



Feedback



Getting Started

This document is aimed to create and run an ASP.NET Zero based project in just 5 minutes. It's assumed that you already [purchased](#) and created your ASP.NET Zero account.

Create Your Project

[Login to aspnetzero.com](#) with your username and password. Go to the [download](#) page. You will see a form as shown below:



ASP.NET CORE ANGULAR

DOCUMENTS

Acme.PhoneBookDemo

Project name is mandatory! As it will be the part of the namespace, the first character must be a letter following letters, digits, dots or underscores. C# keywords are not allowed to be used as a project name. Be aware that project name cannot be renamed later! Sample project names: **MyPortal** or **Acme.MyPortal**.

ASP.NET CORE & Angular

Select the project type.

v6.4.0 (latest)

Select the version. If you are starting a new project, it is recommended to select the latest version.

☒ **Single solution**

Merges **AngularUI** and **Host** projects into a single solution.

.NET Core 2.2

Select the .NET Framework version. Use **.NET Core**, if you want that your project runs across platforms (Windows, Linux and MacOS).

Generate Demo Project

Generate →

Feedback

Select **ASP.NET Core & Angular** as Project Type and fill other required fields. Your project will be ready in one minute. When you open the downloaded zip file, you will see two folders:

Name

- angular
- aspnet-core
- .gitignore
- README.md

- angular** folder contains the Angular UI application which is configured to work with the [angular-cli](#).



🔗 Merging Client and Server Solutions

Client and Server solutions are designed to work separately by default. If you want to work on a single Visual Studio solution, you can select "Single Solution" checkbox while downloading your project.

If you have selected this option, please follow [Getting Started](#) document for ASP.NET Core & Angular merged solution.

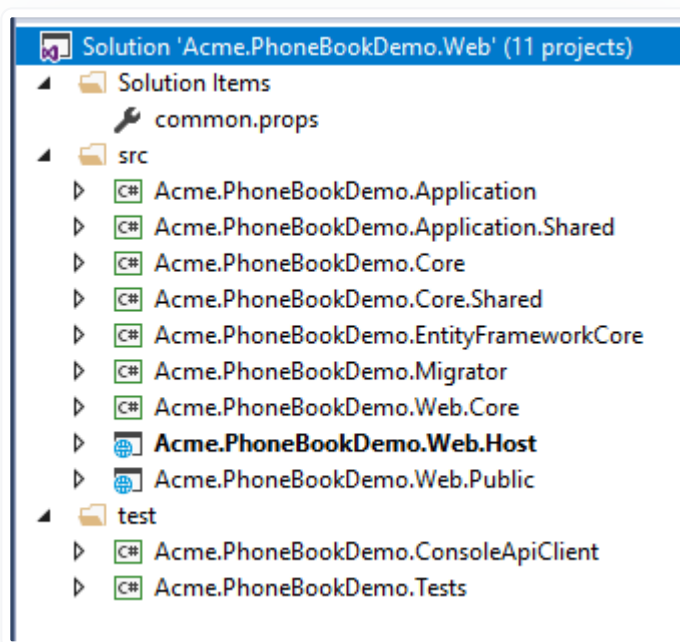


Pre Requirements

- [Visual Studio 2017 \(v15.9.0+\)](#) (for backend ASP.NET Core application)
- [Node.js 6.9+ with NPM 3.10+](#)
- [Yarn](#)

ASP.NET Core Application

When you open the server side solution (*.Web.sln) using **Visual Studio 2017+**, you will see the solution structure as below:



If you want to work on only Xamarin project, open ***.Mobile.sln** solution. If you want to work on both of Xamarin and Web projects, open ***.All.sln** solution.



Database

Connection String

Open **appsettings.json** in **.Web.Host** project and change the **Default** connection string if you want:

```
"ConnectionStrings": {  
  "Default": "Server=localhost; Database=PhoneBookDemoDb; Trusted_Connection=True  
}
```

Feedback

Migrations

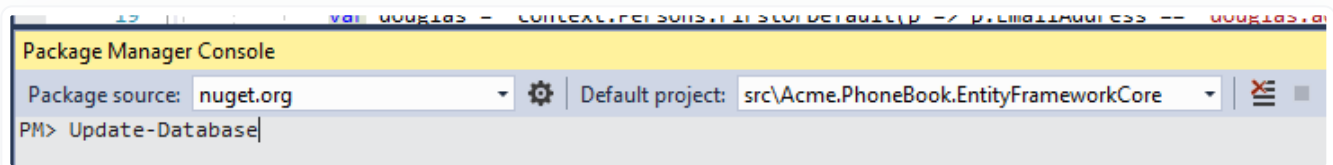
We have two options to create and migrate database to the latest version.

ASP.NET Zero Migrator Application

ASP.NET Zero solution includes a **.Migrator** (like `Acme.PhoneBookDemo.Migrator`) project in the server side solution. You can run this tool for database migrations on development and production (see [migrator docs](#) for more information).

Entity Framework Migration Command

You can also use Entity Framework Core's built-in tools for migrations. Open **Package Manager Console** in Visual Studio, set ***.EntityFrameworkCore** as the **Default Project** and run the **Update-Database** command as shown below:

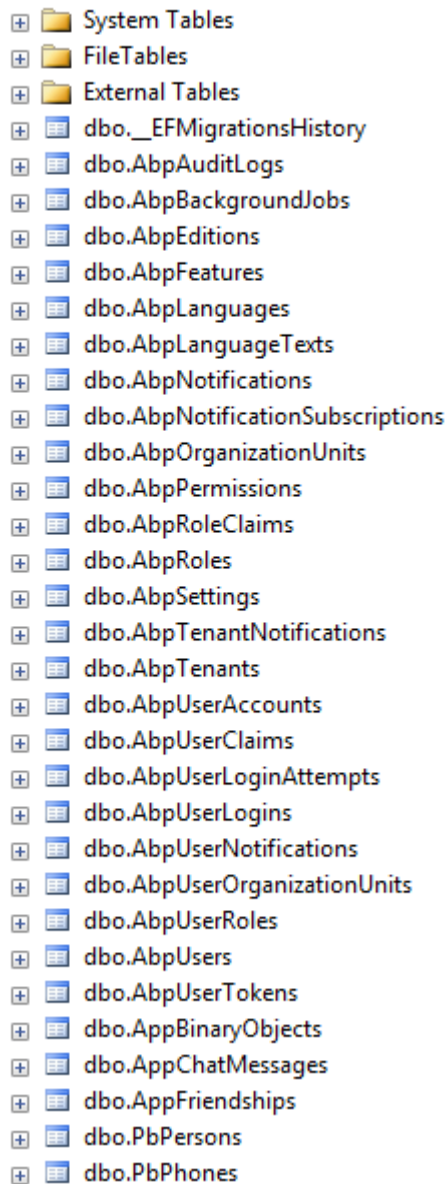


This command will create your database. Initial data will be inserted when you run the ***.Web.Host** project. You can open SQL Server Management Studio to check if database is created:



ASP.NET CORE ANGULAR

DOCUMENTS



- + System Tables
- + FileTables
- + External Tables
- + dbo.__EFMigrationsHistory
- + dbo.AbpAuditLogs
- + dbo.AbpBackgroundJobs
- + dbo.AbpEditions
- + dbo.AbpFeatures
- + dbo.AbpLanguages
- + dbo.AbpLanguageTexts
- + dbo.AbpNotifications
- + dbo.AbpNotificationSubscriptions
- + dbo.AbpOrganizationUnits
- + dbo.AbpPermissions
- + dbo.AbpRoleClaims
- + dbo.AbpRoles
- + dbo.AbpSettings
- + dbo.AbpTenantNotifications
- + dbo.AbpTenants
- + dbo.AbpUserAccounts
- + dbo.AbpUserClaims
- + dbo.AbpUserLoginAttempts
- + dbo.AbpUserLogins
- + dbo.AbpUserNotifications
- + dbo.AbpUserOrganizationUnits
- + dbo.AbpUserRoles
- + dbo.AbpUsers
- + dbo.AbpUserTokens
- + dbo.AppBinaryObjects
- + dbo.AppChatMessages
- + dbo.AppFriendships
- + dbo.PbPersons
- + dbo.PbPhones

Feedback



You can use EF console commands for development and Migrator.exe for production. But notice that; Migrator.exe supports running migrations in multiple databases at once, which can be useful in development/production for multi tenant applications.

Configure Multi-Tenancy

ASP.NET Zero supports multi-tenant and single-tenant applications. Multi-tenancy is **enabled by default**. If you don't have an idea about multi-tenancy, you can read it on [wikipedia.org/wiki/Multitenancy](https://en.wikipedia.org/wiki/Multitenancy). If you don't want to create a multi-tenant application, you can **disable** it by setting

PhoneBookDemoConsts.MultiTenancyEnabled to false in the ***.Core.Shared** project (name of the PhoneBookDemoConsts will be like *YourProjectNameConsts* for your project).



ASP.NET CORE ANGULAR

DOCUMENTS

Once you've done the configuration, you can run the application. Server side application only contains APIs. When you start the application you will see a login page like below:

Log In

Tenancy name

User name or email

Password

☐ Remember me

Log in

Feedback



If you log in to host application, then you will see a page like below:

You are already logged in with user : .admin

Swagger UI **Hangfire** **GraphQL Playground**


[Logout](#)

You can navigate to **Swagger UI**, **Hangfire Dashboard** or **GraphQL Playground** from this page. Note that, by default only Swagger UI is enabled, you can enable Hangfire by following [Hangfire documentation](#) and



ASP.NET CORE ANGULAR

DOCUMENTS

 **swagger**

http://localhost:22742/swagger/v1/swagger.json

PhoneBookDemo API V1

PhoneBookDemo API

Account

Show/Hide | List Operations | Expand Operations

POST	/api/services/app/Account/IsTenantAvailable
POST	/api/services/app/Account/Register
POST	/api/services/app/Account/SendPasswordResetCode
POST	/api/services/app/Account/ResetPassword
POST	/api/services/app/Account/SendEmailActivationLink
POST	/api/services/app/Account/ActivateEmail
POST	/api/services/app/Account/Impersonate
POST	/api/services/app/Account/BackToImpersonator
POST	/api/services/app/Account/SwitchToLinkedAccount

AuditLog

Show/Hide | List Operations | Expand Operations

GET	/api/services/app/AuditLog/GetAuditLogs
GET	/api/services/app/AuditLog/GetAuditLogsToExcel

Caching

Show/Hide | List Operations | Expand Operations

GET	/api/services/app/Caching/GetAllCaches
POST	/api/services/app/Caching/ClearCache
POST	/api/services/app/Caching/ClearAllCaches

Chat

Show/Hide | List Operations | Expand Operations

GET	/api/services/app/Chat/GetUserChatFriendsWithSettings
-----	---

Feedback



Angular Application

Restore Packages

Navigate to the **angular** folder, open a command line and run the following command to restore the packages:



Then, run the following command to create dynamic bundles (*This is only required for the first time when you download the project or when you update dynamic bundles*):

```
npm run create-dynamic-bundles
```

We use [yarn](#) because NPM has some problems; It is slow and can not consistently resolve dependencies. Yarn solves those problems and it is compatible to NPM as well.

Feedback



Running The Application

Run the following command in the command line:

```
npm start
```

Once the application compiled, you can browse <http://localhost:4200> in your browser. ASP.NET Zero also has also **HMR** (Hot Module Replacement) enabled. You can use the following command (instead of NPM start) to enable HMR on development time:

```
npm run hmr
```

Login

All ready! Now, you can login to the application:



ASP.NET CORE ANGULAR

DOCUMENTS

Log in

User name or email *

Password *

☐

Remember me

[Forgot password?](#)

Log in

[New tenant](#) | [Email activation](#)



Feedback



If multi-tenancy is enabled, you will see the current tenant and a change link. If so, click to **Change** and enter **default** as tenant name. If you leave it empty, you login as the host admin user. Then enter **admin** as username and **123qwe** as password (remember to change it to a more secure password on production!).

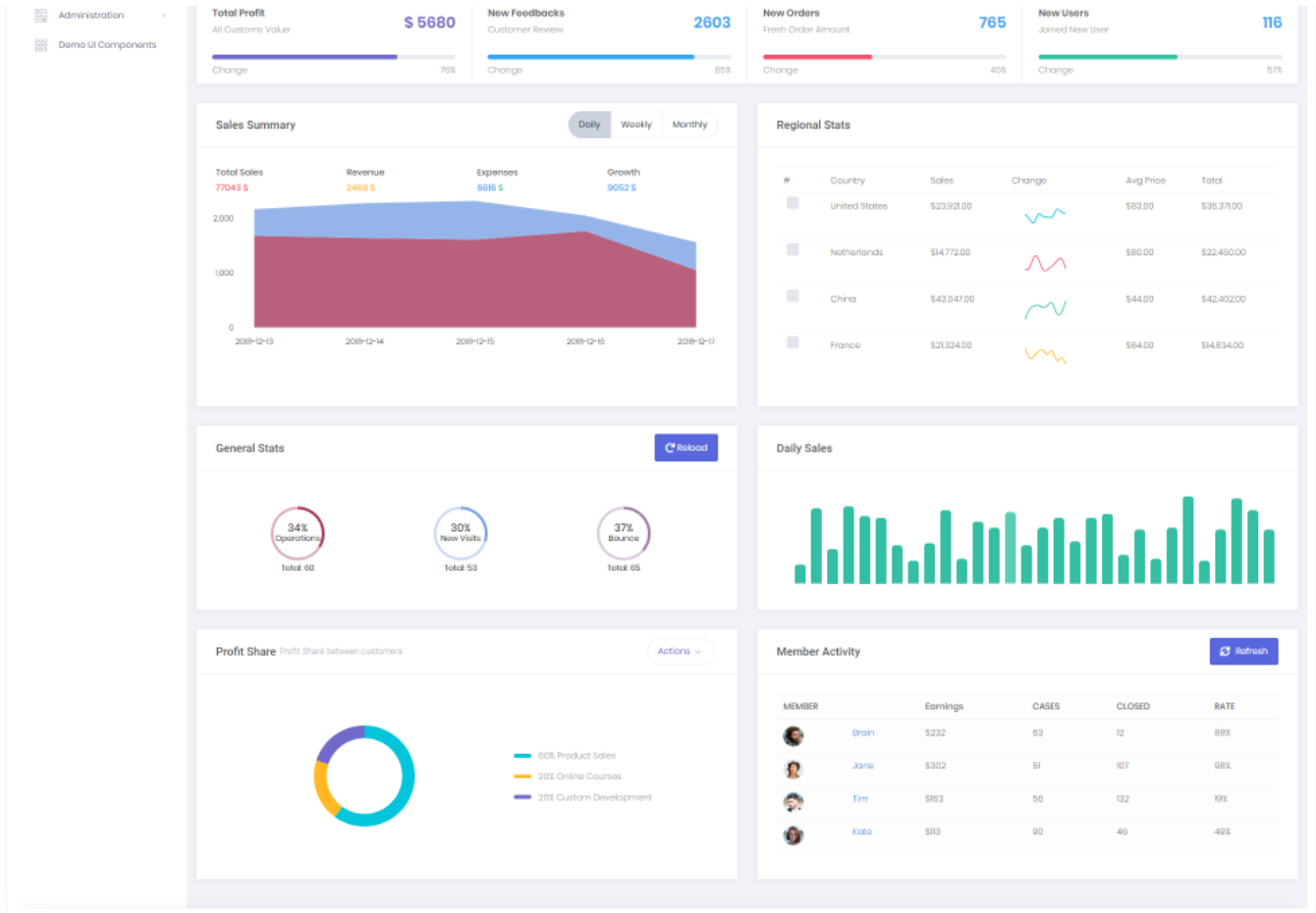
Application UI

When you login to the application as a Tenant admin, you will see the sample dashboard screen:



ASP.NET CORE ANGULAR

DOCUMENTS



ASP.NET Zero Power Tools

ASP.NET Zero Power Tools lets you to create a new page from the database to the UI layer by just typing your entity properties. It creates the entity, related permissions, application services, DTOs, client-side code, a menu element and so on... It also creates & applies database migrations.

Finally you will have a CRUD page with insert, update, delete, list and excel export functions. To minimize the effort of creating a new page, install the Power Tools from the following link:

<https://marketplace.visualstudio.com/items?itemName=Volosoft.AspNetZeroPowerTools>

Next

- [Solution Overview](#)

