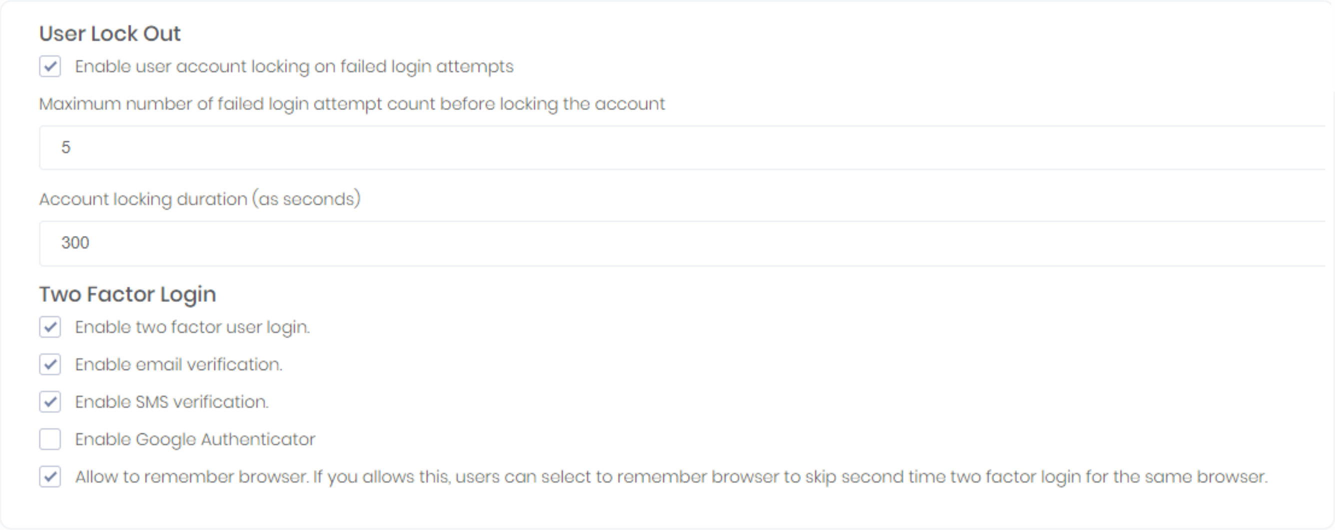
Two Factor Authentication

ASP.NET Zero is ready to provide two factor login, but it's disabled as default. You can easily enable it in host settings page (in Security tab):

Feedback



User Lock Out

☑ Enable user account locking on failed login attempts

Maximum number of failed login attempt count before locking the account

5

Account locking duration (as seconds)

300

Two Factor Login

☑ Enable two factor user login.

☑ Enable email verification.

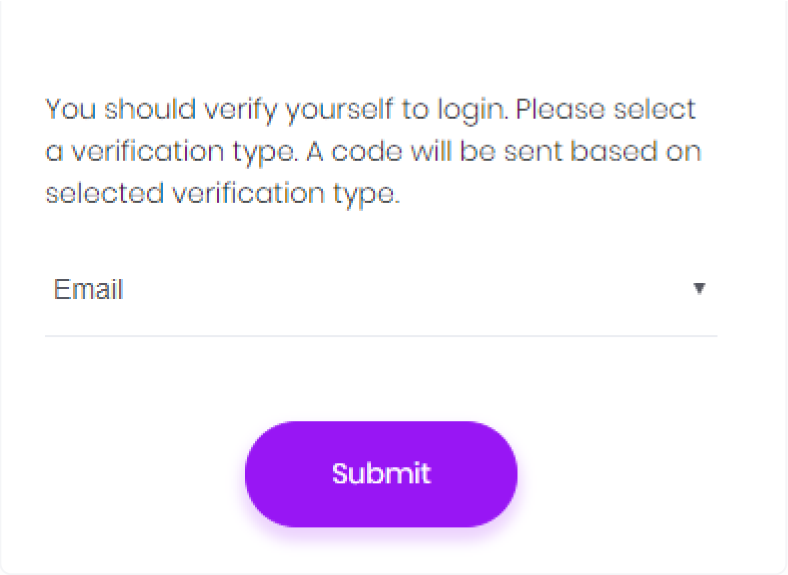
☑ Enable SMS verification.

□ Enable Google Authenticator

☑ Allow to remember browser. If you allows this, users can select to remember browser to skip second time two factor login for the same browser.

Note: In a multi-tenant application, two factor authentication is available to tenants only if it's enabled in the host settings. Also, email verification and SMS verification settings are only available in the host side. This is by design.

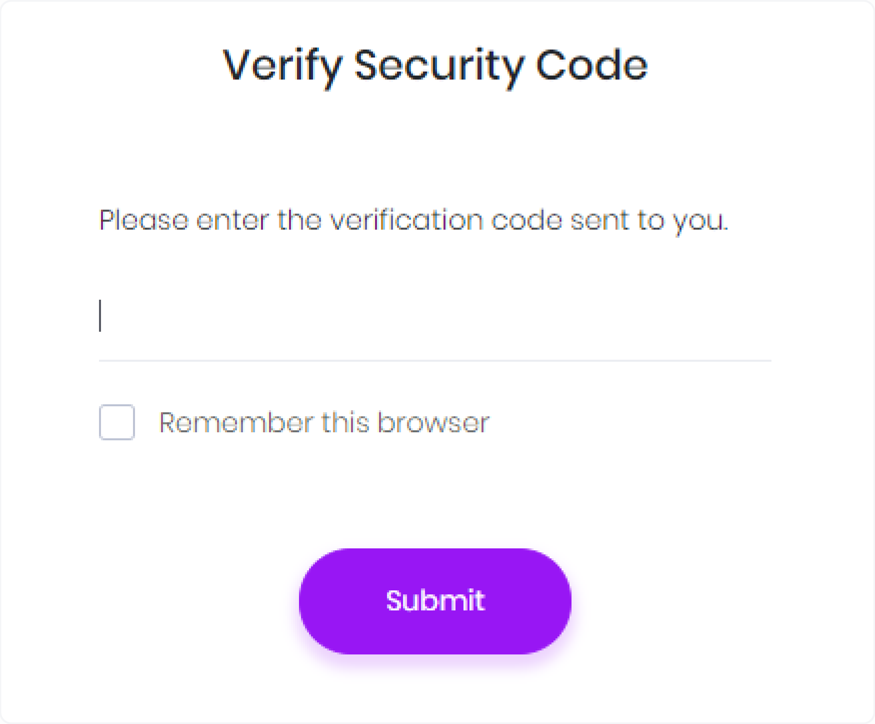
When it's enabled, user is asked to select a verification provider after entering user name and password:



You should verify yourself to login. Please select a verification type. A code will be sent based on selected verification type.

Email ▾

Then a **confirmation code** is sent to the selected provider and user enters the code in the next page:



Verify Security Code

Please enter the verification code sent to you.

□ Remember this browser

Email Verification

This is available if user has a confirmed email address. Since email sending is disabled in debug mode, you can see the code in logs. In release mode, email will be sent. You can change this behaviour in the PreInitialize method of **{YourProjectName}CoreModule.cs**.

```
if (DebugHelper.IsDebug)
{
    //Disabling email sending in debug mode
    Configuration.ReplaceService<IEmailSender, NullEmailSender>(DependencyLifeStyle
}
```

SMS Verification

This is available if user has a confirmed phone number. In order to validate a phone number, a user should open my settings modal as explained in here. When the user enters the phone number, a button will appear to validate to phone number. When the validate button is clicked, an SMS message is sent to entered phone number including a validation code and another modal window is opened for entering the validation code. When user enters this validation code, the phone number for the user will be set as validated. SMS sending is implemented in ASP.NET Zero using Twilio but an empty SmsSender class is used by default which writes SMS messages to log file.

If you want to use Twilio for sending SMS on your app, please refer to next section. You can also implement `ISmsSender` interface and use your custom implementation for sending SMS. In that case, you need to configure ASP.NET Zero to use your custom implementation like below in the PreInitialize method of **{YourProjectName}CoreModule.cs**:

```
Configuration.ReplaceService<ISmsSender,CustomSmsSender>();
```

Twilio Integration

In order to enable Twilio integration, just uncomment the following line in your **{YourProjectName}CoreModule** (in your Core project):

```
Configuration.ReplaceService<ISmsSender,TwilioSmsSender>();
```

You also need to configure **AccountSid**, **AuthToken** and **SenderNumber** in `appsetting.json` file.

Next

ASP.NET CORE ANGULAR

DOCUMENTS