ‘Description’, ‘Contact name’, ‘Social media name’ and ‘Social media account’ fields. ‘New/Add’ button points/navigates to the page to create new organization.

Organization



In this page we can edit all properties/collections of particular organization. After editing some fields and in case the input values are valid, the ‘Save’ button is enabled and you can save your changes. The ‘Name’ field is required and should be at least 3 characters in length. The description field is optional. The ‘Email’ field in contacts section is required. ‘Web site url’ field is optional. We can add new contacts using ‘Add’ button. The required fields for the contact is ‘Name’ and ‘Email’. The other fields are optional. Optionally you can add social media links to each contact. ‘Name’ and ‘Account’ fields are required. You can of course delete social media links or the contact itself.

Licenses



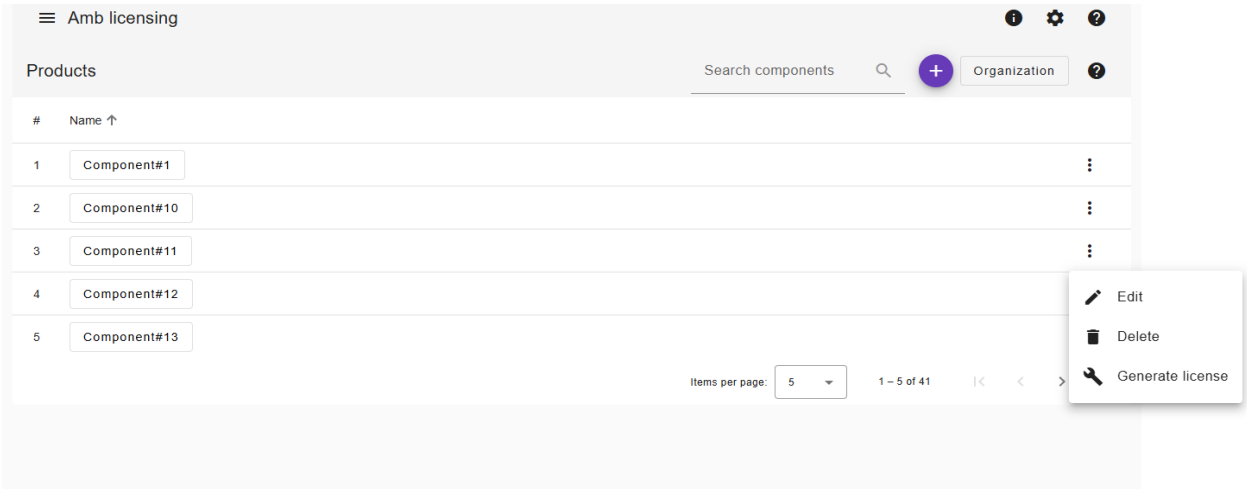
In this page we can see a list of available licenses. Using the paginator we can move between the pages and select how many rows to display in the page. Created and Expires columns are clickable and let us select the sort order (asc or desc). Column ‘#’ shows sequential numbers of rows. ‘View’

link in each row points/navigates to particular license. The link is also available in ‘three dots/more’ menu. The ‘Organization’ column link in each row points/navigates to the organization the license is issued to. In ‘three dots/more’ menu there is also an option to delete the license. Search box gives one an option to search/filter licenses by search value contained in the ‘Organization’ field.

License

This is a view to licensed product, components and features. All fields are readonly/disabled so that already issued license is immutable. The are ‘Created’ and ‘Expires’ fields. There is a button in the toolbar to delete the license. Also there is a button in the page to download the license (as zip file).

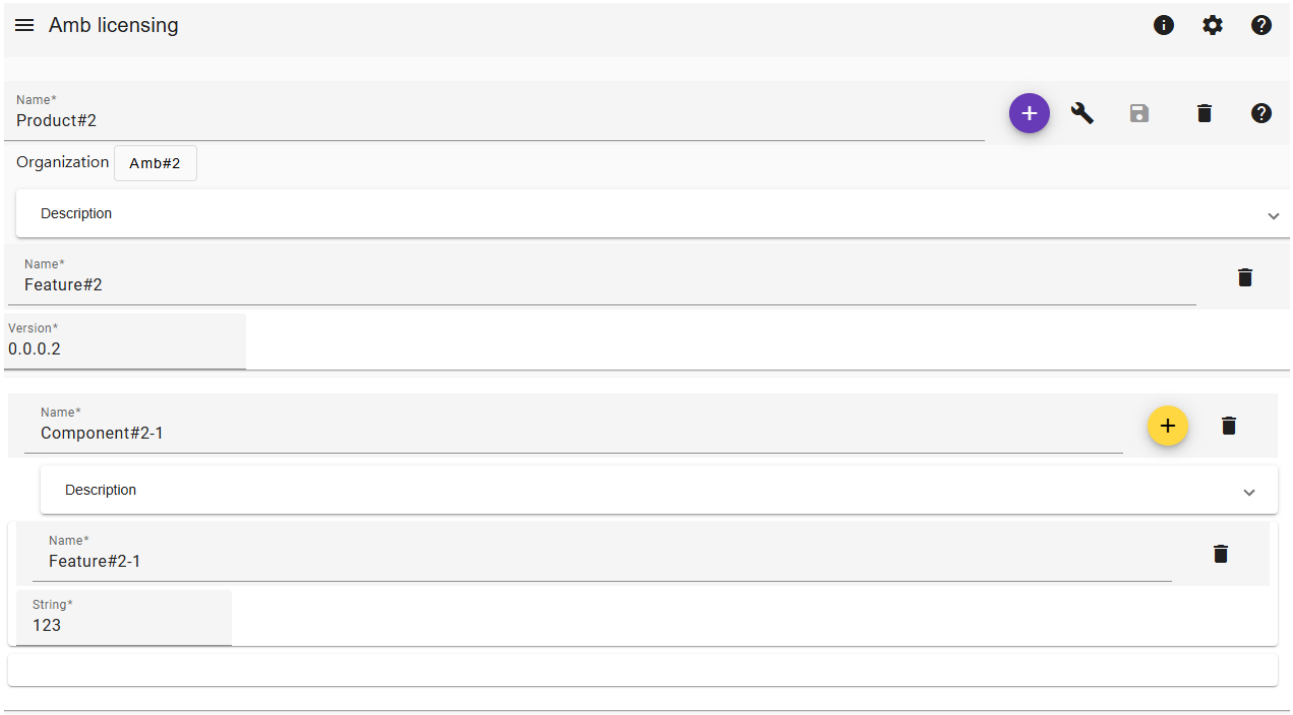
Products



In this page we can see a list of available products. Using the paginator we can move between the pages and select how many rows to display in the page. Name column is clickable and lets us select the sort order (asc or desc). Column ‘#’ shows sequential numbers of rows. ‘Name’ link in each row points/navigates to particular product. The link is also available in ‘three dots/more’ menu. In ‘three

dots/more’ menu there is also an option to delete the product. Search box gives one an option to search/filter products by search value contained in the ‘Name’, ‘Description’, ‘Feature Name’, component’s ‘Name’, ‘Description’, ‘Feature Name’ fields. ‘New/Add’ button points/navigates to the page to create new Product. If we came to this page from ‘Orhanization/Products’ link, we’ve got “Generate license” link in ‘three dots/more’ menu.

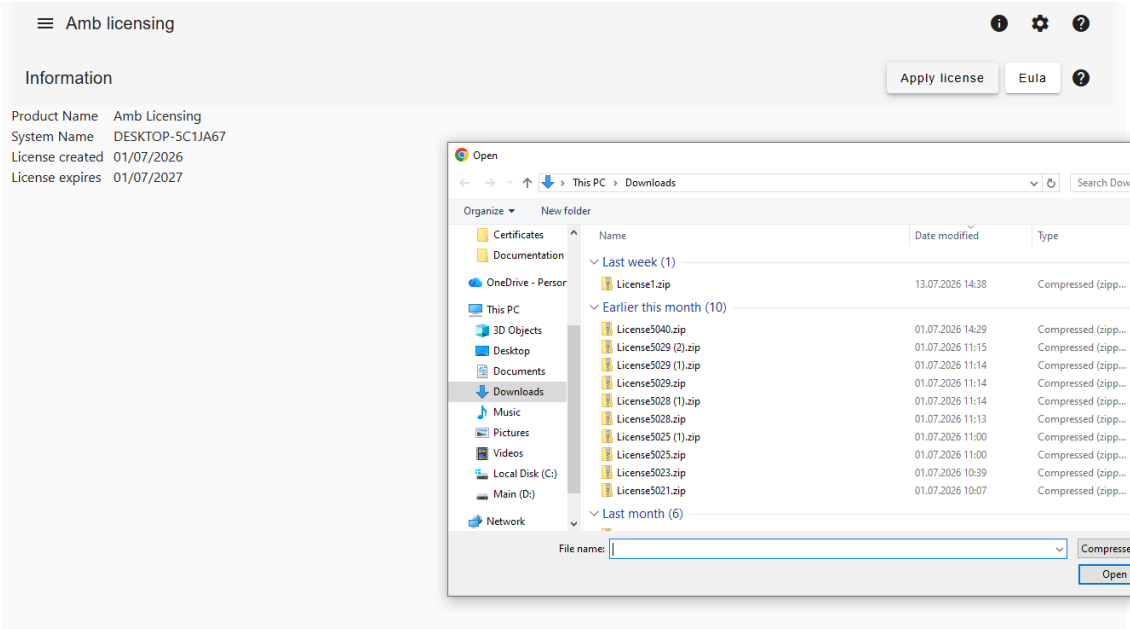
Product



This page is a hierarchical view of the product, components and features. You can add/remove components/features. You can also remove the whole project itself. If you navigated to this page from organization page you’ve got an option to generate a license. You can add the following features to the product/component:

- Version
- Integer
- Float
- DateTime
- String

Information

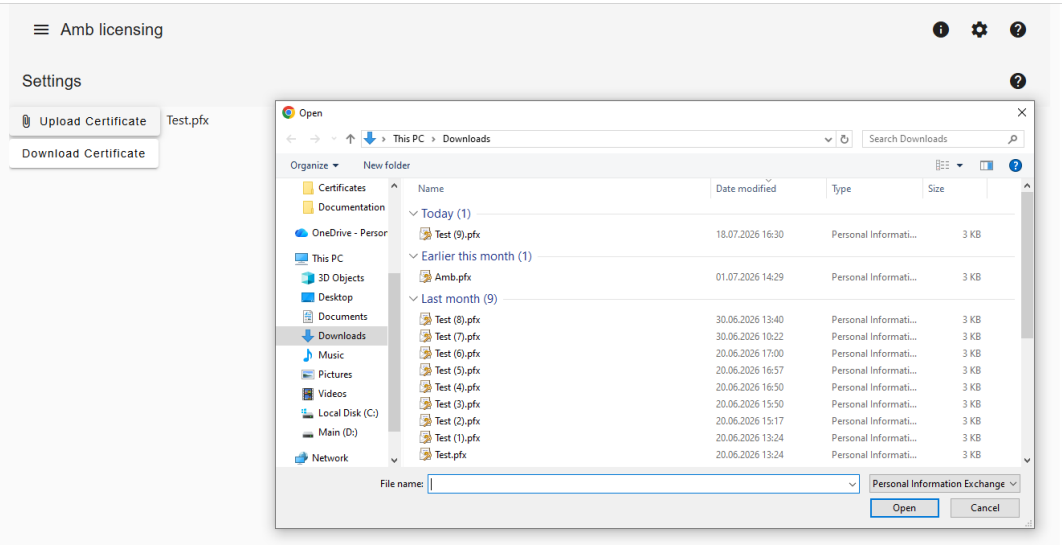


In this page we can see license information:

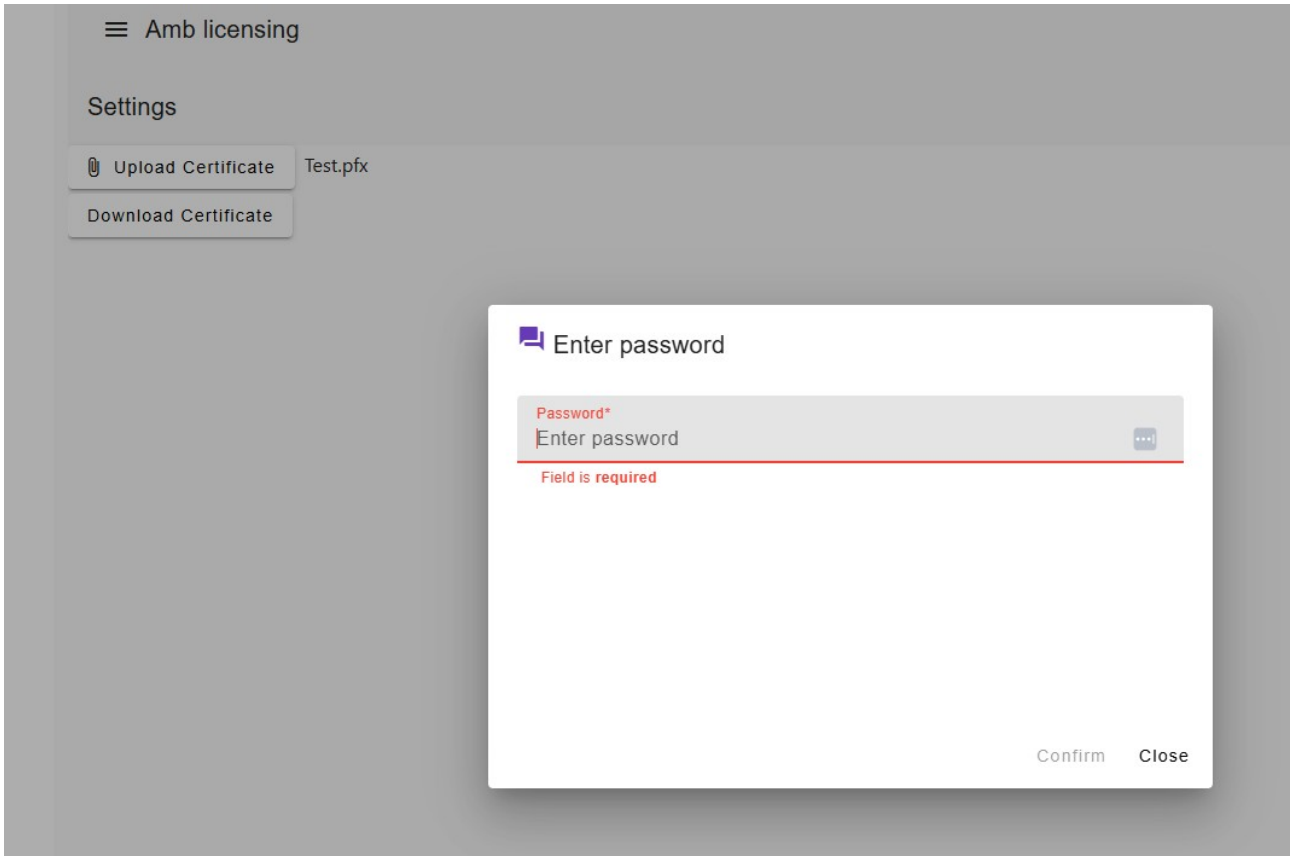
- Product Name;
- System Name;
- License created date;
- License expiration date;

We can also apply the license (by uploading the license file). Also there is an option to download Eula (if available).

Settings



In this page we can upload new pfx file into the system to sign the licenses. If the file is already uploaded, you can see it’s name next to ‘upload’ button. The file contains both public and private crypto keys. The uploaded file is encrypted with data protection services before serializing into database.



You have to enter valid password to pfx file before uploading.

How to

Generate a license

Go to Organizations page, select an organization you want to issue license for, press ‘Products’ button, find a product you want to license and then press the button ‘Generate license’. You will see the following page:

You can remove some components/features from the license here. You can also upload Eula file and set expiration date for the license.

When done, press ‘Generate license’ button in toolbar and the license will be generated.

Test Client Project

Test client project is for testing license files. You can use it for reading/parsing your license files and implementing/embedding license checking logic into your production application. You should replace Certificates/Test.cer public key file with that used to sign your license file (pfx file). Below is an example how to create self-signed pfx file with public/private keys and to extract public and private keys from it (in powershell script).

```
$cert = New-SelfSignedCertificate `
    -Type CodeSigningCert `
    -Subject "CN=LicenseSigning" `
    -KeyLength 2048 `
    -CertStoreLocation "Cert:\CurrentUser\My" `
    -KeyExportPolicy Exportable `
    -NotAfter (Get-Date).AddYears(1)

$secPwd = ConvertTo-SecureString "pfx" -AsPlainText -Force

Export-PfxCertificate `
    -Cert $cert `
    -FilePath "[path to pfx file]" `
    -Password $secPwd `
    -Force
```

```
Export-Certificate `
    -Cert $cert `
    -FilePath "[path to cer file]"`
    -Type CERT `
    -Force
```

The project reads test license file, deserialises it into the models and writes the output json to console.

```
{
  "created": "2026-08-13T13:38:23.0734277Z",
  "expires": "2027-08-13T13:37:39Z",
  "product": {
    "name": "TestClient",
    "description": "Test Client",
    "modules": [
      {
        "name": "Component",
        "description": "",
        "modules": [],
        "features": [
          {
            "featureType": 4,
            "stringValue": "321",
            "name": "String"
          }
        ]
      }
    ],
    "features": [
      {
        "featureType": 5,
        "dateTimeValue": "2026-08-13T13:32:43.275Z",
        "name": "DateTime"
      },
      {
        "featureType": 3,
        "doubleValue": 0.5,
        "name": "Float"
      },
      {
        "featureType": 2,
        "integerValue": 5,
        "name": "Integer"
      },
      {
        "featureType": 4,
        "stringValue": "ABC123",
        "name": "System Name"
      },
      {
        "featureType": 1,
        "version": "1.0.0.5",
        "name": "Version"
      }
    ]
  },
  "eula": "U2FtcGxlIEV1bGEgdGV4dC4="
}
Eula:
Sample Eula text.
Press any key to exit...
```