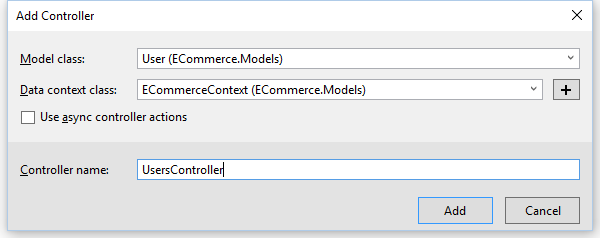
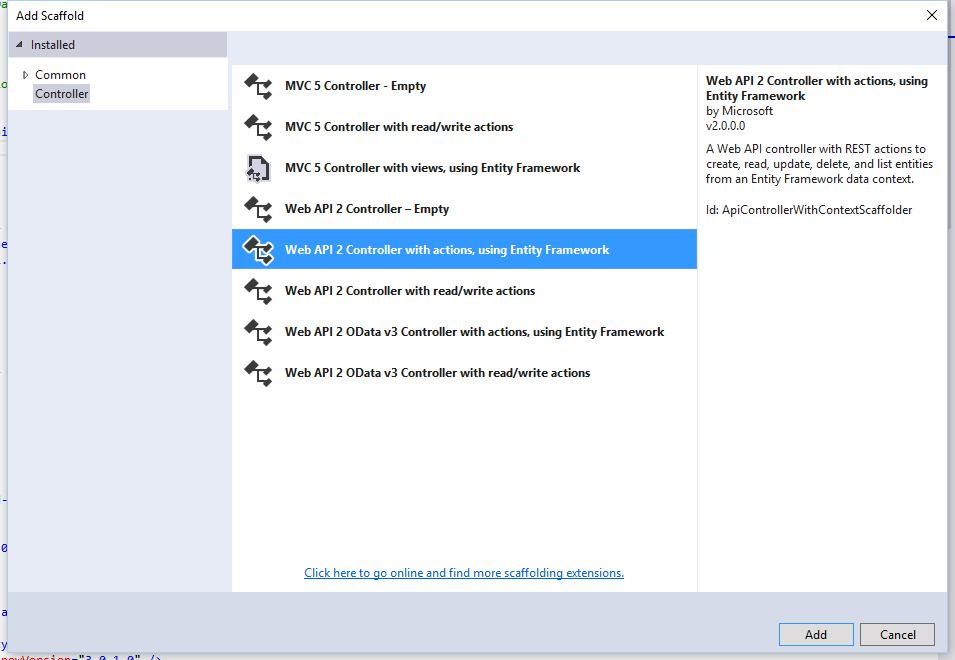
WEB API AND PUBLISH

# WEB API

1. Divide controllers into two folders: MVC and API, and move all the existing controllers to MVC folder:



1. Create the new controller base on WEB API 2, on model Users:



1. Create the class UserResponse:

public class UserResponse

{

public int UserId { get; set; }

public string UserName { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Photo { get; set; }

public string Phone { get; set; }

public string Address { get; set; }

public int DepartmentId { get; set; }

public string DepartmentName { get; set; }

public int CityId { get; set; }

public string CityName { get; set; }

public Company Company { get; set; }

public bool IsAdmin { get; set; }

public bool IsUser { get; set; }

public bool IsCustomer { get; set; }

public bool IsSupplier { get; set; }

}

1. Add this annotation to UsersController in API folder

[RoutePrefix("api/Users")]

1. Add the method login into UsersController in API folder:

[HttpPost]

[Route("Login")]

public IHttpActionResult Longin(JObject form)

{

db.Configuration.ProxyCreationEnabled = false;

string email = string.Empty;

string password = string.Empty;

dynamic jsonObject = form;

try

{

email = jsonObject.Email.Value;

password = jsonObject.Password.Value;

}

catch

{

return this.BadRequest("Incorrect call");

}

var userContext = new ApplicationDbContext();

var userManager = new UserManager<ApplicationUser>(new UserStore<ApplicationUser>(userContext));

var userASP = userManager.Find(email, password);

if (userASP == null)

{

return this.BadRequest("User or password wrong");

}

var user = db.Users.Where(u => u.UserName == email).Include(u => u.Company).Include(u => u.City).Include(u => u.Department).FirstOrDefault();

if (user == null)

{

return this.BadRequest("User not found");

}

var userResponse = new UserResponse

{

Address = user.Address,

CityId = user.CityId,

CityName = user.City.Name,

Company = user.Company,

DepartmentId = user.DepartmentId,

DepartmentName = user.Department.Name,

FirstName = user.FirstName,

IsAdmin = userManager.IsInRole(userASP.Id, "Admin"),

IsCustomer = userManager.IsInRole(userASP.Id, "Admin"),

IsSupplier = userManager.IsInRole(userASP.Id, "Admin"),

IsUser = userManager.IsInRole(userASP.Id, "User"),

LastName = user.LastName,

Phone = user.Phone,

Photo = user.Photo,

UserId = user.UserId,

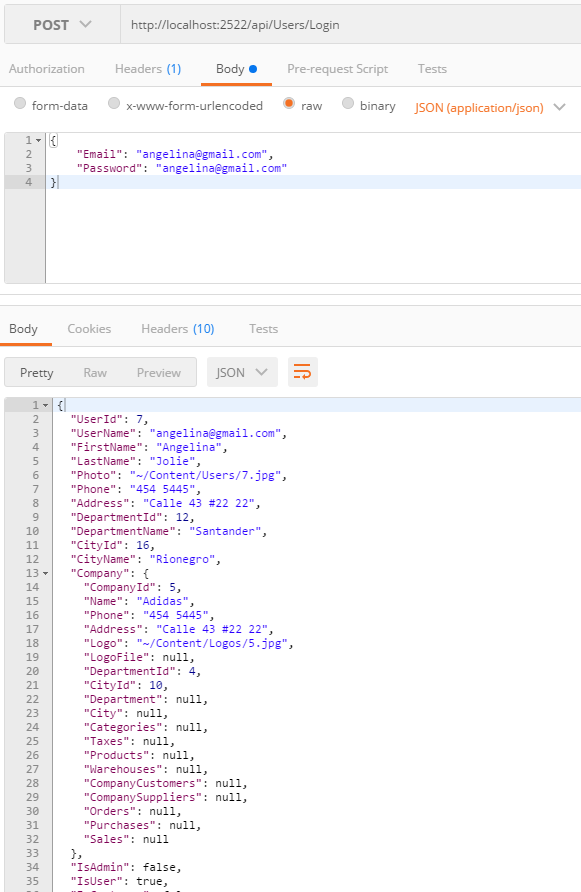
UserName = user.UserName,

};

return this.Ok(userResponse);

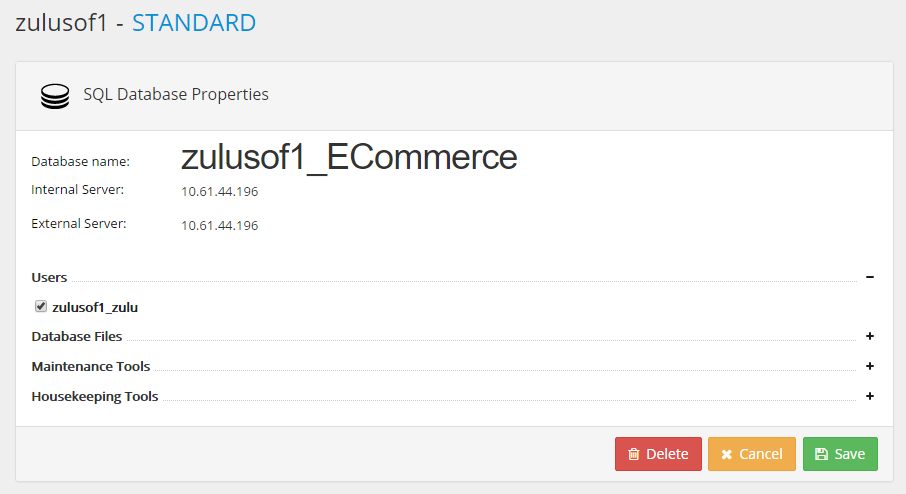
}

1. Test it in postman:



# Publish the APP

1. Make a data base backup, and create the DB in the hosting:



1. Modify the WEB config by:

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<!--<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />-->

</configSections>

<connectionStrings>

<remove name="DefaultConnection"/>

<!--<add name="DefaultConnection"

connectionString="Data Source=.;Initial Catalog=ECommerce5;Integrated Security=True"

providerName="System.Data.SqlClient" />-->

<add name="DefaultConnection"

connectionString="Data Source=10.61.44.196;Initial Catalog=zulusof1\_ECommerce;Persist Security Info=True;User ID=zulusof1\_zulu;Password=Roger1974."

providerName="System.Data.SqlClient" />

</connectionStrings>

And add those lines in run time section:

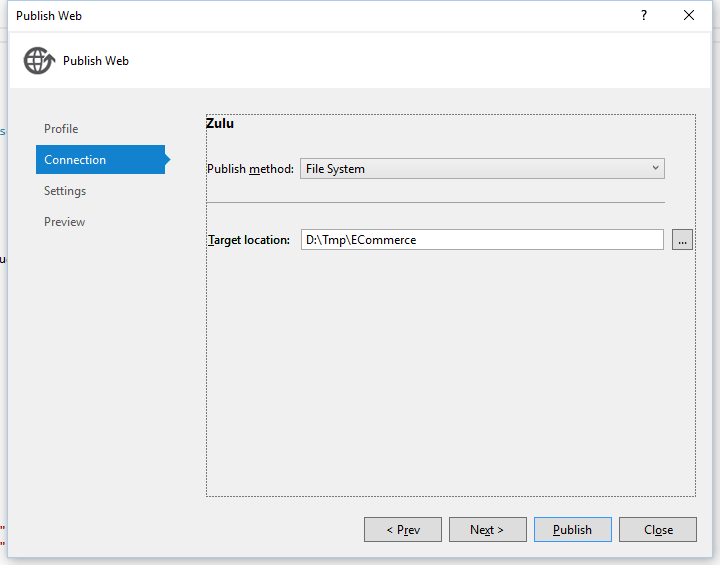
<dependentAssembly>

<assemblyIdentity name="EntityFramework" publicKeyToken="b77a5c561934e089" culture="neutral" />

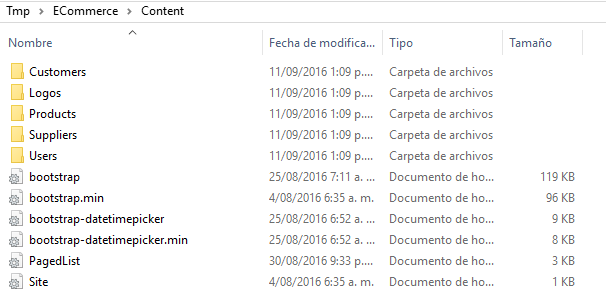
<bindingRedirect oldVersion="0.0.0.0-5.0.0.0" newVersion="6.0.0.0" />

</dependentAssembly>

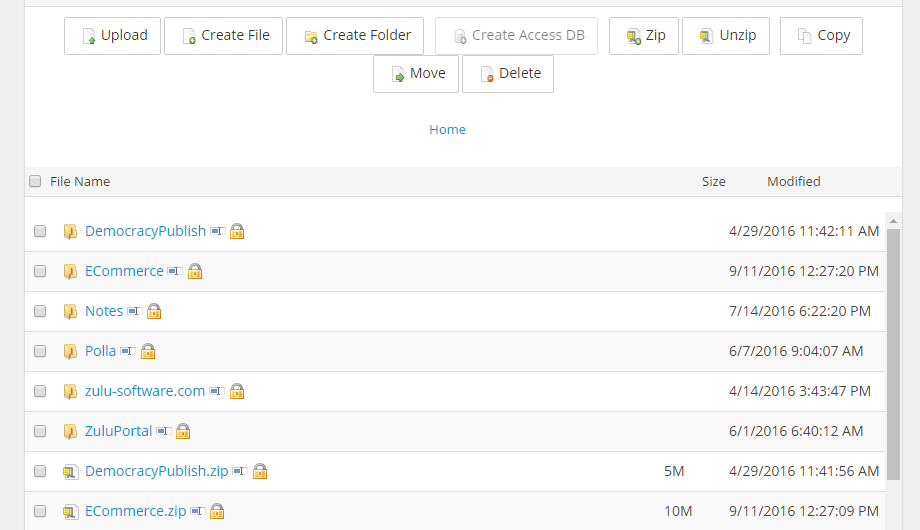
1. Generate the build package:



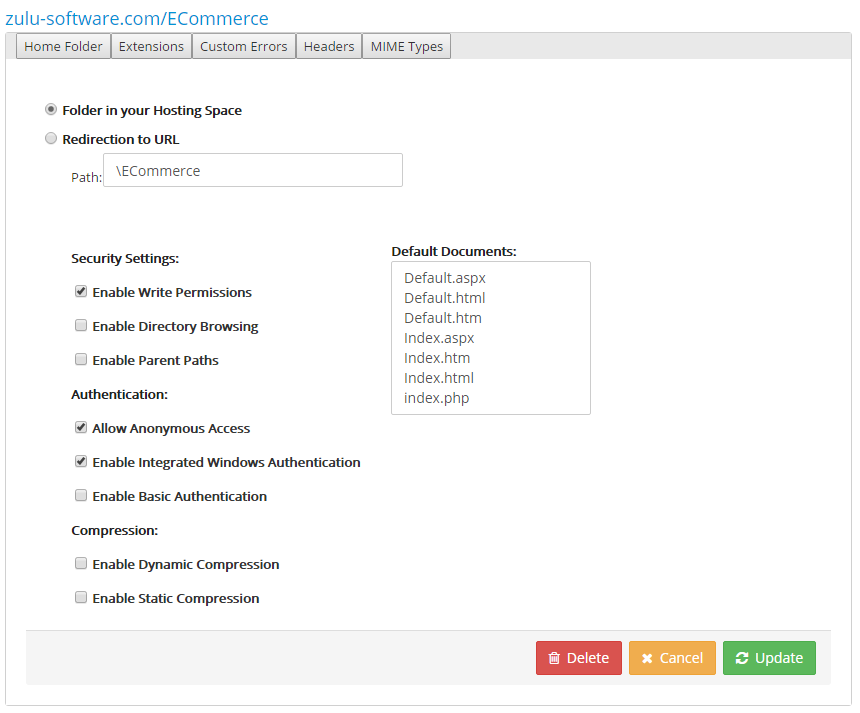
1. Don’t forget add into content folder the images:



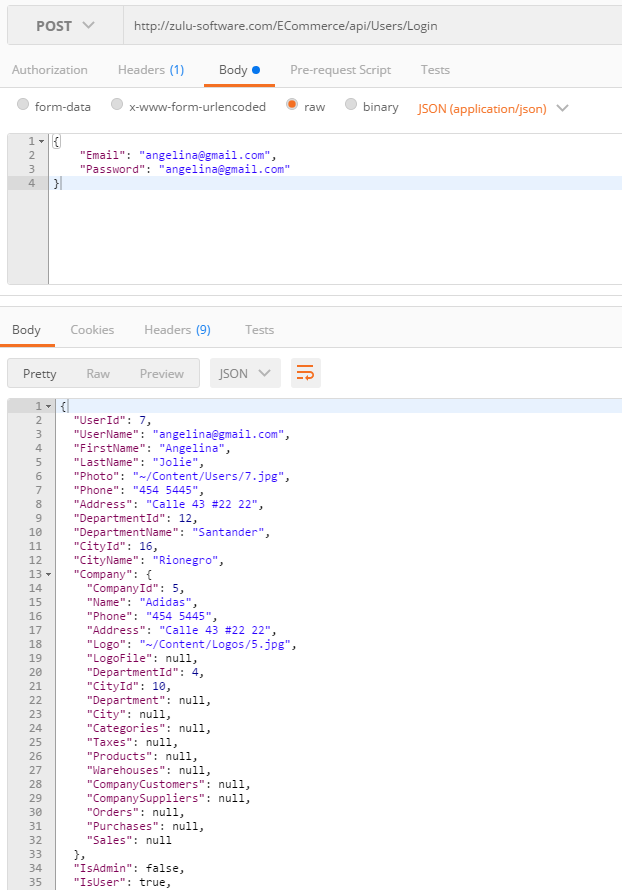
1. Compress and upload the file in the hosting:



1. Create the new virtual folder on hosting:



1. Test it in postman:



# Consume the method

1. Create the cross platform project, and add the LoginPage with this code:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="ECommerceApp.LoginPage"

Title="Login">

<ScrollView>

<StackLayout Spacing="10">

<Label Text="ECommerce"

FontSize="30"

FontAttributes="Bold"

HorizontalOptions="Center"

VerticalOptions="Center"/>

<Entry x:Name="userEntry"

Keyboard="Email"

Placeholder="Enter your email"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

<Entry x:Name="passwordEntry"

IsPassword="True"

Placeholder="Enter your password"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

<StackLayout Orientation="Horizontal">

<Label Text="Remember me"

HorizontalOptions="Start"

VerticalOptions="Center"/>

<Switch x:Name="remembermeSwitch"

HorizontalOptions="EndAndExpand"

VerticalOptions="Center"/>

</StackLayout>

<ActivityIndicator x:Name="waitActivityIndicator"

HorizontalOptions="Center"

VerticalOptions="Center"/>

<Button x:Name="loginButton"

Text="Login"

BackgroundColor="Yellow"

TextColor="Black"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

<Button x:Name="forgotPasswordButton"

Text="Forgot password"

BackgroundColor="Purple"

TextColor="White"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

<Button x:Name="registerButton"

Text="New User"

BackgroundColor="Aqua"

TextColor="Black"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

</StackLayout>

</ScrollView>

</ContentPage>

And this code, in the code behind:

public partial class LoginPage : ContentPage

{

public LoginPage()

{

InitializeComponent();

Padding = Device.OnPlatform(

new Thickness(10, 20, 10, 10),

new Thickness(10),

new Thickness(10));

}

}

Don’t forget to modify the App.cs by:

public App()

{

MainPage = new LoginPage();

}

1. Test It in the whole platforms:

|  |  |  |
| --- | --- | --- |
| Android | iOS | Windows Phone |
|  |  |  |

1. All this nugets: Microsoft.BCL, Microsoft.Net.Http, Newtonsoft.Json to all projects in the solution.
2. Add the class to deserialize the response from server:

public class Company

{

public int CompanyId { get; set; }

public string Name { get; set; }

public string Phone { get; set; }

public string Address { get; set; }

public string Logo { get; set; }

public object LogoFile { get; set; }

public int DepartmentId { get; set; }

public int CityId { get; set; }

public object Department { get; set; }

public object City { get; set; }

public object Categories { get; set; }

public object Taxes { get; set; }

public object Products { get; set; }

public object Warehouses { get; set; }

public object CompanyCustomers { get; set; }

public object CompanySuppliers { get; set; }

public object Orders { get; set; }

public object Purchases { get; set; }

public object Sales { get; set; }

}

public class UserResponse

{

public int UserId { get; set; }

public string UserName { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Photo { get; set; }

public string Phone { get; set; }

public string Address { get; set; }

public int DepartmentId { get; set; }

public string DepartmentName { get; set; }

public int CityId { get; set; }

public string CityName { get; set; }

public Company Company { get; set; }

public bool IsAdmin { get; set; }

public bool IsUser { get; set; }

public bool IsCustomer { get; set; }

public bool IsSupplier { get; set; }

public string Password { get; set; }

public string FullName { get { return string.Format("{0} {1}", FirstName, LastName); }

public string PhotoFullPath

{

get

{

if (!string.IsNullOrEmpty(Photo))

{

return string.Format("http://www.zulu-software.com/ECommerce{0}", Photo.Substring(1));

}

return string.Empty;

}

}

}

1. Add the class to make the request to service:

public class LoginRequest

{

public string Email { get; set; }

public string Password { get; set; }

}

1. Add the method to Login button:

loginButton.Clicked += LoginButton\_Clicked;

private async void LoginButton\_Clicked(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(userEntry.Text))

{

await DisplayAlert("Error", "You must enter an email", "Acept");

return;

}

if (string.IsNullOrEmpty(passwordEntry.Text))

{

await DisplayAlert("Error", "You must enter a password", "Acept");

return;

}

Login(userEntry.Text, passwordEntry.Text);

}

private async void Login(string email, string password)

{

waitActivityIndicator.IsRunning = true;

var loginRequest = new LoginRequest

{

Email = userEntry.Text,

Password = passwordEntry.Text,

};

var result = string.Empty;

try

{

var jsonRequest = JsonConvert.SerializeObject(loginRequest);

var httpContent = new StringContent(jsonRequest, Encoding.UTF8, "application/json");

var client = new HttpClient();

client.BaseAddress = new Uri("http://zulu-software.com");

var url = "/ECommerce/api/Users/Login";

var response = await client.PostAsync(url, httpContent);

if (!response.IsSuccessStatusCode)

{

waitActivityIndicator.IsRunning = false;

await DisplayAlert("Error", "Wrong user or password ", "Acept");

passwordEntry.Text = string.Empty;

passwordEntry.Focus();

return;

}

result = await response.Content.ReadAsStringAsync();

}

catch (Exception ex)

{

waitActivityIndicator.IsRunning = false;

await DisplayAlert("Error", ex.Message, "Acept");

return;

}

var userResponse = JsonConvert.DeserializeObject<UserResponse>(result);

userResponse.Password = password;

await DisplayAlert("Message", "Fuck Yeah!!!", "Acept");

waitActivityIndicator.IsRunning = false;

}

1. Add the Main Page:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="ECommerceApp.Pages.MainPage"

Title="Ecommerce">

<ScrollView>

<StackLayout Spacing="10">

<Image x:Name="photoImage"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Label Text="Welcome"

FontSize="20"

FontAttributes="Bold"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Label x:Name="userNameLabel"

FontSize="30"

FontAttributes="Bold"

HorizontalOptions="Center"

VerticalOptions="Center" />

</StackLayout>

</ScrollView>

</ContentPage>

With the following code behind:

public partial class MainPage : ContentPage

{

private UserResponse userResponse;

public MainPage(UserResponse userResponse)

{

InitializeComponent();

Padding = Device.OnPlatform(

new Thickness(10, 20, 10, 10),

new Thickness(10),

new Thickness(10));

this.userResponse = userResponse;

}

protected override void OnAppearing()

{

base.OnAppearing();

userNameLabel.Text = userResponse.FullName;

photoImage.Source = userResponse.PhotoFullPath;

photoImage.HeightRequest = 280;

photoImage.WidthRequest = 280;

}

}

1. Modify the login pager by:

await Navigation.PushAsync(new MainPage(userResponse));

And App by:

MainPage = new NavigationPage(new LoginPage());