

Microsoft Tech Summit 2017

微软技术暨生态大会



构建前后端跨平台的移动APP架构

HOL - 206

周岳

北京视高盛景软件首席架构师

李加增

北京视高盛景软件架构师



周岳 - 个人介绍



微软最有价值专家 (MVP)



Xamarin认证开发者



视高盛景软件首席架构师

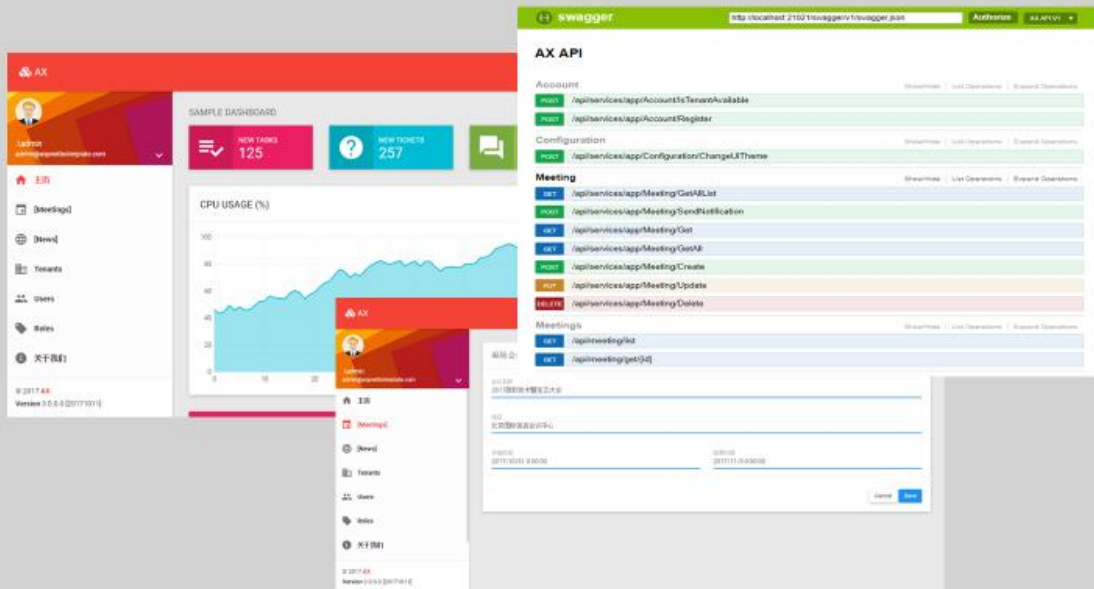


北京Xamarin UG组织者

跨平台企业开发微信交流群




使用.NET Core构建跨平台后台服务



使用Xamarin构建跨平台移动App





构建前后端跨平台的移动 APP 架构 之基于 Core 的跨平台后台服务

目录

Backend	6
1、创建 ABP 项目模板	6
2、恢复 nuget , 编译解决方案	3
3、初始化数据库	3
3.1、在 AX.Core 项目新建文件夹 “Meetings”	3
3.2、在 “Meeting” 目录下新建 Meeting.cs	3
3.3、在 AX.EntityFrameworkCore\EntityFrameworkCore\AXDbContext.cs 添加如下代码	4
3.4、将 AX.EntityFrameworkCore 项目设置为启动项目 , 在 VS 的 Package Manger Console 窗口中使用如下命令动态生成数据库表 (SQL Server 默认 Windows 认证)	5
4、创建 Meetings in Application	5
4.1、AX.Application 项目中新建文件夹 “ Meetings”	5
4.2、在 “Meetings” 目录下新建文件夹 “ Dto”	5
4.3、在 “Dto ”目录下新建 MeetingDto.cs、MeetingListDto.cs	5
4.4、在 “Meetings” 目录下新建 IMeetingAppService.cs、MeetingAppService.cs	6
5、创建 Meetings in Controllers	8
5.1、AX.Web.Mvc\Controllers 下新建 “ MeetingsController.cs”	8
6、创建 Meetings in View	10
6.1、AX.Web.Mv\Views 下新建文件夹 “ Meetings”	10
6.2、在 “Meetings” 目录下新建 Add.cshtml、Edit.cshtml、Index.cshtml	10
6.3、添加 JS	15
7、修改 PageNames	16
8、添加导航	16
9、运行程序	17
API	19

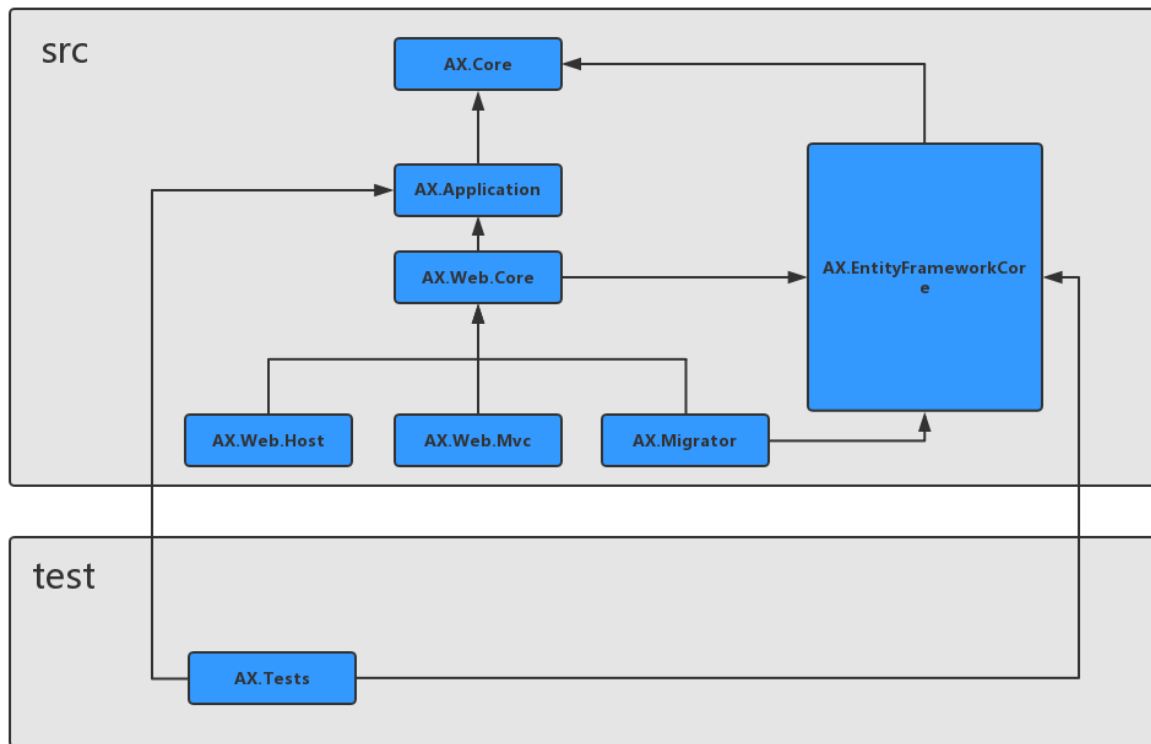
1、创建 Meetings in Controllers	19
1.1、AX.Web.Host\Controllers 下新建 “MeetingsController.cs”	19
2、运行程序.....	20

Backend

1、创建 ABP 项目模板

初始代码：https://projectax.blob.core.chinacloudapi.cn/src/AXCore_01Start.zip

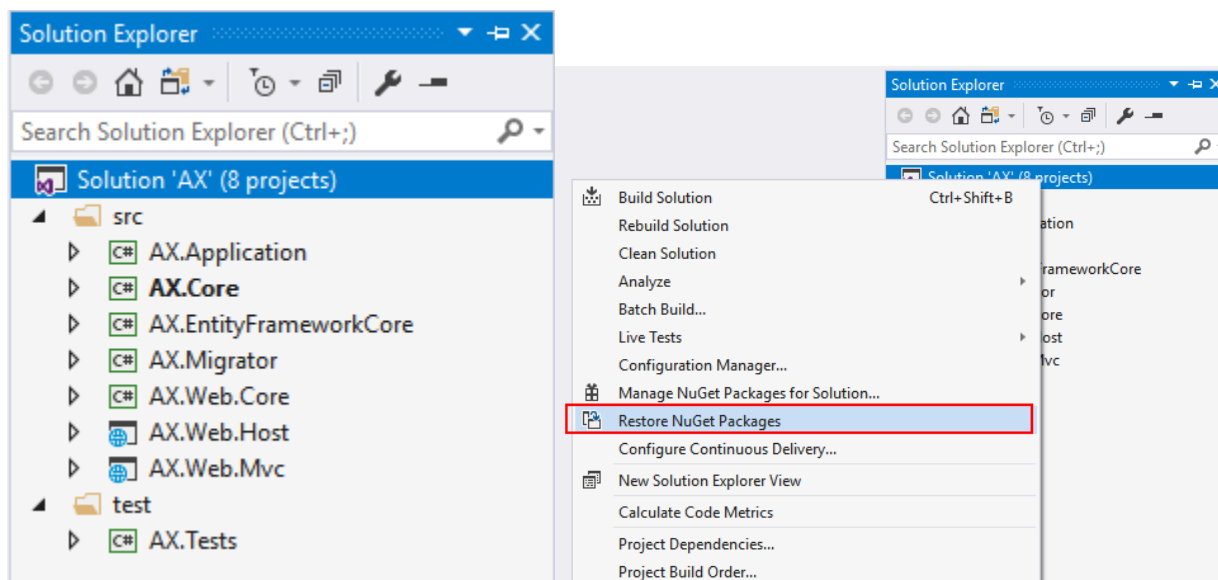
完整代码：https://projectax.blob.core.chinacloudapi.cn/src/AXCore_02Full.zip



解决方案结构介绍：

1. AX.Core 项目核心，从上图就可以看出，在 DDD 分层中，它应该作为领域层
2. AX.Application 应用服务层，包含各个应用逻辑
3. AX.EntityFrameworkCore 基于 EntityFrameworkCore 的封装，提供了数据库连接支持，仓储 基础设施层
4. AX.Migrator 数据库迁移工具
5. AX.Web.Core 展现层核心，封装了一些展现层需要用到的工具
6. AX.Web.Host ABP 的动态 Webapi + Swagger
7. AX.Web.MVC 后台页面
- 8 . AX.Tests 单元测试

2、恢复 nuget , 编译解决方案



常见编译错误：“The current .NET SDK does not support targeting .NET Core 2.0. Either target .NET Core 1.1 or lower, or use a version of the .NET SDK that supports .NET Core 2.0.”

解决方案：下载.NET Core 2.0 SDK (<http://www.microsoft.com/net/download/core>)

Overview

.NET Core

.NET Framework

Xamarin

.NET SDKs

.NET Core is the modular and high performance implementation of .NET for creating web applications and services that run on Windows, Linux and Mac. It is open source and it can share the same code with .NET Framework and Xamarin apps. [More...](#)

All .NET Core downloads

SDK

Runtime

.NET Core 2.0 SDK	
Windows (x64) Installer	.exe download
Windows (x64) Binaries	.zip download
Windows (x86) Installer	.exe download
Windows (x86) Binaries	.zip download
macOS (x64) PKG	.pkg download
macOS Binaries	.tar.gz download

3、初始化数据库

3.1、在 AX.Core 项目新建文件夹 “Meetings”

3.2、在 “Meeting” 目录下新建 Meeting.cs

using Abp.Domain.Entities.Auditing;


```

using System;
using System.ComponentModel.DataAnnotations;
namespace AX.Meetings
{
    public class Meeting : FullAuditedEntity<int>
    {
        public const int MaxSubjectLength = 50;
        public const int MaxLocationLength = 200;
        public const int MaxMoreInfoLength = 500;

        [Required]
        [StringLength(MaxSubjectLength)]
        public virtual string Subject { get; set; }
        [Required]
        [StringLength(MaxLocationLength)]
        public virtual string Location { get; set; }
        [Required]
        public virtual DateTime StartTime { get; set; }
        [Required]
        public virtual DateTime EndTime { get; set; }
        [StringLength(MaxMoreInfoLength)]
        public virtual string MoreInfo { get; set; }
    }
}

```

3.3、在 AX.EntityFrameworkCore\EntityFrameworkCore\AXDbContext.cs 添加如下代码

```

using Abp.Zero.EntityFrameworkCore;
using AX.Authorization.Roles;
using AX.Authorization.Users;
using AX.MultiTenancy;
using Microsoft.EntityFrameworkCore;

namespace AX.EntityFrameworkCore
{
    54 references | lijiazeng, 5 days ago | 1 author, 1 change
    public class AXDbContext : AbpZeroDbContext<Tenant, Role, User, AXDbContext>
    {
        /* Define an IDbSet for each entity of the application */
        1 reference | lijiazeng, 5 days ago | 1 author, 1 change | 0 exceptions
        public AXDbContext(DbContextOptions<AXDbContext> options)
            : base(options)
        {
        }

        0 references | lijiazeng, 5 days ago | 1 author, 1 change | 0 exceptions
        public DbSet<Meetings.Meeting> Meetings { get; set; }
    }
}

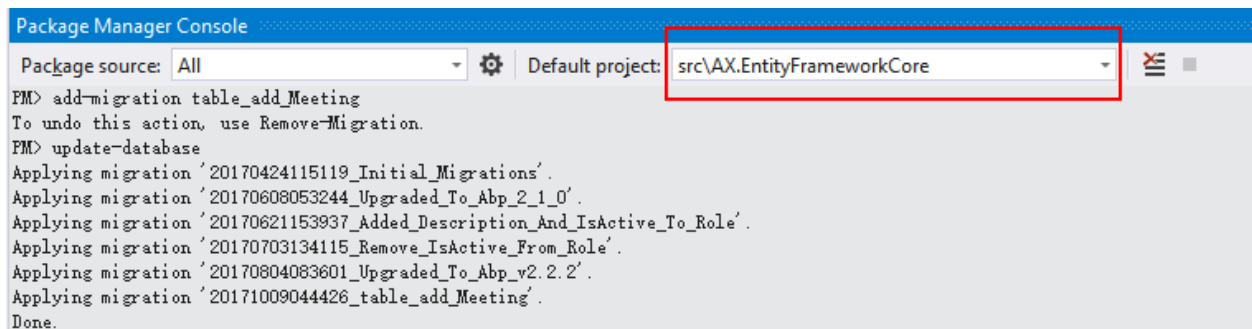
```

Code : `public DbSet<Meetings.Meeting> Meetings { get; set; }`

3.4、将 AX.EntityFrameworkCore 项目设置为启动项目，在 VS 的 Package Manger Console 窗口中使用如下命令动态生成数据库表（SQL Server 默认 Windows 认证）

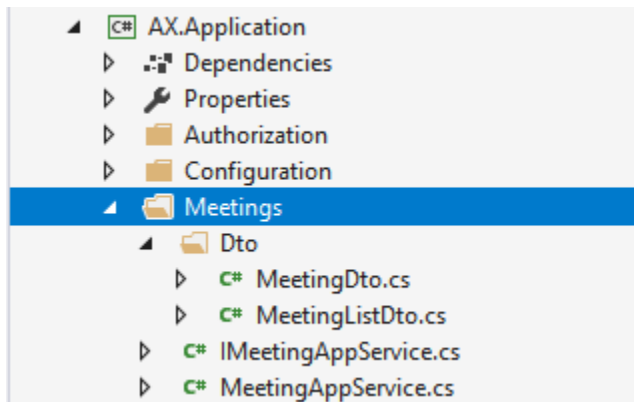
1, add-migration table_add_Meeting ---- 创建迁移

2, update-database ---- 更新数据库



注意：解决方案必须是管理员权限打开，否则再 package manager console 执行命令将失败

4、创建 Meetings in Application



4.1、AX.Application 项目中新建文件夹“ Meetings”

4.2、在 “Meetings” 目录下新建文件夹“ Dto”

4.3、在 “Dto ”目录下新建 MeetingDto.cs、 MeetingListDto.cs

```
using Abp.Application.Services.Dto;
using Abp.AutoMapper;
using System;
using System.ComponentModel.DataAnnotations;
namespace AX.Meetings.Dto
{
    [AutoMap(typeof(Meeting))]
    public class MeetingDto : EntityDto<int>
```

```

{
    [Required]
    [StringLength(50)]
    public string Subject { get; set; }
    [Required]
    [StringLength(200)]
    public string Location { get; set; }
    [Required]
    public DateTime StartTime { get; set; }
    [Required]
    public DateTime EndTime { get; set; }
    [StringLength(500)]
    public string MoreInfo { get; set; }
}
}

```

```

using Abp.Application.Services.Dto;
using Abp.AutoMapper;
using System;
namespace AX.Meetings.Dto
{
    [AutoMap(typeof(Meeting))]
    public class MeetingListDto : EntityDto<int>
    {
        public string Subject { get; set; }
        public string Location { get; set; }
        public DateTime StartTime { get; set; }
        public DateTime EndTime { get; set; }
    }
}

```

4.4、在 “Meetings” 目录下新建 IMeetingAppService.cs、MeetingAppService.cs

```

using Abp.Application.Services;
using Abp.Application.Services.Dto;
using AX.Meetings.Dto;
using System.Threading.Tasks;
namespace AX.Meetings
{
    public interface IMeetingAppService : ICrudAppService<MeetingDto>
    {
        PagedResultDto<MeetingListDto> GetAllList(PagedAndSortedResultRequestDto input);
        Task SendNotification(int id);
    }
}

```

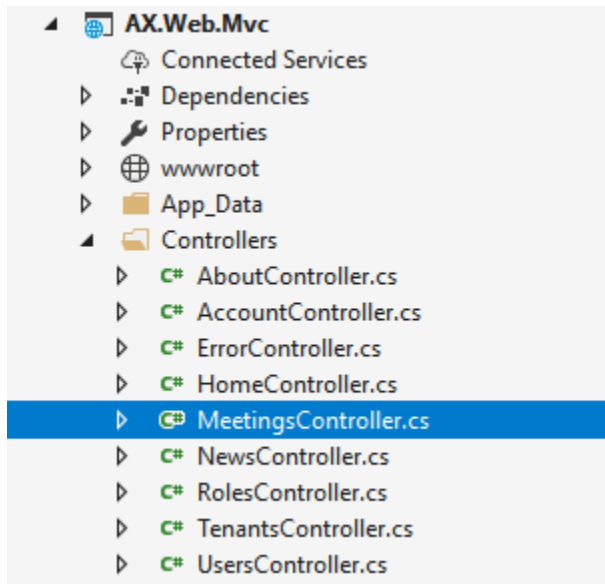
```
}
```

```
using Abp.Application.Services;
using Abp.Application.Services.Dto;
using Abp.Domain.Repositories;
using AutoMapper;
using AX.Meetings.Dto;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace AX.Meetings
{
    public class MeetingAppService : CrudAppService<Meeting, MeetingDto>, IMeetingAppService
    {
        public MeetingAppService(IRepository<Meeting> repository) : base(repository)
        {
        }

        public PagedResultDto<MeetingListDto> GetAllList(PagedAndSortedResultRequestDto input)
        {
            var list = this.Repository.GetAll().OrderBy(x =>
x.CreationTime).Skip(input.SkipCount).Take(input.MaxResultCount);
            return new PagedResultDto<MeetingListDto> { Items = Mapper.Map<List<MeetingListDto>>>(list),
TotalCount = this.Repository.Count() };
        }

        public Task SendNotification(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

5、创建 Meetings in Controllers



5.1、AX.Web.Mvc\Controllers 下新建“ MeetingsController.cs”

```
using Abp.Application.Services.Dto;
using Abp.AspNetCore.Mvc.Authorization;
using AX.Controllers;
using AX.Meetings;
using AX.Meetings.Dto;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Threading.Tasks;

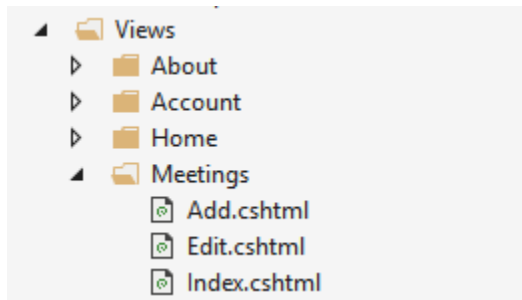
namespace AX.Web.Mvc.Controllers
{
    [AbpMvcAuthorize]
    public class MeetingsController : AXControllerBase
    {
        private readonly IMeetingAppService _meetingAppService;
        public MeetingsController(IMeetingAppService meetingAppService)
        {
            _meetingAppService = meetingAppService;
        }
        public ActionResult Index()
        {
            var model = _meetingAppService.GetAllList(new PagedAndSortedResultRequestDto
            {
                MaxResultCount = 100
            });
            return View(model);
        }
    }
}
```

```

    }
    public ActionResult Add()
    {
        return View(new MeetingDto { StartTime = DateTime.Today.AddDays(1).AddHours(8), EndTime =
DateTime.Today.AddDays(1).AddHours(18) });
    }
    [HttpPost]
    public ActionResult Add(MeetingDto model)
    {
        _meetingAppService.Create(model);
        return RedirectToAction("Index");
    }
    public ActionResult Edit(int id)
    {
        var dto = _meetingAppService.Get(new EntityDto { Id = id });
        return View(dto);
    }
    [HttpPost]
    public ActionResult Edit(MeetingDto model)
    {
        _meetingAppService.Update(model);
        return View(model);
    }
    [HttpPost]
    public JsonResult Delete(int id)
    {
        _meetingAppService.Delete(new EntityDto { Id = id });
        return Json(null);
    }
    [HttpPost]
    public async Task<JsonResult> Notify(int id)
    {
        await _meetingAppService.SendNotification(id);
        return Json(null);
    }
}
}

```

6、创建 Meetings in View



6.1、AX.Web.Mv\Views 下新建文件夹“ Meetings”

6.2、在 “Meetings” 目录下新建 Add.cshtml、Edit.cshtml、Index.cshtml

6.2.1 Add.cshtml

```
@using Abp.Authorization.Users
@using Abp.MultiTenancy
@using AX.MultiTenancy
@using AX.Web.Startup
@model AX.Meetings.Dto.MeetingDto
```

```
@{
    ViewBag.CurrentPageName = PageNames.Meetings; //The menu item will be active for this page.
}
<div class="row clearfix">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <div class="card">
            <div class="header">
                <h2>
                    添加会议
                </h2>
            </div>
            <div class="body table-responsive">
                @using (Html.BeginForm())
                {
                    @Html.AntiForgeryToken()

                    <div class="content">
                        <div class="row clearfix" style="margin-top:10px;">
                            <div class="col-sm-12">
                                <div class="form-group form-float">
                                    <div class="form-line">
                                        @Html.TextBoxFor(m => m.Subject, new { @class = "form-control", required =
"required", maxlength = @AX.Meetings.Meeting.MaxSubjectLength })
                                        <label class="form-label">会议名称</label>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>
</div>
<div class="row clearfix">
    <div class="col-sm-12">
        <div class="form-group form-float">
            <div class="form-line">
                @Html.TextBoxFor(m => m.Location, new { @class = "form-control", required =
"required", maxlength = @AX.Meetings.Meeting.MaxLength })
                <label class="form-label">地点</label>
            </div>
        </div>
    </div>
</div>
<div class="row clearfix">
    <div class="col-sm-6">
        <div class="form-group form-float">
            <div class="form-line">
                @Html.TextBoxFor(m => m.StartTime, new { @class = "form-control", required
= "required", type = "datetime" })
                <label class="form-label">开始时间</label>
            </div>
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group form-float">
            <div class="form-line">
                @Html.TextBoxFor(m => m.EndTime, new { @class = "form-control", required =
"required", type = "datetime" })
                <label class="form-label">结束时间</label>
            </div>
        </div>
    </div>
</div>
</div>
<div class="footer pull-right">
    <a class="btn btn-default waves-effect" href="@Url.Action("Index")">@L("Cancel")</a>
    <button type="submit" class="btn btn-primary waves-effect">@L("Save")</button>
</div>
}
</div>
</div>
</div>

```



```
</div>
```

6.2.2 Edit.cshtml

```
@using Abp.Authorization.Users
@using Abp.MultiTenancy
@using AX.MultiTenancy
@using AX.Web.Startup
@model AX.Meetings.Dto.MeetingDto
```

```
@{
    ViewBag.CurrentPageName = PageNames.Meetings; //The menu item will be active for this page.
}
<div class="row clearfix">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <div class="card">
            <div class="header">
                <h2>
                    编辑会议
                </h2>
            </div>
            <div class="body table-responsive">
                @using (Html.BeginForm())
                {
                    @Html.AntiForgeryToken()
                    @Html.HiddenFor(x => x.Id)

                    <div class="content">
                        <div class="row clearfix" style="margin-top:10px;">
                            <div class="col-sm-12">
                                <div class="form-group form-float">
                                    <div class="form-line">
                                        @Html.TextBoxFor(m => m.Subject, new { @class = "form-control", required =
"required", maxlength = @AX.Meetings.Meeting.MaxSubjectLength })
                                        <label class="form-label">会议名称</label>
                                    </div>
                                </div>
                            </div>
                        </div>
                        <div class="row clearfix">
                            <div class="col-sm-12">
                                <div class="form-group form-float">
                                    <div class="form-line">
                                        @Html.TextBoxFor(m => m.Location, new { @class = "form-control", required =
"required", maxlength = @AX.Meetings.Meeting.MaxLocationLength })
                                        <label class="form-label">地点</label>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>
</div>
<div class="row clearfix">
    <div class="col-sm-6">
        <div class="form-group form-float">
            <div class="form-line">
                @Html.TextBoxFor(m => m.StartTime, new { @class = "form-control", required
= "required", type = "datetime" })
                <label class="form-label">开始时间</label>
            </div>
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group form-float">
            <div class="form-line">
                @Html.TextBoxFor(m => m.EndTime, new { @class = "form-control", required =
"required", type = "datetime" })
                <label class="form-label">结束时间</label>
            </div>
        </div>
    </div>
</div>

<div class="footer pull-right">
    <a class="btn btn-default waves-effect" href="@Url.Action("Index")">@L("Cancel")</a>
    <button type="submit" class="btn btn-primary waves-effect">@L("Save")</button>
</div>
}

</div>
</div>
</div>
</div>

```

6.2.3 Index.cshtml

```

@using Abp.Authorization.Users
@using Abp.MultiTenancy
@using AX.MultiTenancy
@using AX.Web.Startup
@model Abp.Application.Services.Dto.PagedResultDto<AX.Meetings.Dto.MeetingListDto>
@{

```

```
ViewBag.CurrentPageName = PageNames.Meetings; //The menu item will be active for this page.
```

```
}
@section scripts
{
    <environment>
        <script src="~/view-resources/Views/Meetings/index.js" asp-append-version="true"></script>
    </environment>
}

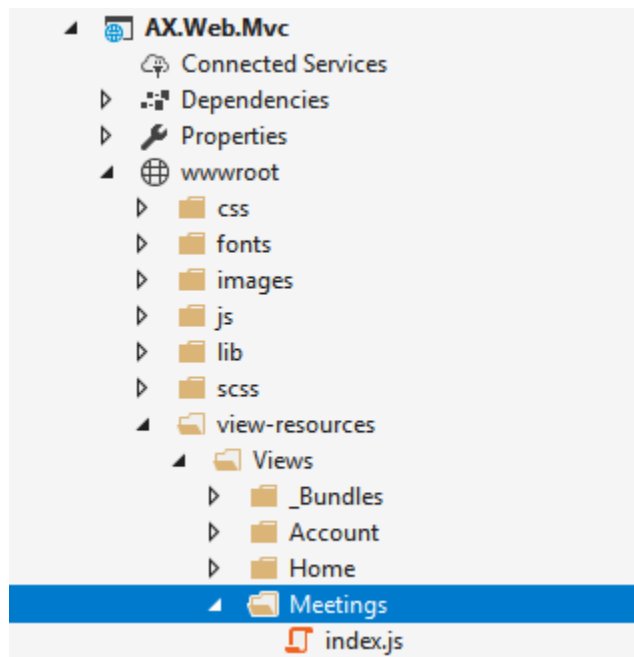
<div class="row clearfix">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <div class="card">
            <div class="header">
                <h2>
                    会议
                </h2>
            </div>
            <div class="body table-responsive">
                <table class="table">
                    <thead>
                        <tr>
                            <th>会议名称</th>
                            <th>地点</th>
                            <th>开始时间</th>
                            <th>结束时间</th>
                            <th>@L("Actions")</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var meeting in Model.Items)
                        {
                            <tr>
                                <td>@meeting.Subject</td>
                                <td>@meeting.Location</td>
                                <td>@meeting.StartTime</td>
                                <td>@meeting.EndTime</td>
                                <td class="dropdown">
                                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false">
                                        <i class="material-icons">menu</i>
                                    </a>
                                    <ul class="dropdown-menu pull-right">
                                        <li><a href="@Url.Action("Edit",new { id = meeting.Id})" class="waves-effect
waves-block"><i class="material-icons">edit</i> @L("Edit")</a></li>
                                        <li><a href="#" onclick="notify('@meeting.Id')" class="waves-effect waves-
block"><i class="material-icons">message</i> @L("Notify")</a></li>
                                    </ul>
                                </td>
                            </tr>
                        }
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>
```

```

        </td>
    </tr>
}
</tbody>
</table>
<a class="btn btn-primary btn-circle waves-effect waves-circle waves-float pull-right"
href="@Url.Action("Add")">
    <i class="material-icons">add</i>
</a>
</div>
</div>
</div>

```

6.3、添加 JS



在 AX.Web.Mvc\wwwroot\view-resources\Views 下新建文件夹“ Meetings” ,并添加 index.js

```

Code : function notify(id) {
    $.ajax({
        type: "POST",
        url: "/Meetings/Notify/" + id,
        contentType: "application/json; charset=utf-8",
        error: function () {
            alert("服务器错误!");
        },
        success: function (result) {
            alert("已发送推送!");
        }
    });
}

```

```
    },  
    dataType: "json"  
  });  
}
```

7、修改 PageNames

在 AX.Web.Mvc\Startup\PageNames.cs 中添加如下代码

```
namespace AX.Web.Startup  
{  
    7 references | lijiazeng, 5 days ago | 1 author, 1 change  
    public class PageNames  
    {  
        public const string Home = "Home";  
        public const string About = "About";  
        public const string Tenants = "Tenants";  
        public const string Users = "Users";  
        public const string Roles = "Roles";  
        public const string Meetings = "Meetings";  
    }  
}
```

Code : public const string Meetings = "Meetings";

8、添加导航

在 AX.Web.Mvc\Startup\AXNavigationProvider.cs 添加如下代码

```

public class AXNavigationProvider : NavigationProvider
{
    0 references | 0 exceptions
    public override void SetNavigation(INavigationProviderContext context)
    {
        context.Manager.MainMenu
            .AddItem(
                new MenuItemDefinition(
                    PageNames.Home,
                    L("HomePage"),
                    url: "",
                    icon: "home",
                    requiresAuthentication: true
                )
            ).AddItem(
                new MenuItemDefinition(
                    PageNames.Meetings,
                    L("Meetings"),
                    url: "Meetings",
                    icon: "event",
                    requiresAuthentication: true
                )
            ).AddItem(
                new MenuItemDefinition(
                    PageNames.Tenants,
                    L("Tenants"),
                    url: "Tenants",
                    icon: "business",
                    requiredPermissionName: PermissionNames.Pages_Tenants
                )
            )
    }
}

```

```

Code : .AddItem(
    new MenuItemDefinition(
        PageNames.Meetings,
        L("Meetings"),
        url: "Meetings",
        icon: "event",
        requiresAuthentication: true
    )
)

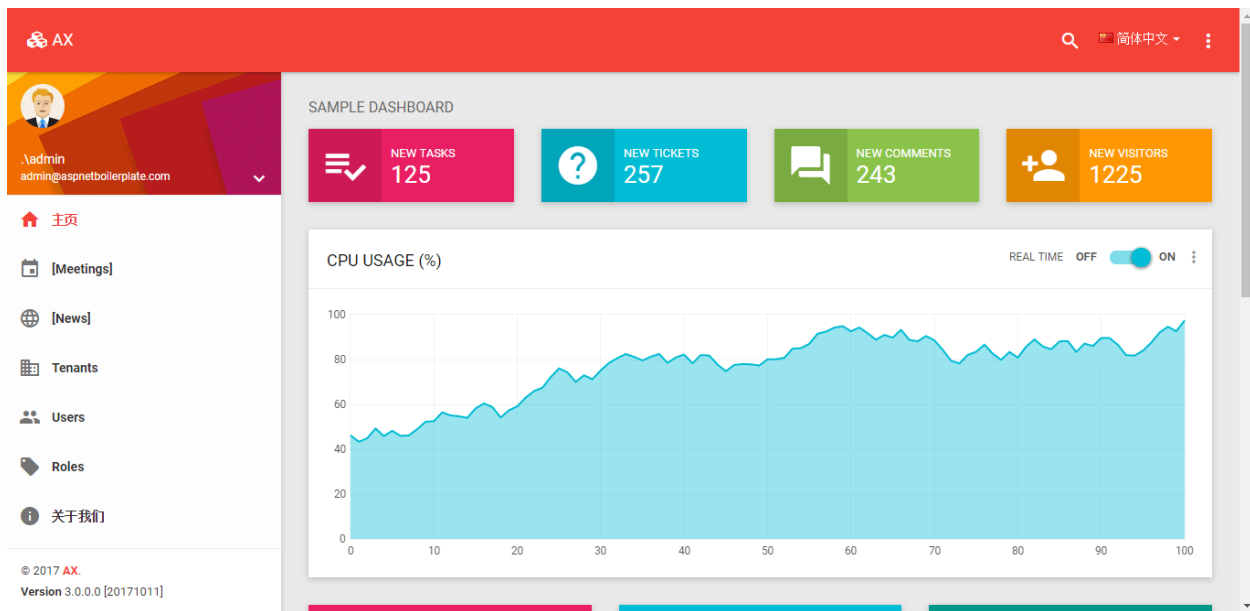
```

9、运行程序

设置 AX.Web.Mvc 项目为启动项

账号 : admin

密码 : 123qwe



添加样例会议数据

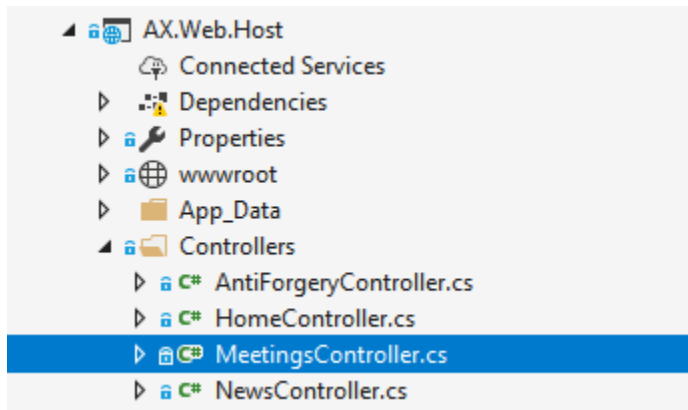
The '编辑会议' (Edit Meeting) form is displayed within the same dashboard context. It contains the following fields:

- 会议名称 (Meeting Name): 2017微软技术暨生态大会
- 地点 (Location): 北京国际饭店会议中心
- 开始时间 (Start Time): 2017/10/31 0:00:00
- 结束时间 (End Time): 2017/11/3 0:00:00

At the bottom right of the form are 'Cancel' and 'Save' buttons.

API

1、创建 Meetings in Controllers



1.1、AX.Web.Host\Controllers 下新建 “MeetingsController.cs”

```
using Abp.Application.Services.Dto;
using AX.Controllers;
using AX.Meetings;
using AX.Meetings.Dto;
using Microsoft.AspNetCore.Mvc;
using Swashbuckle.AspNetCore.SwaggerGen;

namespace AX.Web.Host.Controllers
{
    [Route("api/meeting")]
    public class MeetingsController : AXControllerBase
    {
        private readonly IMeetingAppService _meetingAppService;
        public MeetingsController(IMeetingAppService meetingAppService)
        {
            _meetingAppService = meetingAppService;
        }
        [Route("list")]
        [HttpGet]
        [SwaggerResponse((int)System.Net.HttpStatusCode.OK, Type = typeof(PagedResultDto<MeetingListDto>))]
        public IActionResult List(int pageIndex = 0)
        {
            var result = _meetingAppService.GetAll(new PagedAndSortedResultRequestDto { MaxResultCount = 10,
SkipCount = pageIndex * 10 });
            return Ok(result);
        }
        [Route("get/{id}")]
        [HttpGet]
```



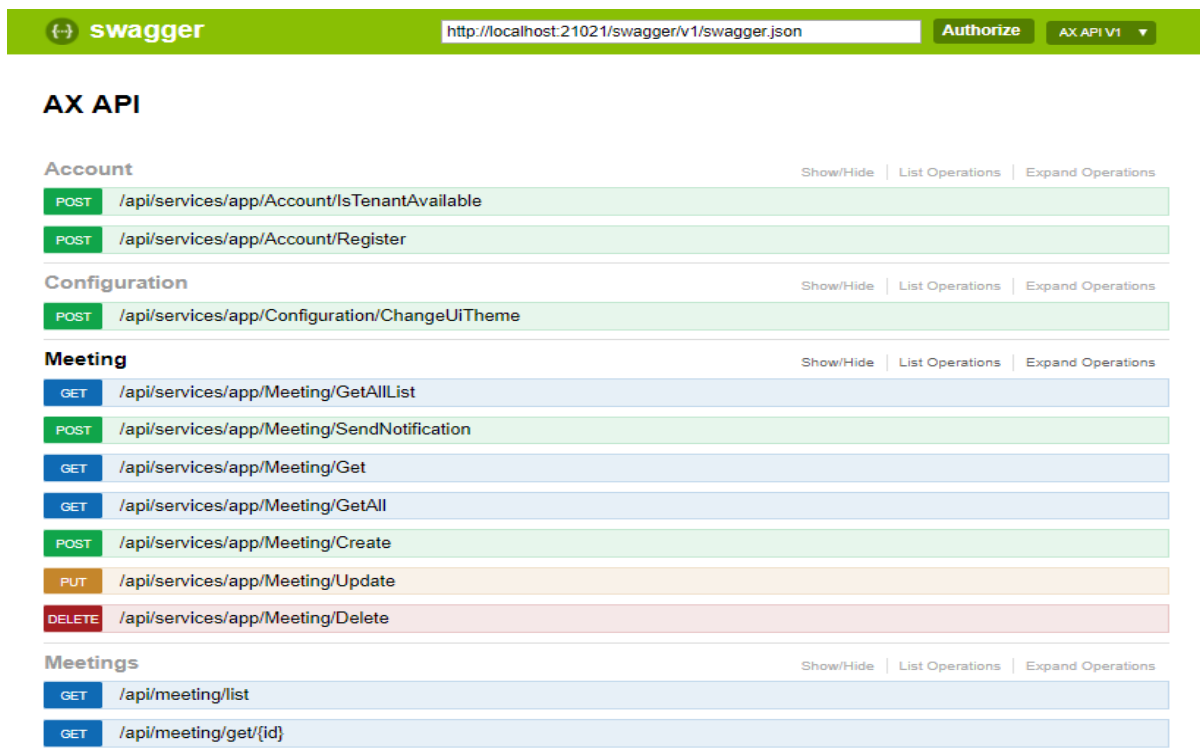
```

[SwaggerResponse((int)System.Net.HttpStatusCode.OK, Type = typeof(MeetingDto))]
[SwaggerResponse((int)System.Net.HttpStatusCode.NotFound)]
public IActionResult Get(int id)
{
    var result = _meetingAppService.Get(new EntityDto { Id = id });
    return Ok(result);
}
}
}

```

2、运行程序

设置 AX.Web.Host 项目为启动项



The image shows the Swagger UI for the AX API. At the top, there is a green header bar with the Swagger logo, a text input field containing the URL 'http://localhost:21021/swagger/v1/swagger.json', an 'Authorize' button, and a dropdown menu showing 'AX API V1'. Below the header, the title 'AX API' is displayed. The main content area lists several API endpoints grouped into categories: 'Account', 'Configuration', 'Meeting', and 'Meetings'. Each category has a 'Show/Hide', 'List Operations', and 'Expand Operations' link. The 'Account' category lists two POST endpoints: '/api/services/app/Account/IsTenantAvailable' and '/api/services/app/Account/Register'. The 'Configuration' category lists one POST endpoint: '/api/services/app/Configuration/ChangeUITheme'. The 'Meeting' category lists seven endpoints: a GET endpoint '/api/services/app/Meeting/GetAllList', a POST endpoint '/api/services/app/Meeting/SendNotification', a GET endpoint '/api/services/app/Meeting/Get', a GET endpoint '/api/services/app/Meeting/GetAll', a POST endpoint '/api/services/app/Meeting/Create', a PUT endpoint '/api/services/app/Meeting/Update', and a DELETE endpoint '/api/services/app/Meeting/Delete'. The 'Meetings' category lists two GET endpoints: '/api/meeting/list' and '/api/meeting/get/{id}'.

Try it out 以下两个接口

api/meeting/list

api/meeting/get/{id}



构建前后端跨平台的移动 APP 架构 之基于 Xamarin 的跨平台 App

目录

实验目的.....	1
前提条件.....	1
下载项目模板代码:.....	1
检查 Android 7.0(或以上版本是否安装)	2
c:检查 Android SDK 版本是否为 7.0 以上.....	3
设备调试前提:	4
实验步骤.....	5
01:恢复 Nuget 包.....	5
02:检测 AXMobile 是否可以通过编译	5
03:创建 LoginPage.Xaml 页面	7
a:创建 LoginPage 页面:	7
b:添加页面代码:.....	9
c:为 LoginPage 页面添加 LoginViewModel:.....	14
d:设置 LoginPage 为启动页面:	17
e:为页面添加图片素材:	18
f:隐藏登录界面导航栏:	22
04:创建 MainPage.Xaml 页面	23
实现从 LoginPage 跳转到 MainPage:	23
05:添加会议列表信息:	27
a:添加获取 Web API 数据功能:.....	27
b:创建并修改 MeetingListPage.xaml 页面代码并且展示从 API 上获取的数据:	37
06: 完成所有步骤后, 选择对应的设备运行 App 显示结果.....	43

实验目的

使用 Xamarin.Forms 技术实现同一套代码部署到 iOS 和 Android 平台，并分别运行项目得到最终结果。

前提条件

下载项目模板代码：

起始模板：https://projectax.blob.core.chinacloudapi.cn/src/AXMobile_01Start.zip

参考代码：https://projectax.blob.core.chinacloudapi.cn/src/AXMobile_02Full.zip

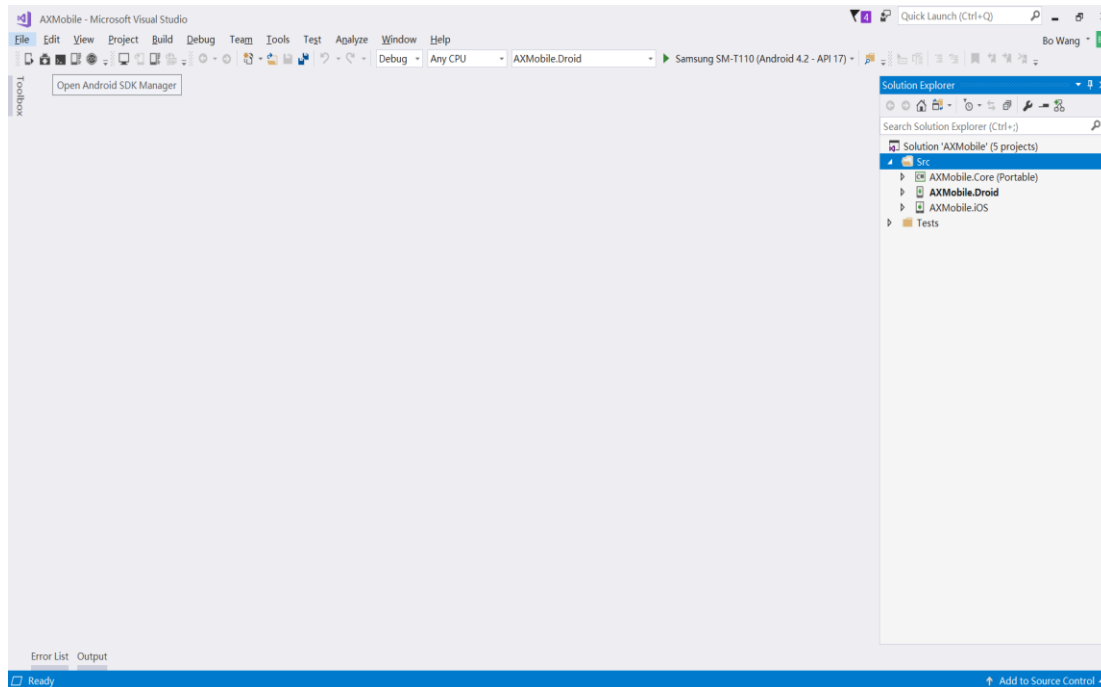
素材资源：<https://projectax.blob.core.chinacloudapi.cn/src/Res.zip>

接受培训者请下载起始模板即可开始动手操作，参考代码作为本项目的完整代码，可供接受培训者在使用起始模板操作时当作样例，素材资源为设计页面 UI 准备，若没有素材编译则会报错，个关于素材使用在下边文章中会详细解释。

注：如果按本教程文档操作出现编译报错问题，请右键项目选择清理然后在生成，如果生成成功则继续往下操作，没有的话若自己不能解决请找当天在场老师帮助。

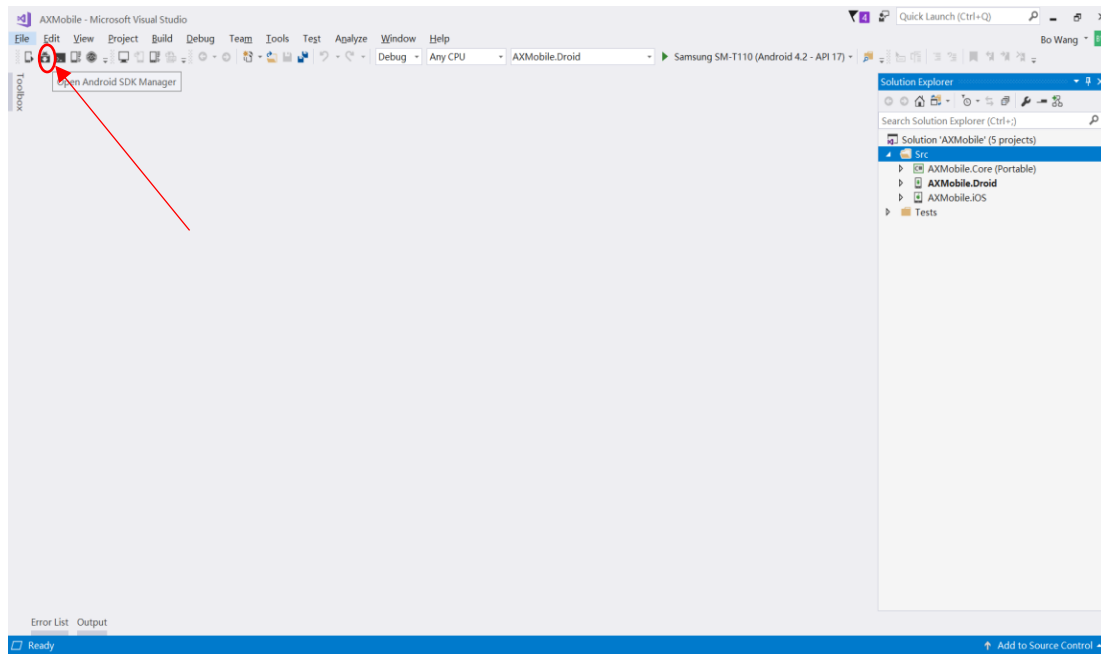
检查 Android 7.0(或以上版本是否安装)

a:从模板的下载目录下解压后并打开 AXMobile.sln 文件

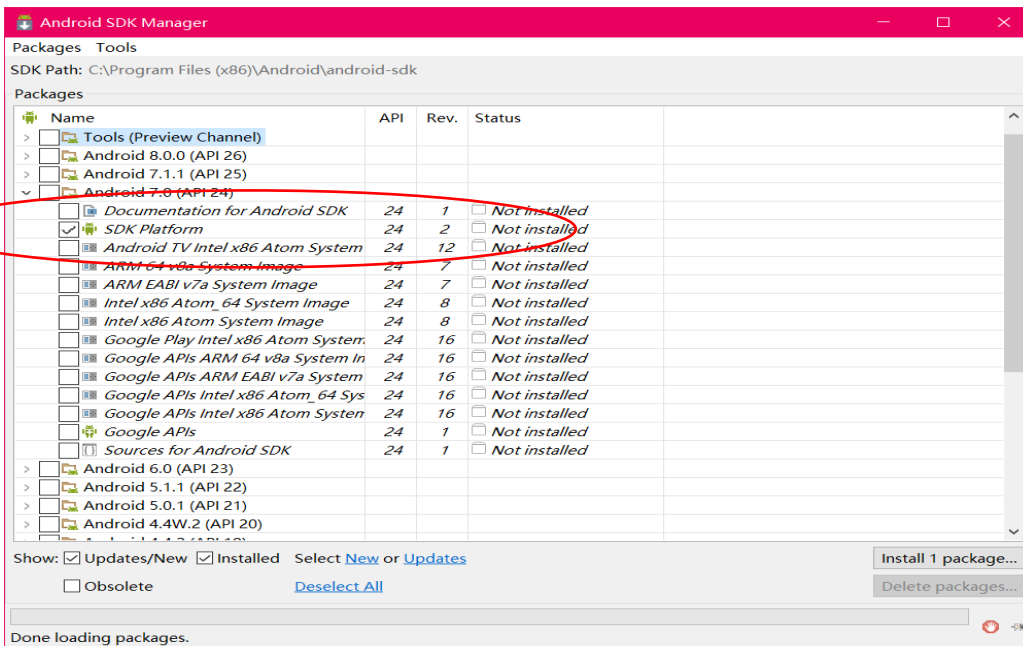


注：若出现一个或多个项目目标不支持生成操作”EmbeddedResource”这个错误，请不用管他，右击项目生成成功就可以通过(具体操作请接着本文档继续往下操作)。

b: 打开 Android 管理工具



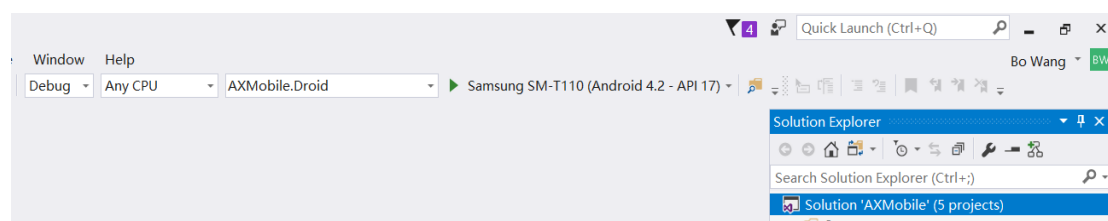
c: 检查 Android SDK 版本是否为 7.0 以上



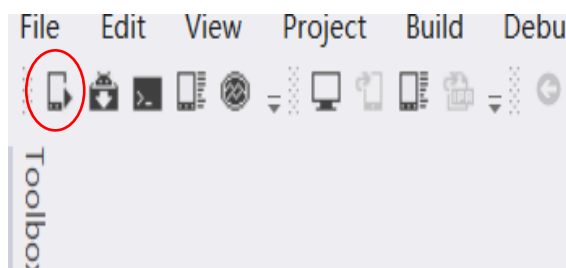
设备调试前提:

iOS:编译或部署 iOS App 需要在 Xcode 环境下编译或部署 (iOS 无法直接真机调试), 如果没有此环境则考虑使用 Android 环境查看结果。

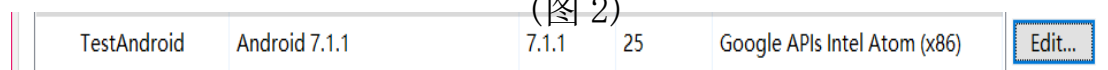
Android:使用模拟器 (图 2) 或真机调试 (图 1) (选择图 3 所示模拟器并运行), 真机调试只需要插入数据线至 PC 以连接手机和 PC, 之后在 Android 系统的手机设置中打开开发者模式, 并且打开 USB 调试, 之后 Visual Studio (绿色三角箭头) 后的设备名称变为你接入 PC 端的手机名称, 在此之后即可以使用 Android 进行真机调试; 模拟器则点击如图 2 至图 3 所示 (如果 AVD 模拟器无法使用则建议改为真机调试)。



(图 1)



(图 2)

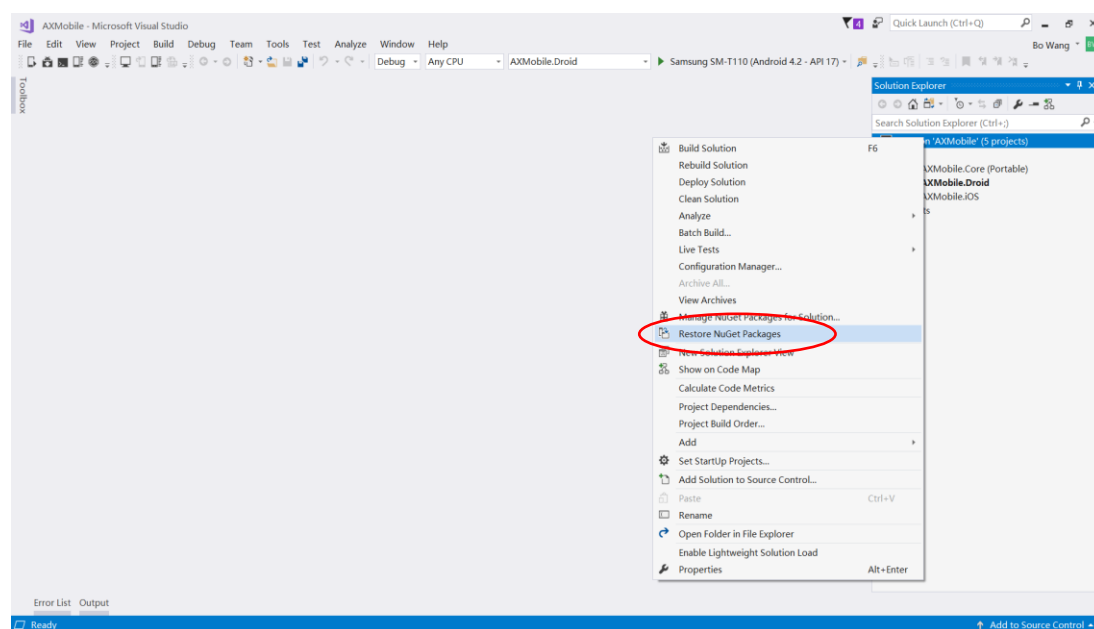


(图 3)

实验步骤

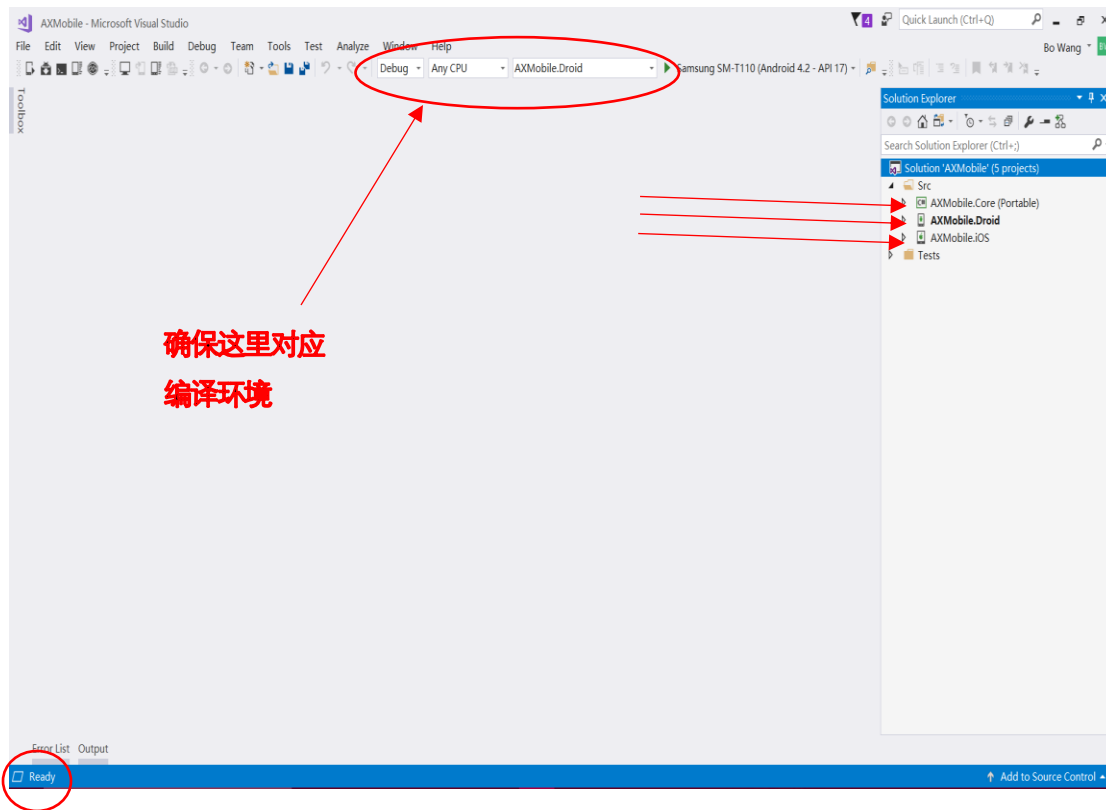
01:恢复 Nuget 包

在打开 AXMobile 项目后，在右边找到 Solution Explorer(解决方案管理器)栏目(如果没有 Solution Explorer 则在工具栏的 View(视图)中找到单击添加)，在鼠标右击 Solution Explorer 后在下边的列表中找到 Restore Nuget Packages(还原 Nuget 包) 点击后查看 OutPut 输出栏(没有的话也在工具栏的 View 中添加)运行后显示 Finish 表示恢复 Nuget 包完成。



02:检测 AXMobile 是否可以通过编译

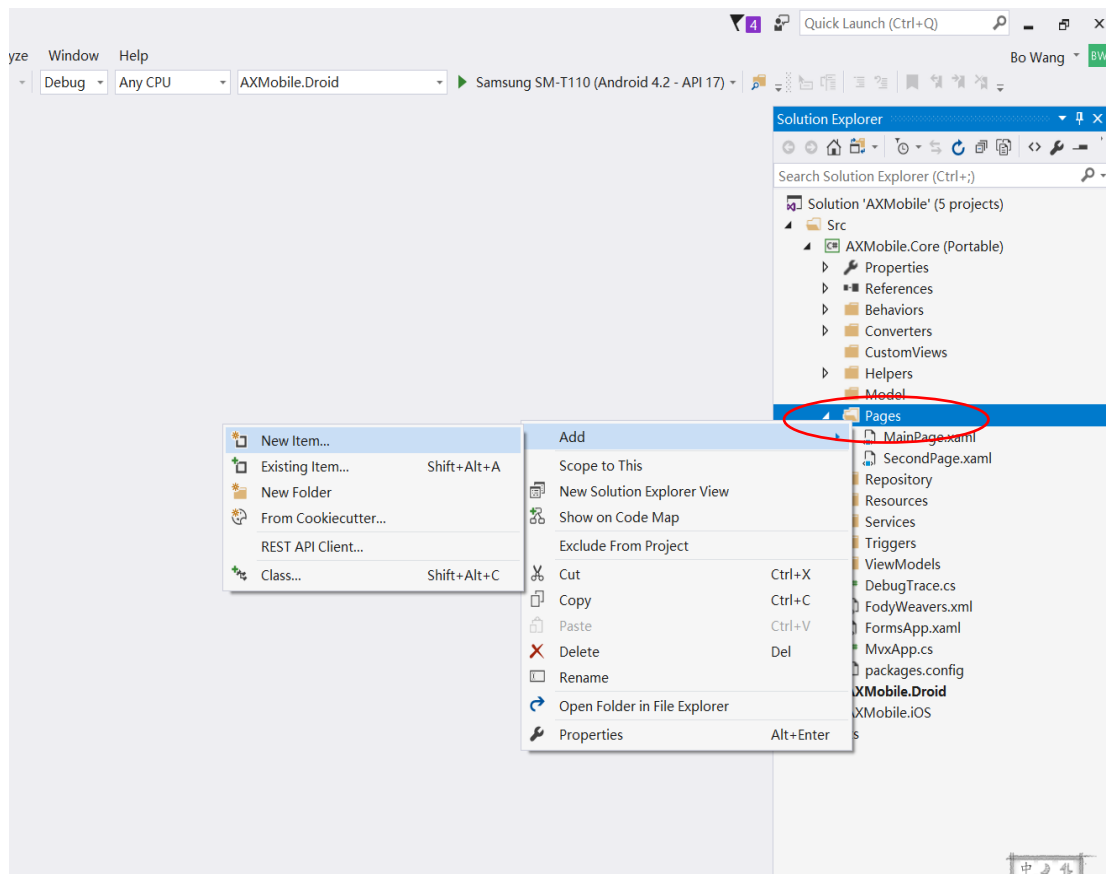
分别右击这三个红色箭头指向的项目，找到 Clean(清理)，之后再分别并点击 Rebuild(重新生成)，在 Visual Studio 运行 Rebuild 之后在左下角显示 Build All succeeded(生成成功)，则表示检测 AXMobile 项目通过。



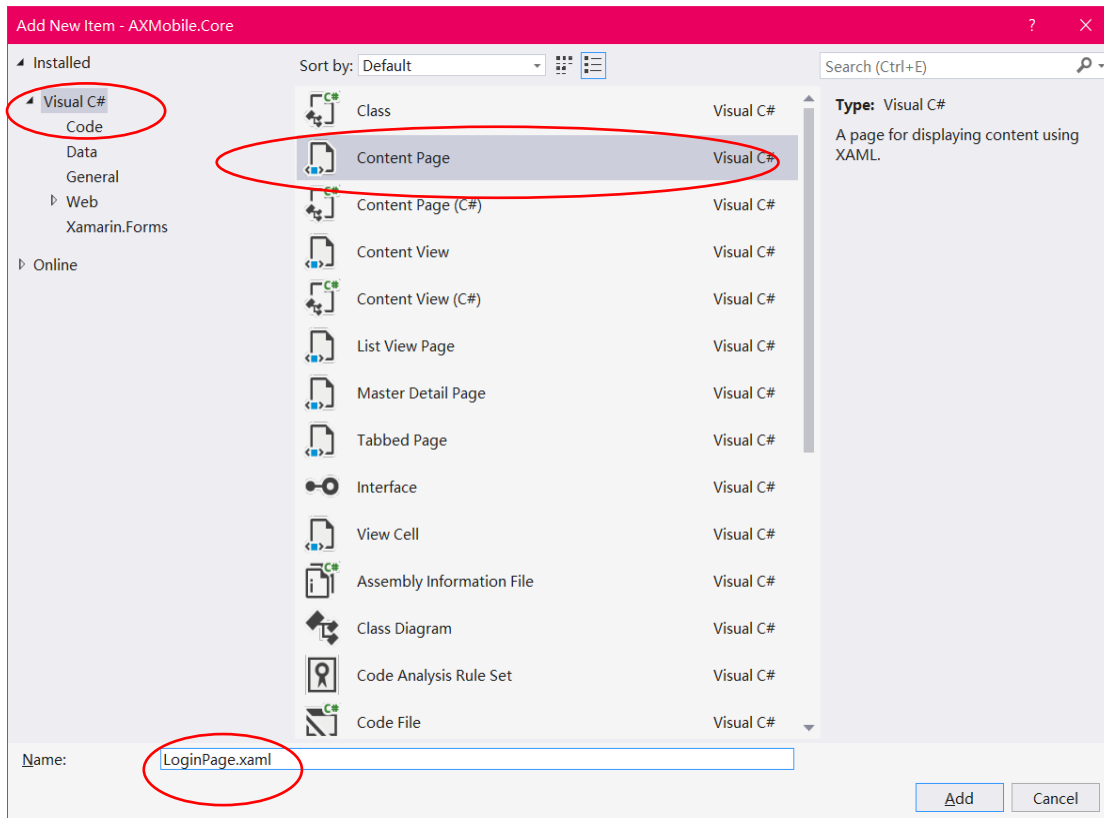
注：在 Rebuild Android 或 iOS 时，确保他们处于相应的环境下编译，改变环境按图中红圈处更改。

03:创建 LoginPage.Xaml 页面

a:创建 LoginPage 页面:



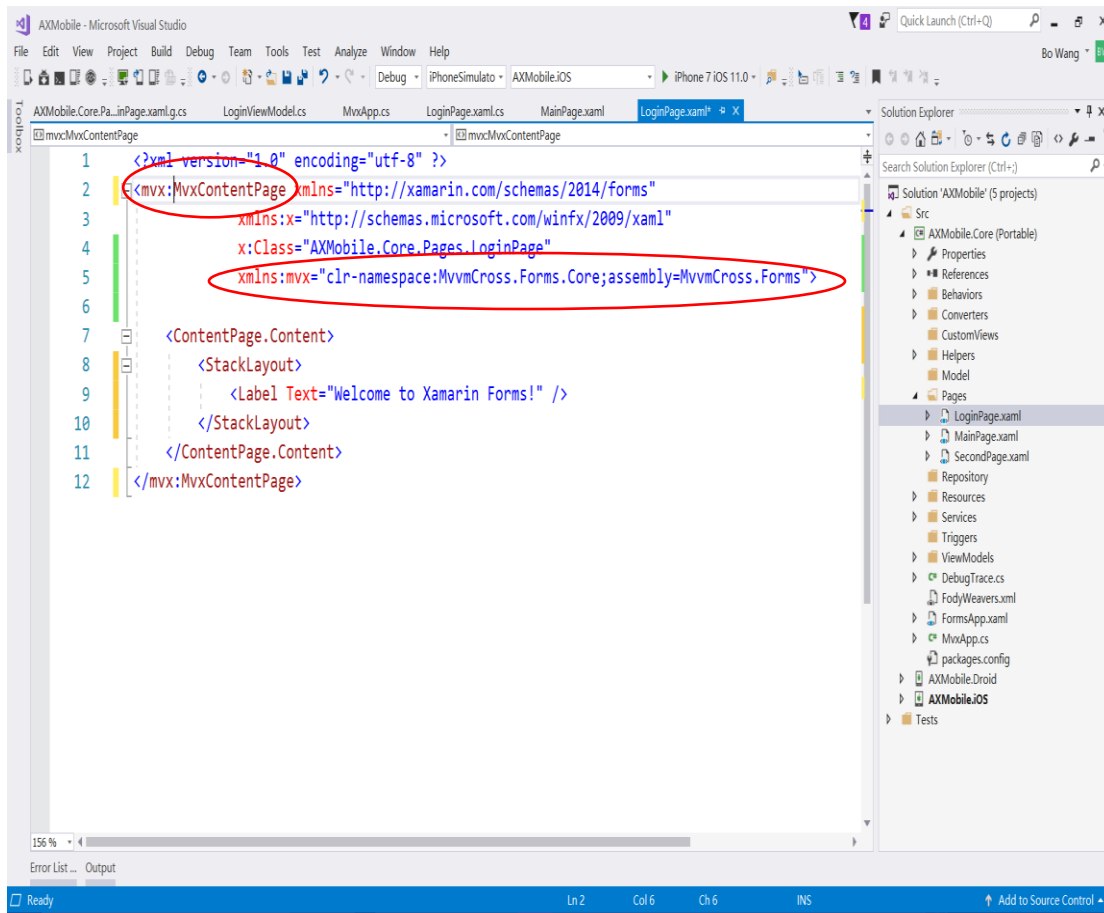
如图所示，在 Core 项目中找到 Pages 文件夹，之后右击鼠标选择 Add(添加)再选择 New Item(新建项)



点击 New Item 后进入如图对话框，并进行以下操作：

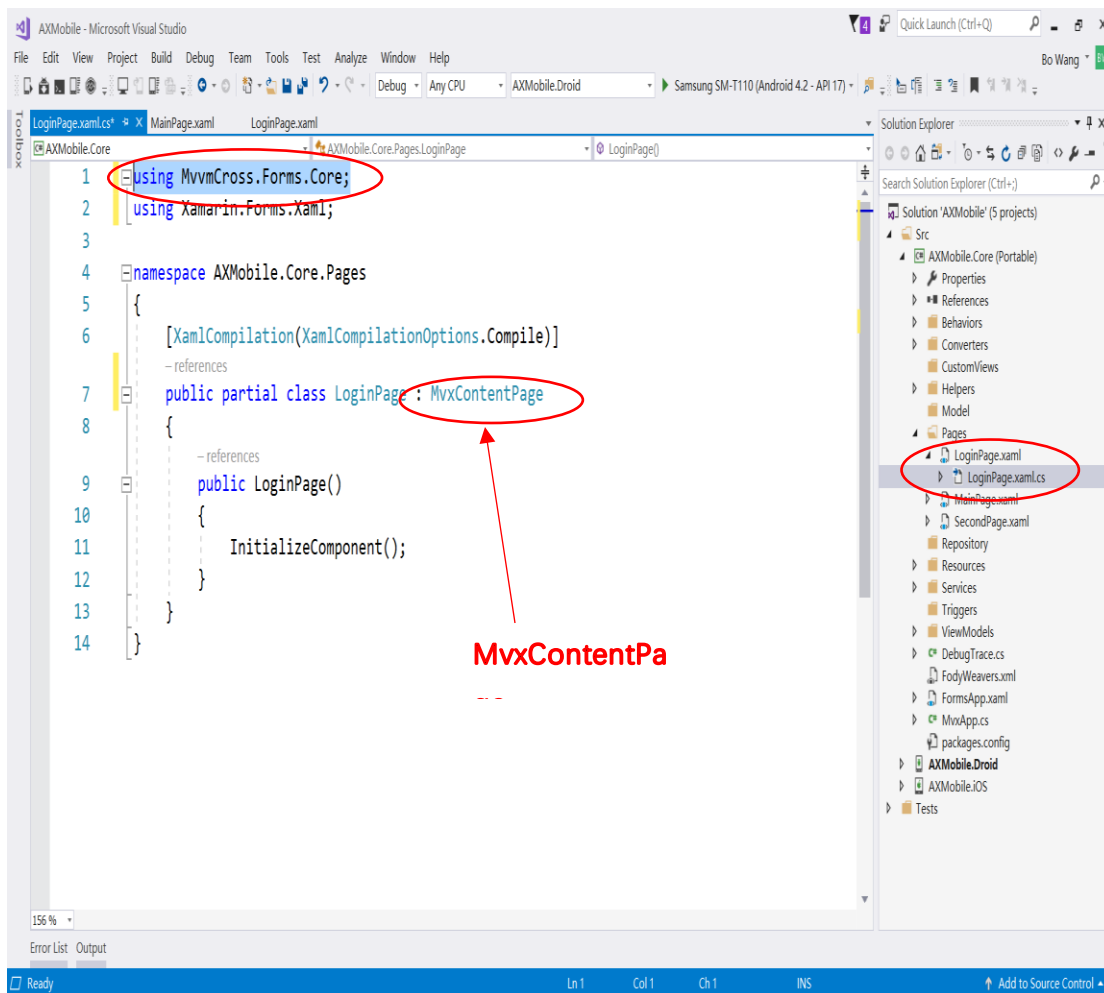
- 1、在对话框中的左边列表选择 Visual C#。
- 2、在中间的列表中选择 Content Page。
- 3、在下边的 Name (名称) 中把名称改为 LoginPage.xaml

b:添加页面代码:



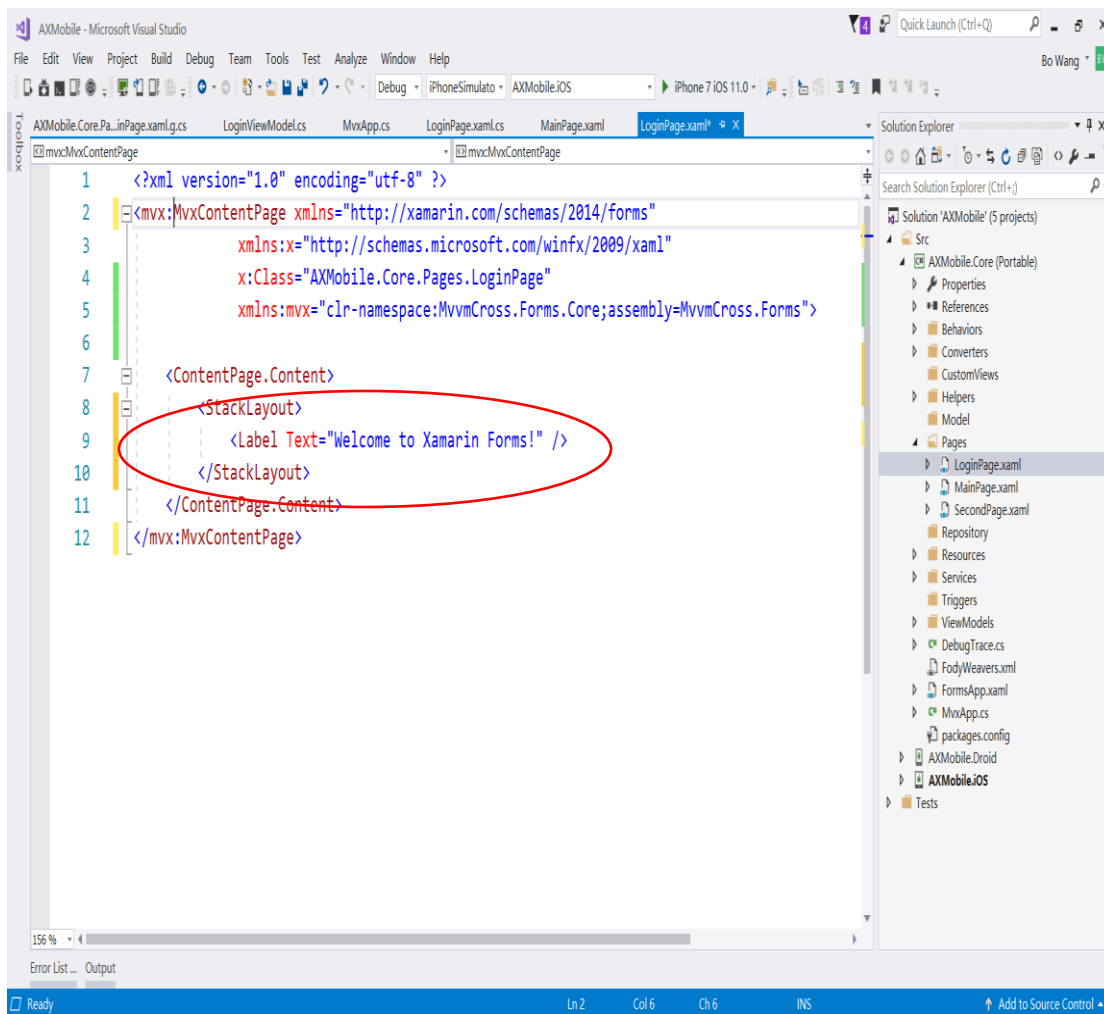
按此图中展示的 Xaml 代码进行添加和修改,修改完成后打开 LoginPage.Xaml 项目,在打开后的列表中双击 LoginPage.Xaml.cs。

注:左上角代码为: `mvx:MvxContentPage`, 下边圈里的代码为: `xmlns:mvx="clr-namespace:MvvmCross.Forms.Core;assembly=MvvmCross.Forms"`



打开 LoginPage.xaml.cs 后按如图方式修改并且添加代码。

注：AXMobile 用到了 MvvmCross 框架，但本文档不详细叙述。



回到 LoginPage.Xaml 文件中，在<ContentPage.Content>.....
<ContentPage .Content >中使用以下代码覆盖此段代码。

```

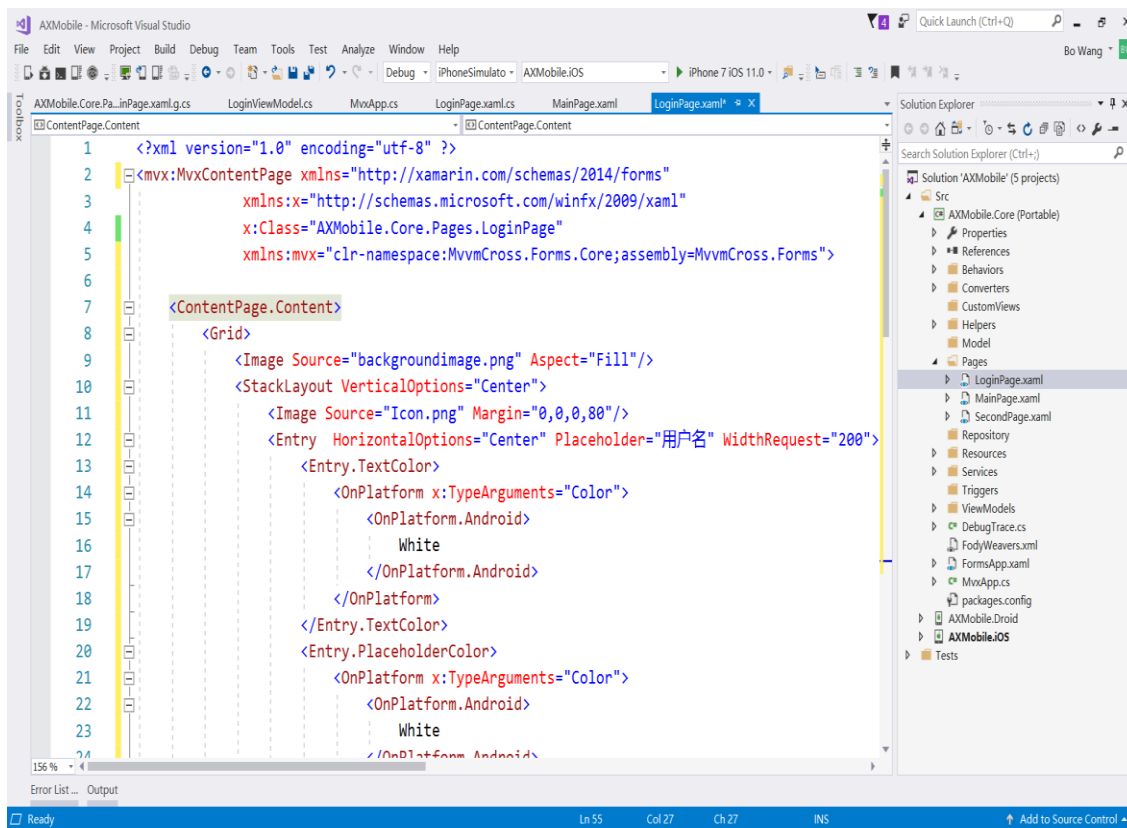
<ContentPage.Content>
  <Grid>
    <Image Source="backgroundimage.png" Aspect="Fill"/>
    <StackLayout VerticalOptions="Center">
      <Image Source="Icon.png" Margin="0, 0, 0, 80"/>
      <Entry HorizontalOptions="Center" Placeholder="用户名" WidthRequest="200">
        <Entry.TextColor>
          <OnPlatform x:TypeArguments="Color">
            <OnPlatform.Android>
              White
            </OnPlatform.Android>
          </OnPlatform>
        </Entry.TextColor>
        <Entry.PlaceholderColor>

```

```

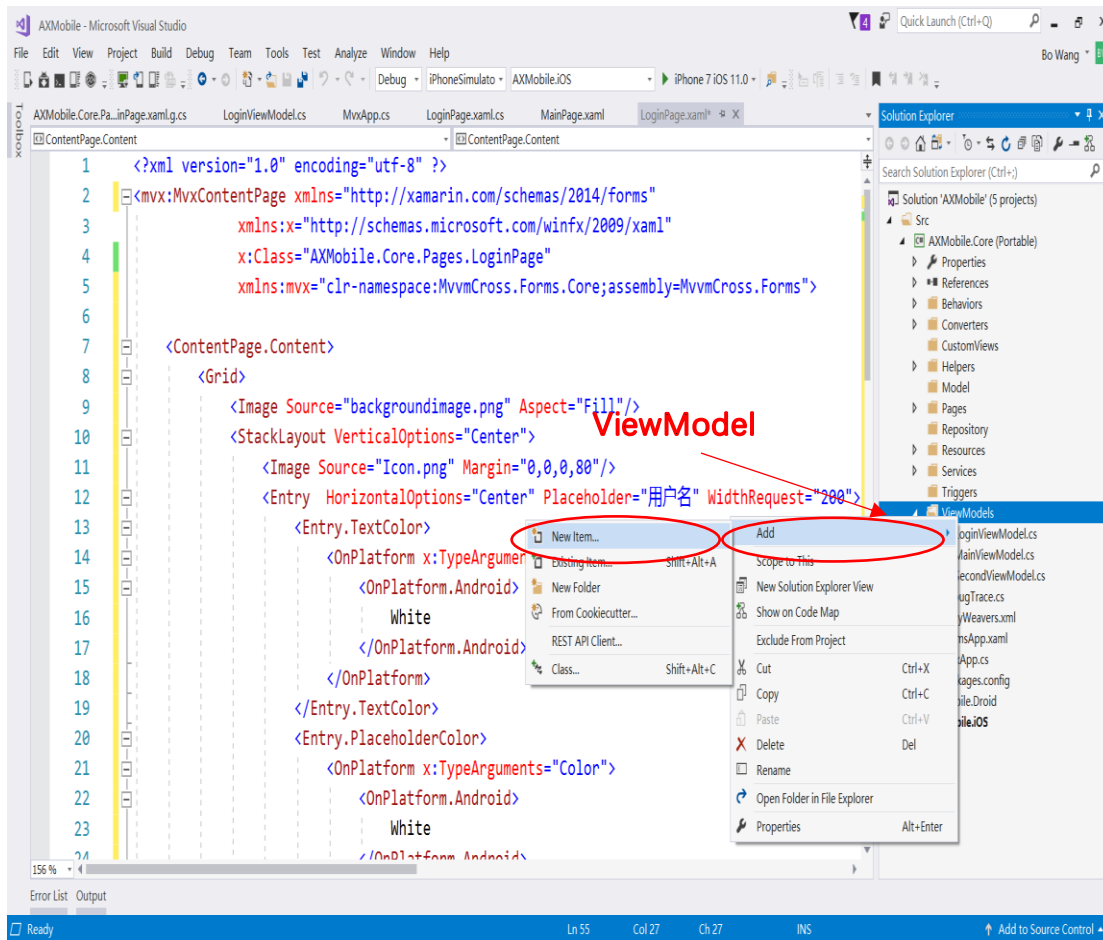
        <OnPlatform x:TypeArguments="Color">
            <OnPlatform.Android>
                White
            </OnPlatform.Android>
        </OnPlatform>
    </Entry.PlaceholderColor>
</Entry>
<Entry HorizontalOptions="Center" Placeholder="密码" WidthRequest="200">
    <Entry.TextColor>
        <OnPlatform x:TypeArguments="Color">
            <OnPlatform.Android>
                White
            </OnPlatform.Android>
        </OnPlatform>
    </Entry.TextColor>
    <Entry.PlaceholderColor>
        <OnPlatform x:TypeArguments="Color">
            <OnPlatform.Android>
                White
            </OnPlatform.Android>
        </OnPlatform>
    </Entry.PlaceholderColor>
</Entry>
    <Button BackgroundColor="#1976d2" HorizontalOptions="Center" WidthRequest="120"
Text="登录" TextColor="White" Margin="0,20,0,150" Command="{Binding GoToMainPage}" />
</StackLayout>
    <StackLayout Orientation="Horizontal" VerticalOptions="EndAndExpand" Margin="0,0,0,15"
HorizontalOptions="CenterAndExpand">
        <Image Source="logo.png" WidthRequest="25" HeightRequest="25" />
        <Label Text="http://www.highviewsoft.com" TextColor="White" Margin="5,0,0,0">
            <Label.GestureRecognizers>
                <TapGestureRecognizer Command="{Binding OpenGithubUrlCommand}" />
            </Label.GestureRecognizers>
        </Label>
    </StackLayout>
</Grid>
</ContentPage.Content>

```

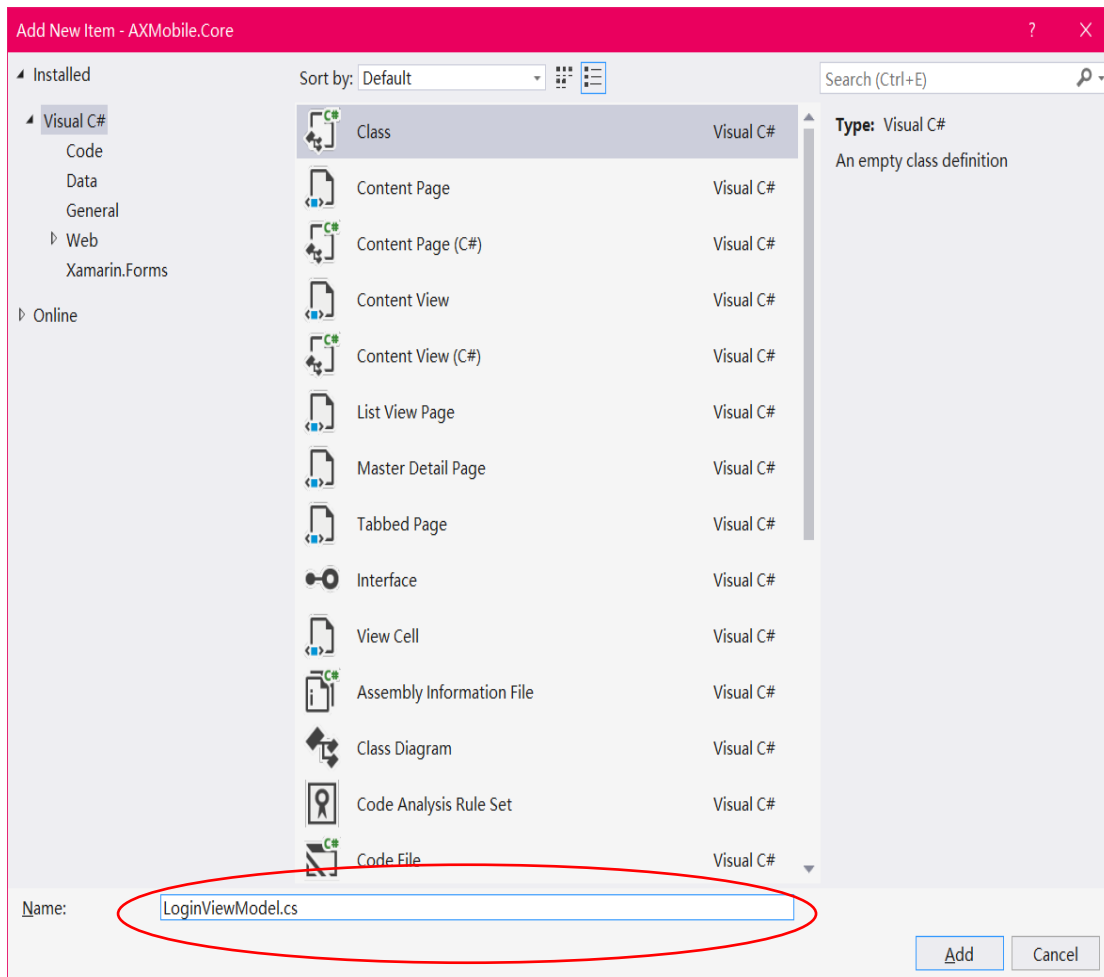


(添加代码后效果如图)

c:为 LoginPage 页面添加 LoginViewModel:



在 AXMobile.Core 项目中找到 ViewModels 文件夹，鼠标右键按图所示添加一个 Class (类)。



在 Class 对话框中选择 Visual C# 并且在中间的列表中选择 Class(类)，并将对话框中底下的 Name 框中的值改为 LoginViewModel，之后点击 Add(添加)以完成添加操作。

在 LoginViewModel 中添加以下代码：

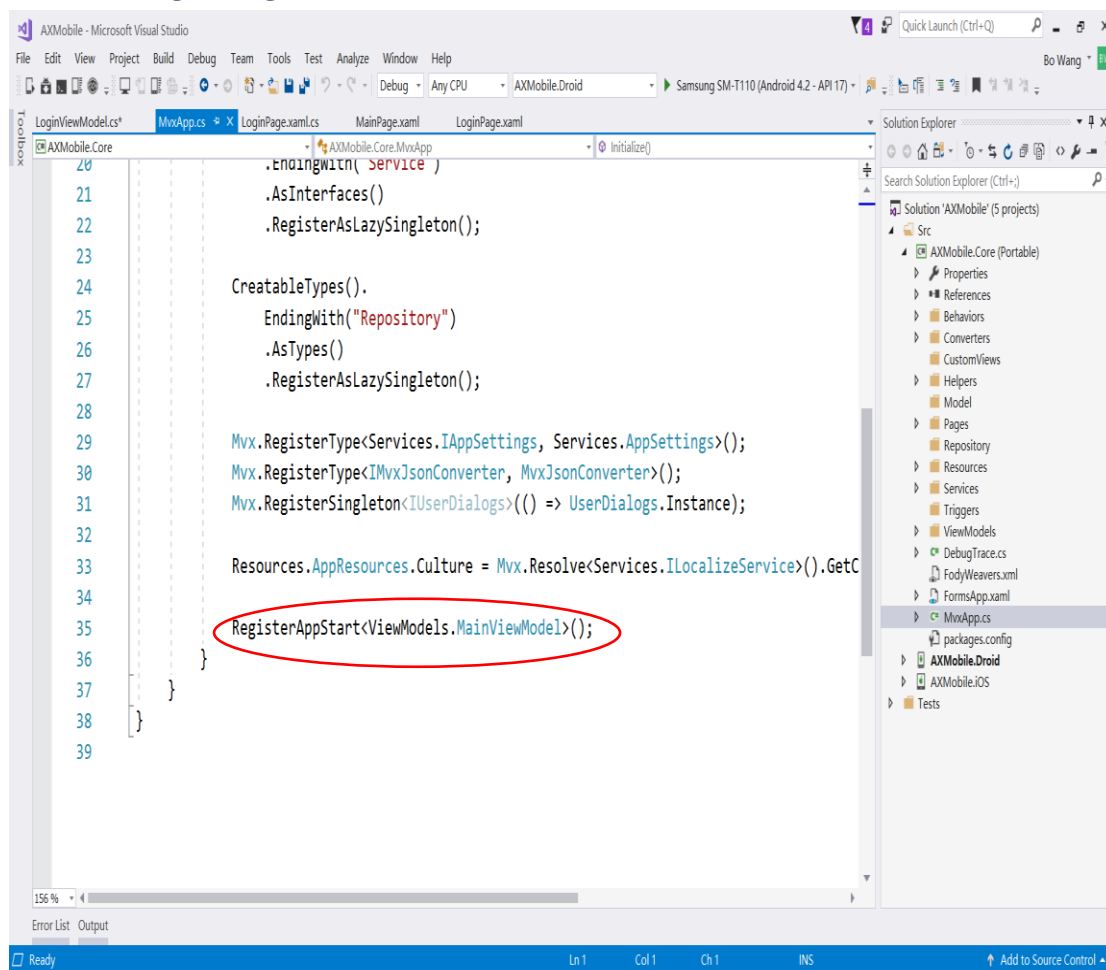
```
public class LoginViewModel:MvxViewModel
{
    private readonly IMvxNavigationService _navigationService;
    public LoginViewModel()
    {
        _navigationService = Mvx.Resolve<IMvxNavigationService>();
    }

    public IMvxAsyncCommand GoToMainPage => new MvxAsyncCommand(async () =>
    {
        await _navigationService.Navigate<MainViewModel>();
    });

    public IMvxCommand OpenGithubUrlCommand => new MvxCommand(() =>
    {
        Device.OpenUri(new Uri("http://www.highviewsoft.com/"));
    });
}
```

注：在 LoginPage.Xaml 对应到 LoginViewModel 是 MvvmCross 框架的约定，在这里不对 MvvmCross 框架进行叙述。

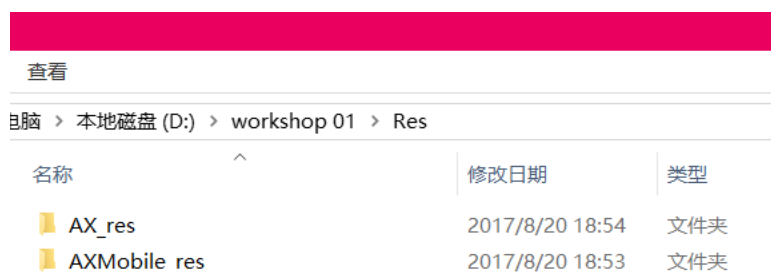
d:设置 LoginPage 为启动页面:



在 AXMobile.Core 项目中找到 MvxApp 打开并且找到图中红圈中所示的代码，将 `RegisterAppStart<ViewModels.MainViewModel>();` 改为 `RegisterAppStart<ViewModels.LoginViewModel>();`;

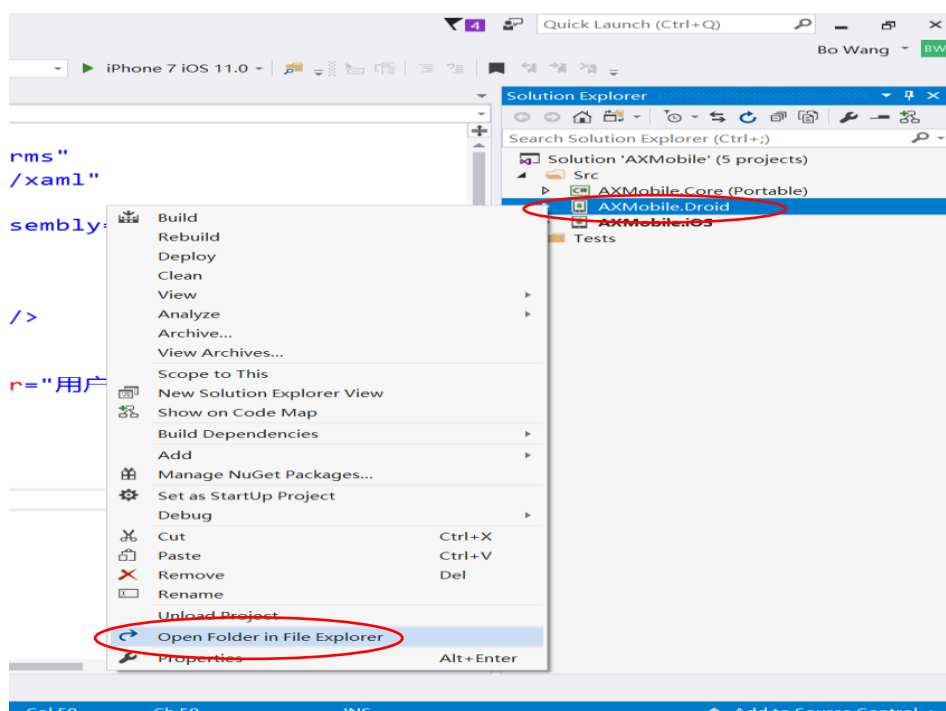
e:为页面添加图片素材:

此项目所需要的素材需要在 PDF 文件下, 之前下载 AXMobile 项目那下载 Res.zip 文件, 下载下来并解压得到图 1 所示两个文件夹, 之后打开 AXMobile_res 文件夹, 以 Android 为例, 把 Android 中的所有文件夹复制并覆盖到 AXMobile_01Start\AXMobile\AXMobile.Droid\Resources 位置下(也可以从图 2 所示打开 Droid 目录)。

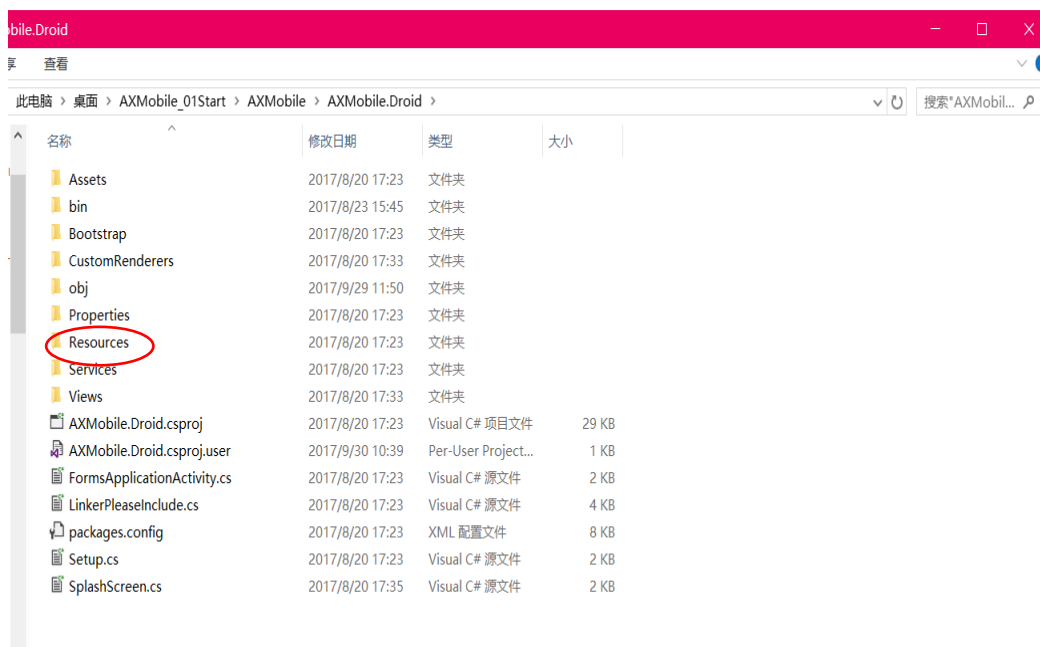


(图 1)

也可以从 Solution 中打开, 方式见图 2

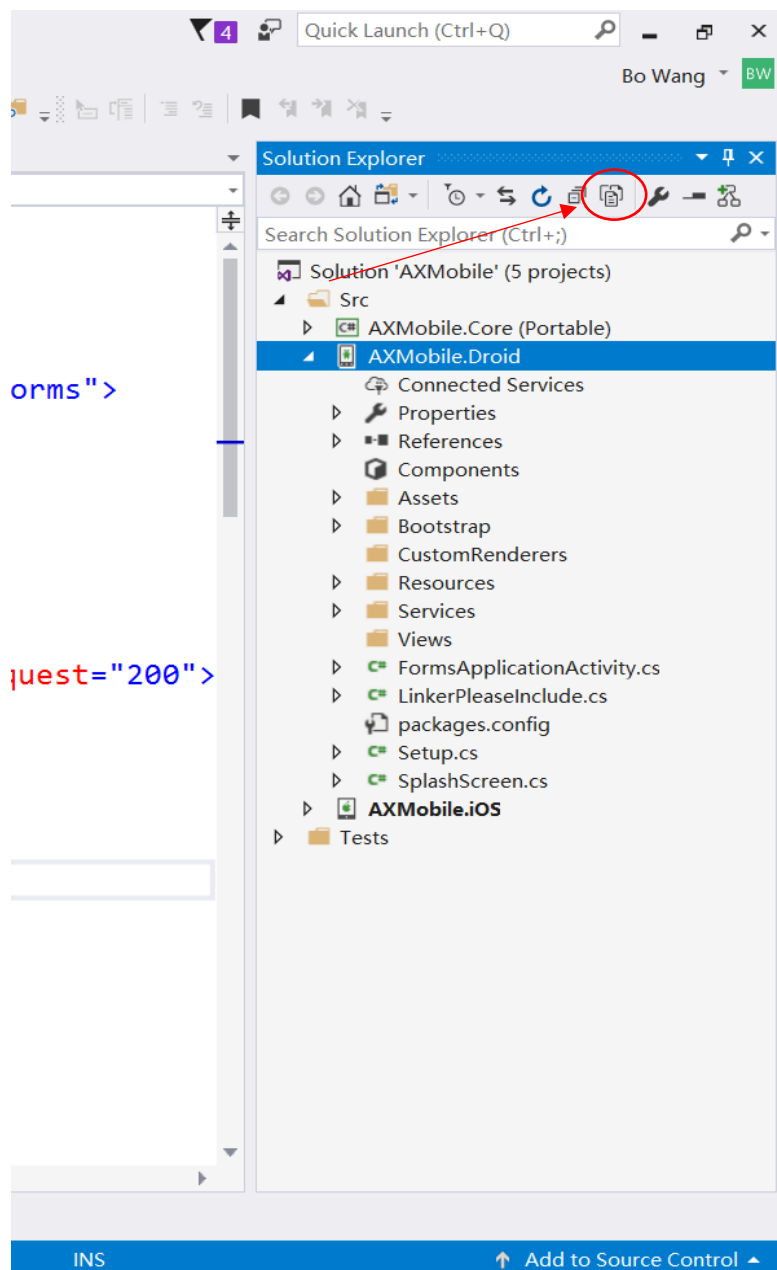


(图 2)

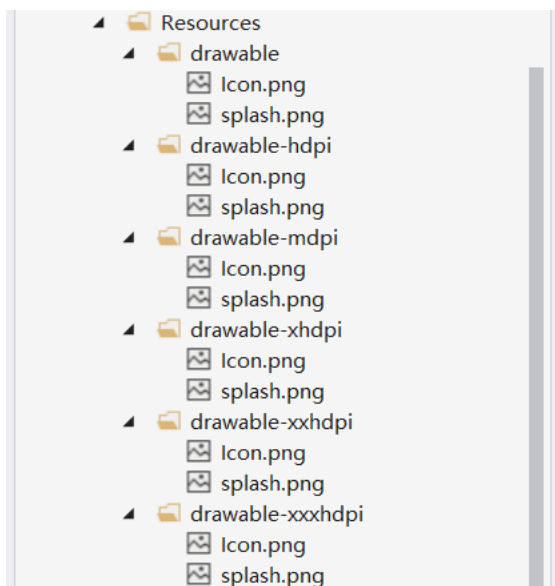


出现这个界面后，打开 Resources，将从 Res.zip 中解压得到的 Android 素材替换里边的所有文件夹，iOS 同理，这样就完成对 AXMobile 的素材添加。

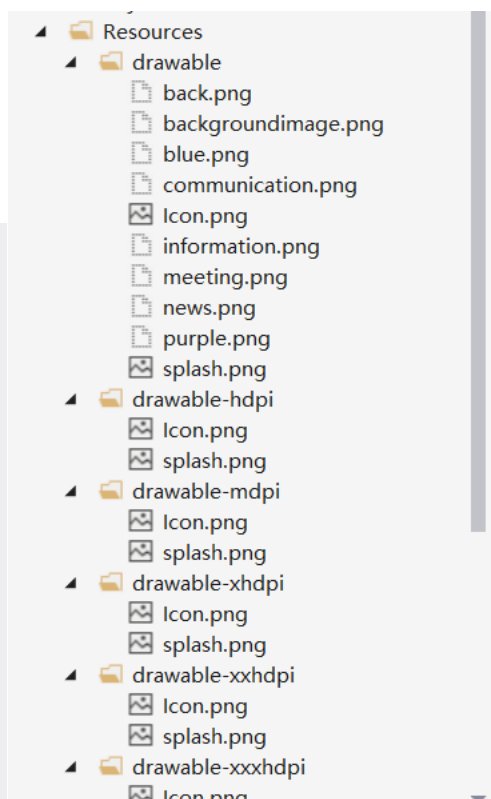
覆盖完成后，打开 AXMobile.Droid 项目中的 Resources 文件夹，之后，发现刚才覆盖的文件没有被包括进来(图 2)，这时点击图 1 中的小红圈，之后可以看见有一些虚线边框的文件(图 3)，选中这些文件点击鼠标右键，再点击 Include in Project（包括在项目中），这样图片素材就被添加进 AXMobile 项目中了(iOS 同理)。



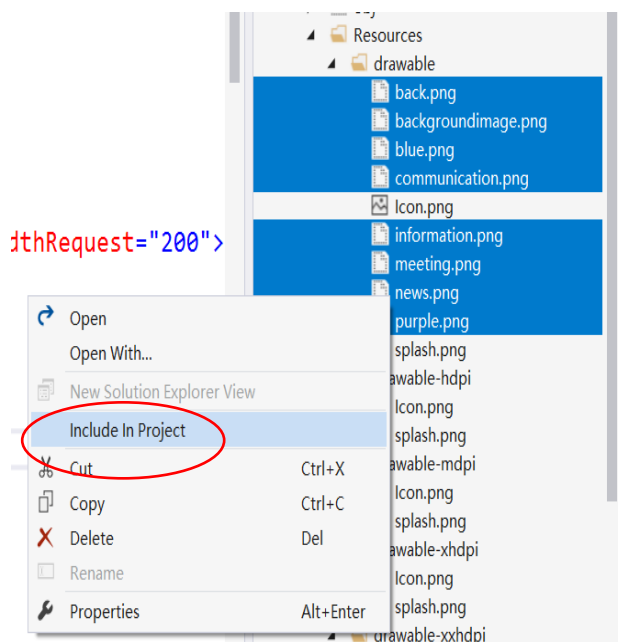
(图 1)



(图 2)



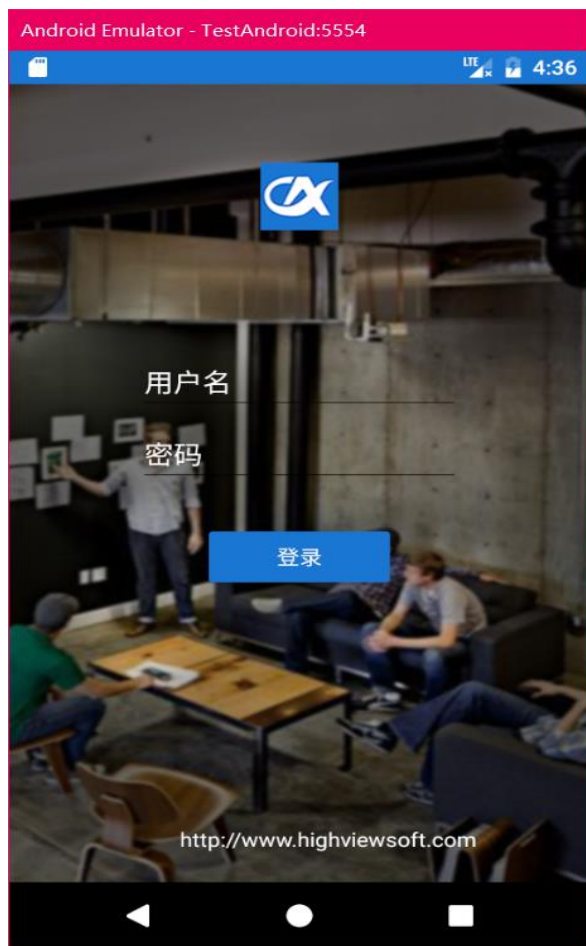
(图 3)



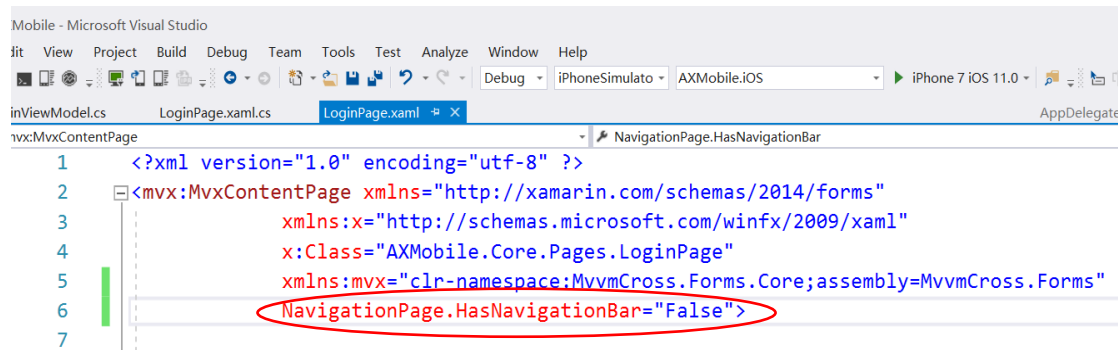
(图 4)

f:隐藏登录界面导航栏:

这时使用 Android 模拟器或真机启动后, 由于框架原因(不是 Xamarin 所致), UI 界面出现了一个导航栏, 现在得把他去掉从而使界面变得漂亮, 返回 LoginPage.Xaml 界面, 在图 2 中添加红圈中的代码 (`NavigationPage.HasNavigationBar= "False"`), 这样登录界面上的蓝色导航栏区域就被清理掉了, Android 可直接用真机调试显示结果, 图 1 为 Android 7.0 运行效果图。



(图 1)



(图 2)

04:创建 MainPage.Xaml 页面

实现从 LoginPage 跳转到 MainPage:

在 Solution 生成后会自动生成一个 MainPage 项目, 直接使用 LoginPage.Xaml 页面导航 Navigation 至 MainPage.Xaml 页面即可。

之后在 MainPage.xaml 页面下修改成以下代码:

```

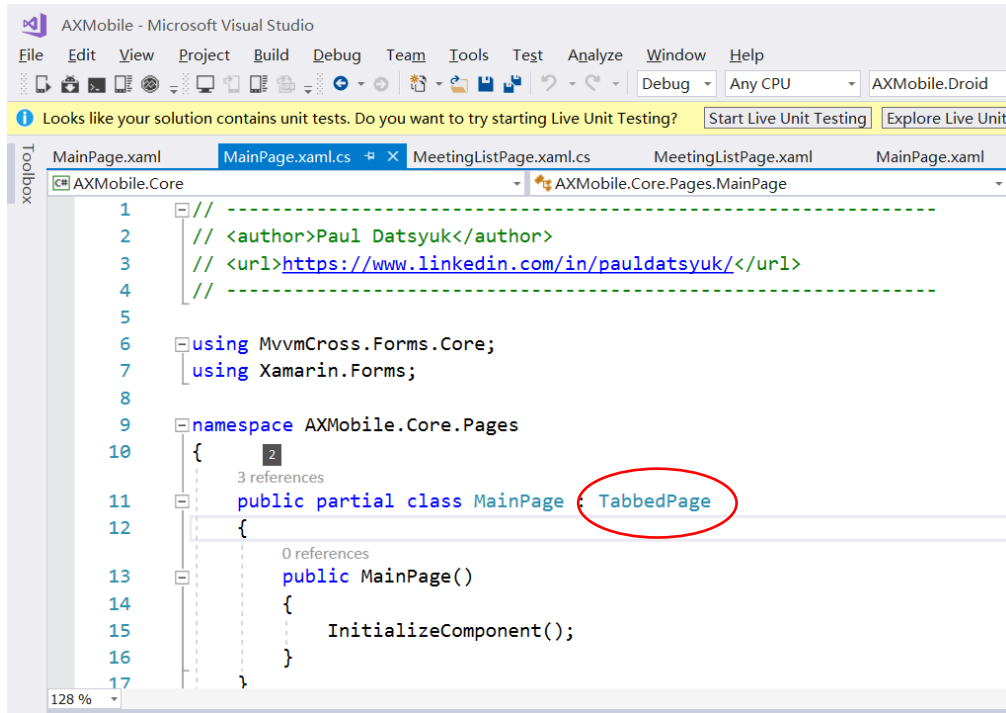
<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AXMobile.Core.Pages.MainPage"
    xmlns:meeting="clr-namespace:AXMobile.Core.Pages;assembly=AXMobile.Core"
    NavigationPage.HasBackButton="False"
    Title="AX">

    <TabbedPage.ToolbarItems>
        <ToolbarItem Icon="back.png" Command="{Binding SignOutCommand}"/>
    </TabbedPage.ToolbarItems>

    <TabbedPage.Children>
        <meeting:MeetingListPage Title="会议" Icon="meeting.png"/>
        <ContentPage Title="新闻" Icon="news.png"/>
        <ContentPage Title="聊天" Icon="communication.png"/>
        <ContentPage Title="我的" Icon="information.png"/>
    </TabbedPage.Children>
</TabbedPage>

```

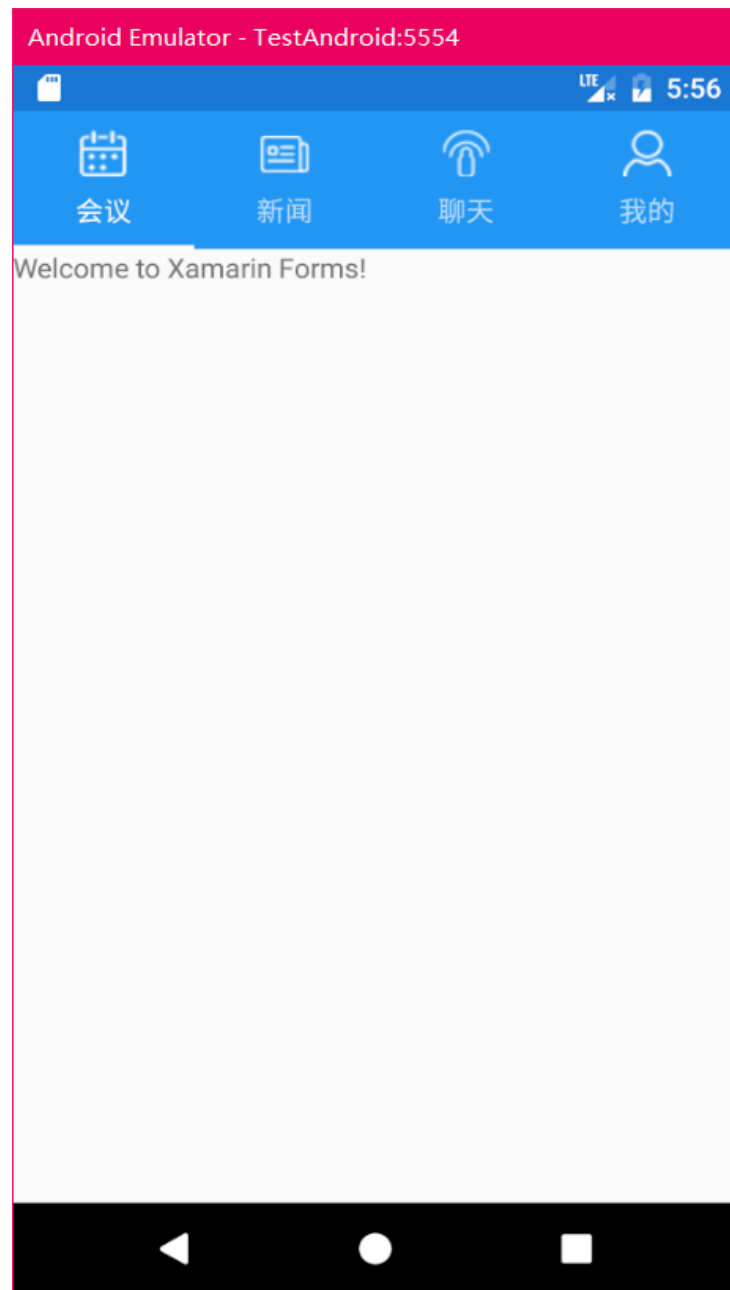
添加完代码后，打开 MainPage.xaml.cs 并按下图所示修改代码。



添加完 MainPage 的 UI 后在 LoginViewModel 下添加如下代码即可从 LoginPage 跳转到 MainPage。

```
public class LoginViewModel:MvxViewModel
{
    private readonly IMvxNavigationService _navigationService;
    public LoginViewModel()
    {
        _navigationService = Mvx.Resolve<IMvxNavigationService>();
    }
    public IMvxAsyncCommand GoToMainPage => new MvxAsyncCommand(async () =>
    {
        await _navigationService.Navigate<MainViewModel>();
    });
    public IMvxCommand OpenGithubUrlCommand => new MvxCommand(() =>
    {
        Device.OpenUri(new Uri("http://www.highviewsoft.com/"));
    });
}
```

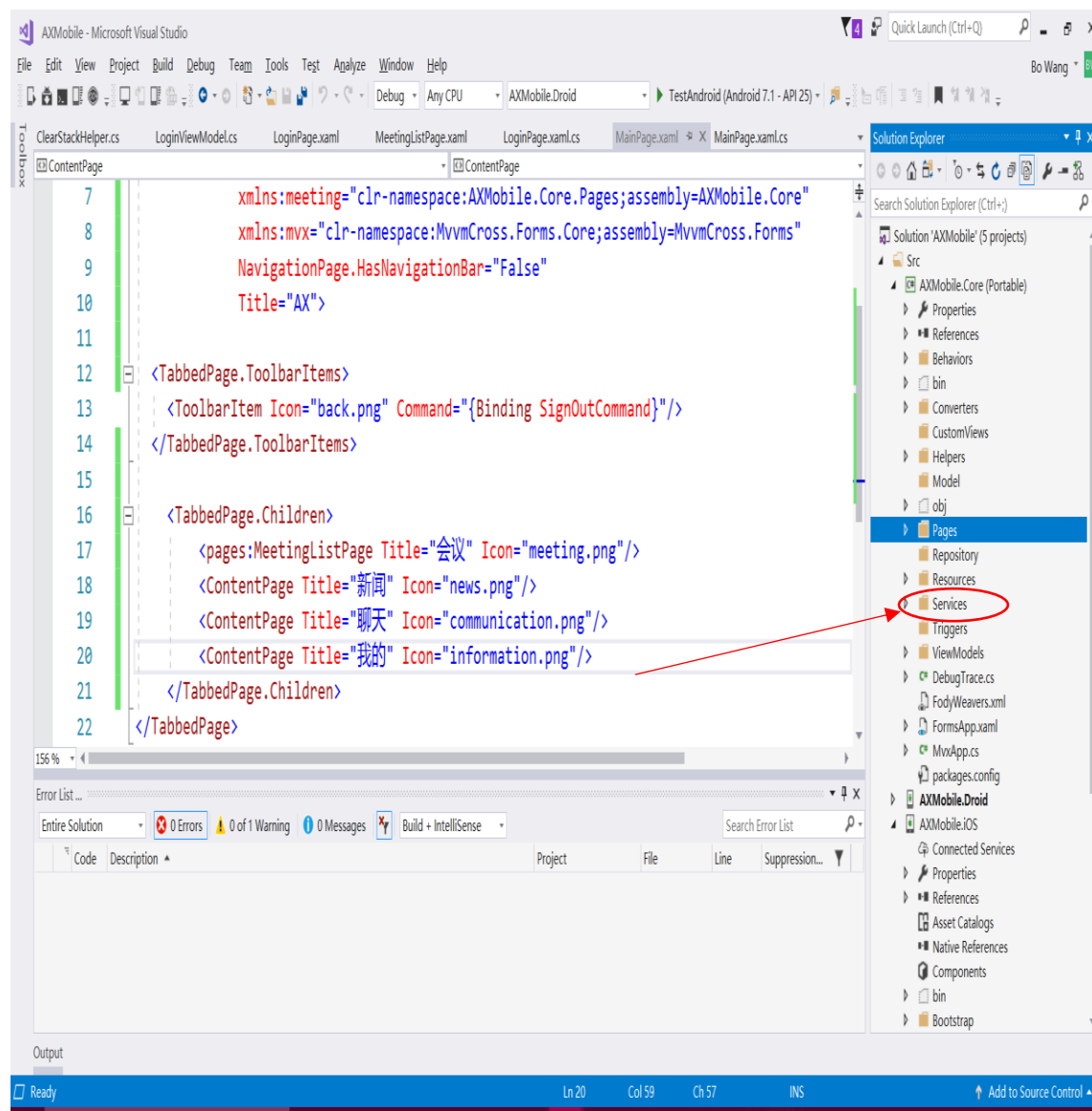
添加后运行效果图：



05:添加会议列表信息:

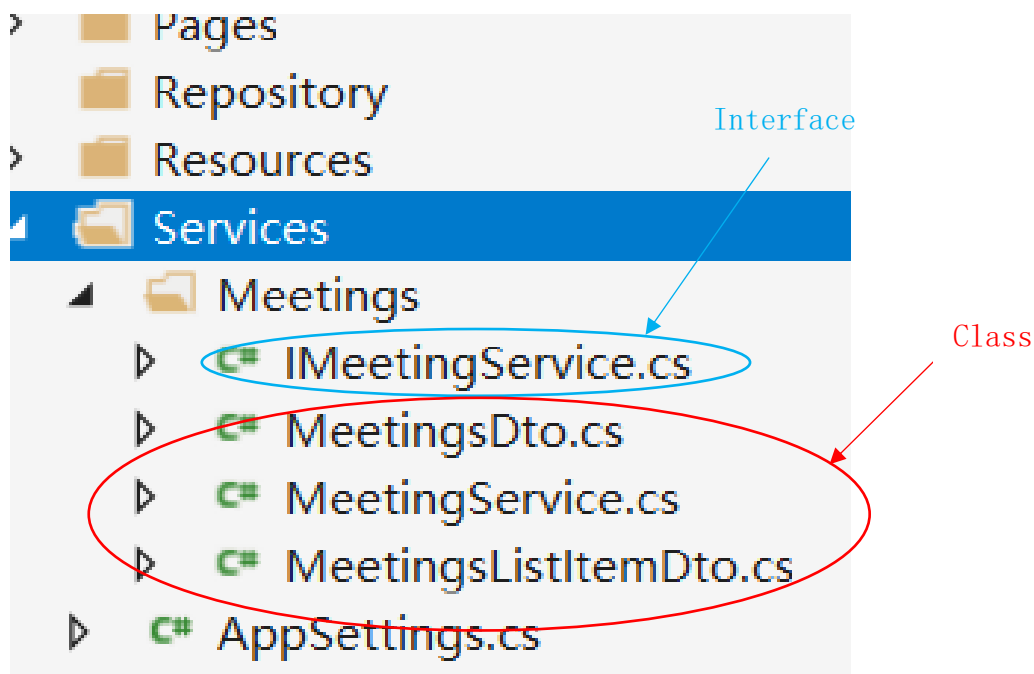
注意: 这里开始是文档操作的难点, 以下若出现报错, 请自己先根据文档的步骤完成, 因为这些代码是互相关联的, 少添加了肯定会报错, 所以请按文档步骤来, 如果弄完了, 还是报错请找现场老师解决。

a:添加获取 Web API 数据功能:



(图 1)

如图 1 所示，右键点击 Services 选择 New Folder 选项，并给文件夹取名为 Meetings，结果为图 2 所示，之后在 Meetings 下创建 1 个 Interface 和 3 个 Class(创建方法参考之前创建的 ViewModel)，并且取名分别如图 2 所示。



(图 2)

在 IMeetingService 中添加以下代码：

```
public interface IMeetingService
{
    Task<Tuple<MeetingsListItemDto, BaseErrorDto>> GetMeetingsListAsync();
    Task<Tuple<MeetingsDto, BaseErrorDto>> GetMeetingsAsync(in
```

在 MeetingsDto 下添加以下代码:

```
public class MeetingsDto
{
    public int Id { get; set; }
    public string Subject { get; set; }
    public string SubjectEnglish { get; set; }
    public string Location { get; set; }
    public string LocationEnglish { get; set; }
    public string Content { get; set; }
    public string ContentEnglish { get; set; }
    public DateTime StartTime { get; set; }
    public DateTime? EndTime { get; set; }
    public string PictureUrl { get; set; }
}
```

在 MeetingService 中添加以下代码:

```
using AXMobile.Core.Helpers;
using AXMobile.Core.Model;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;

namespace AXMobile.Core.Services.Meetings
{
    public class MeetingService : IMeetingService
    {

```



```

public async Task<Tuple<MeetingsDto, BaseErrorDto>> GetMeetingsAsync(int id)
{
    string url = $"{AXMobile.Core.AppConstants.Base_API_Url}/api/Meeting/get/{id}";

    try
    {
        HttpResponseMessage response = await HttpClientHelper.GetAsync(url);

        //登录成功

        switch (response.StatusCode)
        {
            //200

            case System.Net.HttpStatusCode.OK:
                var content_success = await response.Content.ReadAsStringAsync();
                var items =
                JsonConvert.DeserializeObject<MeetingsDto>(content_success);

                return new Tuple<MeetingsDto, BaseErrorDto>(items, new BaseErrorDto
                { IsSucceed = true });

            //500

            case System.Net.HttpStatusCode.InternalServerError:
                var content = await response.Content.ReadAsStringAsync();
                var commonError =
                JsonConvert.DeserializeObject<CommonErrorDto>(content);

                return new Tuple<MeetingsDto, BaseErrorDto>(null, new BaseErrorDto
                { IsSucceed = false, ErrorMessage = commonError.Error.Message });

            //404

            case System.Net.HttpStatusCode.NotFound:
                return new Tuple<MeetingsDto, BaseErrorDto>(null, new BaseErrorDto
                { IsSucceed = false, ErrorKey = "NotFund" });

            default:
                return new Tuple<MeetingsDto, BaseErrorDto>(null, new BaseErrorDto
                { IsSucceed = false, ErrorKey = "UnexpectedError" });
        }
    }
}

```

```

        catch (Exception e)
        {
            var s = e.Message;

            return new Tuple<MeetingsDto, BaseErrorDto>(null, new BaseErrorDto
{ IsSucceed = false, ErrorKey = "ServerConnectError" });
        }
    }

    public async Task<Tuple<MeetingsListItemDto, BaseErrorDto>> GetMeetingsListAsync()
    {
        string url = $"{AXMobile.Core.AppConstants.Base_API_Url}/meeting/list";
        try
        {
            HttpResponseMessage response = await HttpClientHelper.GetAsync(url);

            //登录成功

            switch (response.StatusCode)
            {
                //200

                case System.Net.HttpStatusCode.OK:
                    var content_success = await response.Content.ReadAsStringAsync();

                    var items =
JsonConvert.DeserializeObject<MeetingsListItemDto>(content_success);

                    return new Tuple<MeetingsListItemDto, BaseErrorDto>(items, new
BaseErrorDto { IsSucceed = true });

                    //500

                case System.Net.HttpStatusCode.InternalServerError:
                    var content = await response.Content.ReadAsStringAsync();

                    var commonError =
JsonConvert.DeserializeObject<CommonErrorDto>(content);

                    return new Tuple<MeetingsListItemDto, BaseErrorDto>(null, new
BaseErrorDto { IsSucceed = false, ErrorMessage = commonError.Error.Message });

                default:
                    return new Tuple<MeetingsListItemDto, BaseErrorDto>(null, new
BaseErrorDto { IsSucceed = false, ErrorKey = "UnexpectedError" });
            }
        }
    }

```

```

        }
    }
    catch (Exception e)
    {
        var s = e.Message;

        return new Tuple<MeetingsListItemDto, BaseErrorDto>(null, new BaseErrorDto
{ IsSucceed = false, ErrorKey = "ServerConnectError" });
    }
}
}
}
}

```

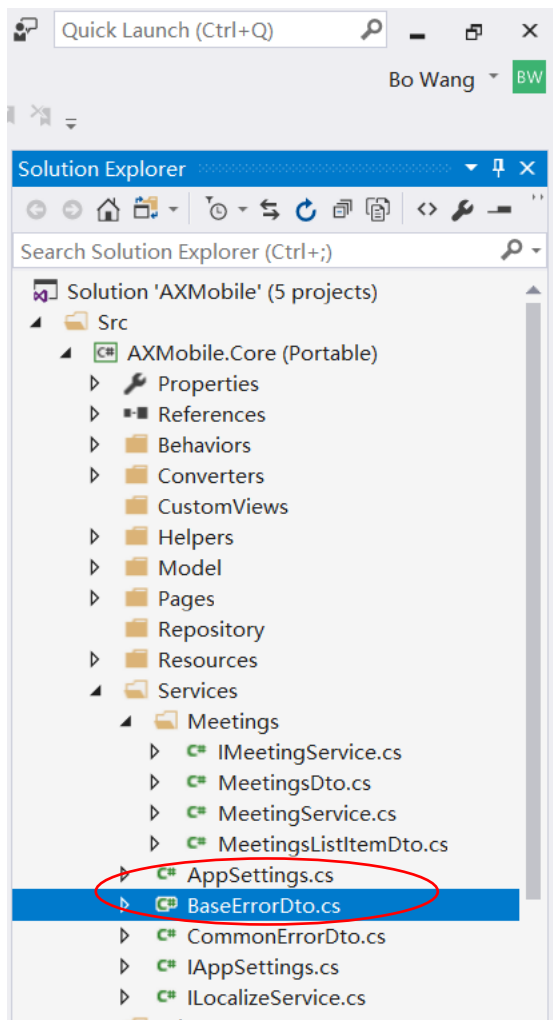
在 MeetingsListItemDto 中添加以下代码：

```

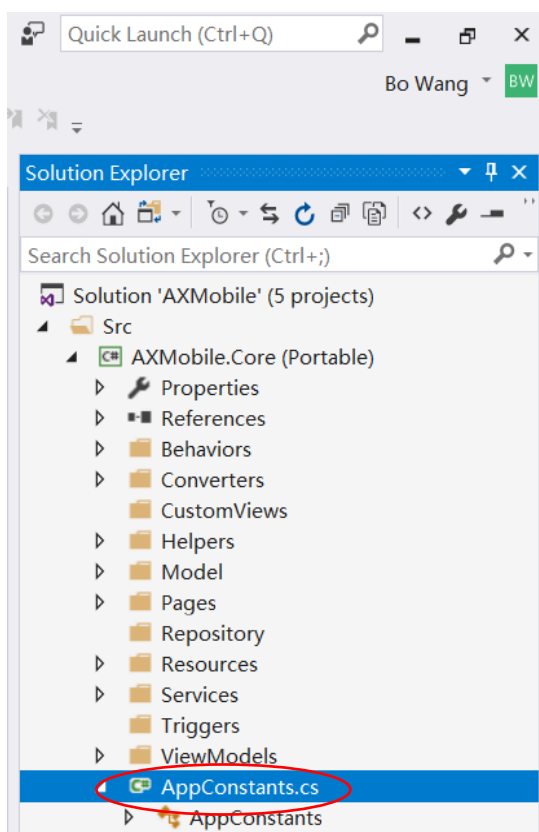
using System;
using System.Collections.Generic;
namespace AXMobile.Core.Services.Meetings
{
    public class MeetingsListItemDto
    {
        public int TotalCount { get; set; }
        public List<Items> Items { get; set; }
    }
    public class Items
    {
        public string Subject { get; set; }
        public string Location { get; set; }
        public DateTime StartTime { get; set; }
        public DateTime EndTime { get; set; }
        public int Id { get; set; }
    }
}

```

添加完成后回到 Services 目录下，在 Services 目录下右键创建两个 Class，如图 1 所示，再返回到 AXMobile.Core 目录下创建一个名为 AppConstants 的 Class (如图 2)。



(图 1)



(图 2)

在 AppConstants 中添加以下代码:

```
public class AppConstants
{
    public const string Azure_Notification_Hub_Name = "projectax";

    public const string Azure_Notification_Hub_ListenConnectionString = "Endpoint=sb://projectaxhub.servicebus.windows.net/;SharedAccessKeyName=DefaultListenSharedAccessSignature;SharedAccessKey=Cex0Qsn1lzQWUHdxT+WQJ3TtgwXGXQwnv28nudZoD/c=";

    public const string Base_API_Url = "https://projectax.azurewebsites.net/api/";
}
```

在 BaseErrorDto 中添加以下代码:

```
public class BaseErrorDto
{
    public bool IsSucceed { get; set; }
    public string ErrorKey { get; set; }
    public string ErrorMessage { get; set; }
}
```

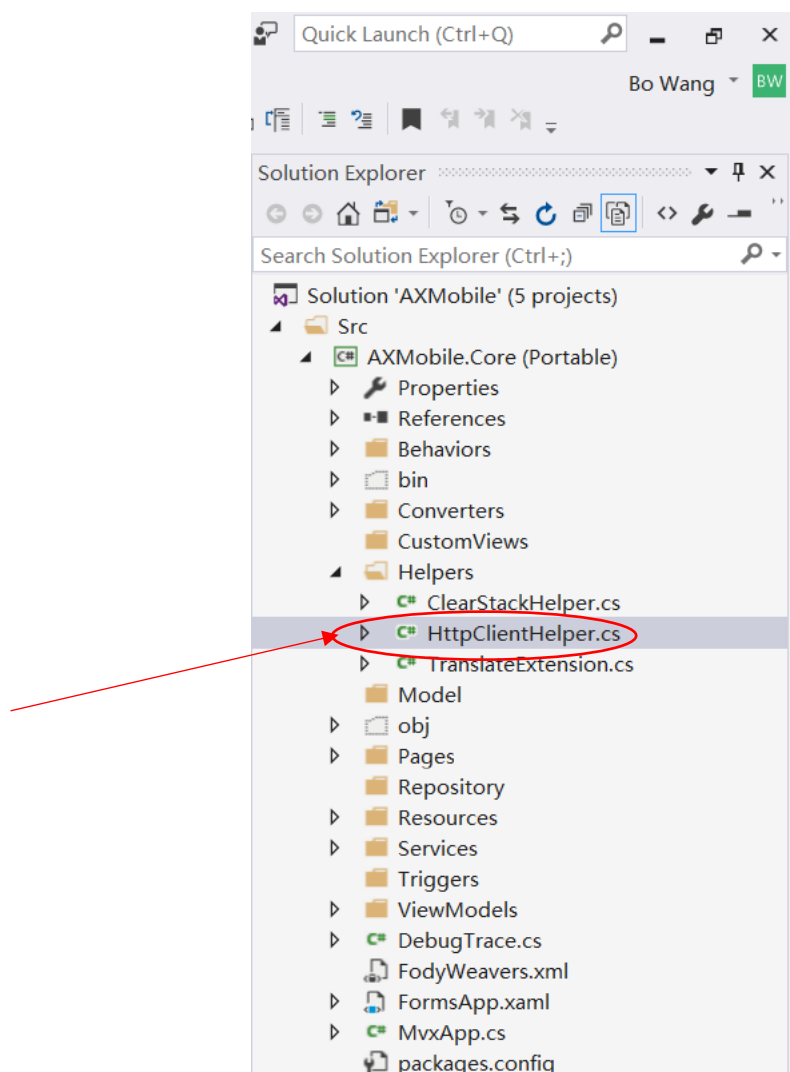
在 CommonErrorDto 中添加以下代码:

```
public class CommonErrorDto
{
    public string Result { get; set; }
    public string TargetUrl { get; set; }
    public bool Success { get; set; }
    public Error Error { get; set; }
    public bool UnauthorizedRequest { get; set; }
    public bool __abp { get; set; }
}
```

```
public class Error
{
    public int code { get; set; }
    public string Message { get; set; }
    public string Details { get; set; }
    public List<ValidationErrors> ValidationErrors { get; set; }
}

public class ValidationErrors
{
    public string Message { get; set; }
    public List<string> Members { get; set; }}
```

在 AXMobile.Core 项目下的 Helpers 下创建名为 HttpClientHelper 的 Class(如图 1 所示)：



(图 1)

之后在 HttpClientHelper 中添加以下代码：

```
public static class HttpClientHelper
{
    private static HttpClient _client;

    public static HttpClient client => _client ?? (_client = new HttpClient());

    public static async Task<HttpResponseMessage> PostAsync(string url, object model)
```

```

{
    var json = JsonConvert.SerializeObject(model);
    var content = new StringContent(json, Encoding.UTF8, "application/json");
    var response = await GetHttpClient().PostAsync(url, content);
    return response;
}

public static async Task<HttpResponseMessage> GetAsync(string url)
{
    var response = await GetHttpClient().GetAsync(url);
    return response;
}

private static HttpClient GetHttpClient()
{
    client.DefaultRequestHeaders.Clear();
    return client;
}
}

```

b:创建并修改 MeetingListPage.xaml 页面代码并且展示从 API 上获取的数据:

注: 创建过程请参考 LoginPage.xaml

修改 MeetingListPage.xaml 代码如下:

```

<?xml version="1.0" encoding="utf-8" ?>
<mvx:MvxContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:mvx="clr-namespace:MvvmCross.Forms.Core;assembly=MvvmCross.Forms"
    xmlns:local="clr-namespace:AXMobile.Core.Behaviors;assembly=AXMobile.Core"
    BackgroundColor="#d8d8d8"
    x:Class="AXMobile.Core.Pages.MeetingListPage"
    Title="会议列表">

    <Grid >
        <Grid.Margin>

```



```

        <OnPlatform x:TypeArguments="Thickness">
            <OnPlatform.iOS>
                0,65,0,0
            </OnPlatform.iOS>
        </OnPlatform>
    </Grid.Margin>
    <!--<Image Source="background_meetinglist" Aspect="Fill"/>-->
    <StackLayout Orientation="Vertical" HorizontalOptions="FillAndExpand"
VerticalOptions="FillAndExpand" Padding="0">
        <ListView SeparatorVisibility="None" BackgroundColor="Transparent" Margin="0"
RowHeight="96" IsPullToRefreshEnabled="True" RefreshCommand="{Binding
MeetingListViewModel.RefreshCommand}"
            ItemsSource="{Binding MeetingListViewModel.All}" SelectedItem="{Binding
MeetingListViewModel.SelectedMeetingListItem, Mode=TwoWay}" IsRefreshing="{Binding
MeetingListViewModel.IsRefreshing}">
            <ListView.Behaviors>
                <local:EventToCommandBehavior EventName="ItemSelected" Command="{Binding
MeetingListViewModel.ToDetailCommand}"/>
            </ListView.Behaviors>
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout Orientation="Horizontal" Margin="10,11,10,0"
Spacing="0" BackgroundColor="White">
                            <BoxView Color="{Binding GroupHeaderBackgroundColor}"
WidthRequest="8" HeightRequest="85"/>
                            <StackLayout Orientation="Vertical" HeightRequest="85"
Spacing="0">
                                <Label BackgroundColor="White" Text="{Binding
MeetingDateTime}" FontSize="{Binding MeetingDateTimeSize}" TextColor="#616161"
VerticalTextAlignment="Center"
                                    HorizontalTextAlignment="Center" WidthRequest="65" Margin="0,32,0,2"/>
                                <Label BackgroundColor="White" Text="{Binding
MeetingDateMonth}" FontSize="11" TextColor="#616161" VerticalTextAlignment="Center"
                                    HorizontalTextAlignment="Center" WidthRequest="65" IsVisible="{Binding
MeetingDateMonthVisible}" Margin="0"/>
                            </StackLayout>
                            <BoxView Color="#dddede" WidthRequest="2" HeightRequest="85"/>
                            <Image Source="{Binding GroupHeaderImageSource}"
Margin="15,45,0,5" Aspect="AspectFill" VerticalOptions="EndAndExpand"/>
                            <StackLayout Orientation="Vertical" Margin="14,0,0,0"
Spacing="0" HorizontalOptions="FillAndExpand">
                                <Label TextColor="#232323" Text="{Binding NameText}"
FontSize="13" Margin="0,8,0,10"/>
                                <Label TextColor="#4c4b4b" Text="{Binding TimeText}"
FontSize="11" Margin="0,0,0,0" VerticalOptions="EndAndExpand"/>
                                <BoxView Color="eeeeee" HeightRequest="2"
HorizontalOptions="FillAndExpand" Margin="0,0,36,3"/>
                                <Label TextColor="#4c4b4b" Text="{Binding AddressText}"
FontSize="11" Margin="0,0,0,6"/>
                            </StackLayout>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>

```

```

        </StackLayout>
    </Grid>
</mvx:MvxContentPage>

```

新建 MeetingListViewModel 并且向里边添加代码(新建方法参考 LoginViewModel 步骤):

```

public class MeetingListViewModel : MvxViewModel
{
    public List<MeetingListItem> All { get; private set; }

    private readonly IMeetingService _meetingService;
    private readonly IMvxNavigationService _navigationService;
    private readonly Services.IAppSettings _settings;
    private readonly IUserDialogs _userDialogs;

    public MeetingListViewModel()
    {
        _meetingService = Mvx.Resolve<IMeetingService>();
        _navigationService = Mvx.Resolve<IMvxNavigationService>();
        _settings = Mvx.Resolve<Services.IAppSettings>();
        _userDialogs = Mvx.Resolve<IUserDialogs>();
    }

    public async Task SourceToModel()
    {
        this.IsRefreshing = true;
        var result = await _meetingService.GetMeetingsListAsync();
        UserDialogs.Instance.HideLoading();
        if (result.Item2.IsSucceed)
        {
            List<MeetingListItem> meetingListSource = new List<MeetingListItem>();
            DateTime lastMeetingDate = DateTime.Today.Date;
            foreach (Items meetingItem in result.Item1.Items)
            {
                //meeting.MeetingDateTime =
                meetingItem.datePublished.Equals(DateTime.Today) ? "今天" :
                meetingItem.datePublished.ToString();
                if (meetingItem.StartTime.Date.Equals(lastMeetingDate))
                {
                    if (lastMeetingDate.Equals(DateTime.Today.Date))
                    {
                        MeetingListItem meeting = new MeetingListItem();
                        meeting.MeetingDateTime = "今天";
                        meeting.MeetingDateTimeSize = 16;
                        meeting.MeetingDateMonthVisible = false;
                        meeting.GroupHeaderBackgroundColor = "#42a5f5";
                        meeting.GroupHeaderTextColor = "#FFFFFF";
                        //add image source
                        meeting.GroupHeaderImageSource = "blue.png";
                        meeting.MeetingId = meetingItem.Id;
                        meeting.TimeText = $"{meetingItem.StartTime.ToString("HH:mm")}—
                        {meetingItem.EndTime.ToString("HH:mm")}";
                    }
                }
            }
        }
    }
}

```

```

        meeting.NameText = meetingItem.Subject;
        meeting.AddressText = meetingItem.Location;
        meetingListSource.Add(meeting);
    }
}
else
{
    lastMeetingDate = meetingItem.StartTime.Date;

    MeetingListItem meeting = new MeetingListItem();
    meeting.MeetingDateTime = $"{meetingItem.StartTime.ToString("dd")}";
    meeting.MeetingDateTimeSize = 17;
    meeting.MeetingDateMonth =
    $"{meetingItem.StartTime.ToString("MMM")}";
    meeting.MeetingDateMonthVisible = true;
    meeting.GroupHeaderBackgroundColor = "#7e57c2";
    meeting.GroupHeaderTextColor = "#000000";
    //add image source
    meeting.GroupHeaderImageSource = "purple.png";
    meeting.MeetingId = meetingItem.Id;
    meeting.TimeText = $"{meetingItem.StartTime.ToString("HH:mm")}—
    {meetingItem.EndTime.ToString("HH:mm")}";

    meeting.NameText = meetingItem.Subject;
    meeting.AddressText = meetingItem.Location;

    meetingListSource.Add(meeting);
}
}
this.All = meetingListSource;
this.IsRefreshing = false;
}
else
{
    if (!string.IsNullOrEmpty(result.Item2.ErrorKey))
        _userDialogs.Alert(result.Item2.ErrorKey);
    else
        _userDialogs.Alert(result.Item2.ErrorMessage);
}
}

}

public MeetingListItem SelectedMeetingListItem { get; set; }

public bool IsRefreshing { get; set; }

public IMvxAsyncCommand RefreshCommand => new MvxAsyncCommand(async () =>
{
    await SourceToModel();
});
public IMvxCommand ToDetailCommand => new MvxCommand(() =>
{
    if (this.SelectedMeetingListItem == null)
        return;
    //_navigationService.Navigate<MettingDetailViewModel,
    MeetingListItem>(this.SelectedMeetingListItem);
}

```

```

        if (this.SelectedMeetingListItem != null)
            SelectedMeetingListItem = null;
    });
}
}

```

接下来双击打开 MainViewModel 并修改成以下代码：

```

public class MainViewModel : MvxViewModel
{
    private readonly IMvxNavigationService _navigationService;
    private readonly Services.IAppSettings _settings;

    public MainViewModel(IMvxNavigationService navigationService, Services.IAppSettings
settings)
    {
        _navigationService = navigationService;
        _settings = settings;
        MeetingListViewModel = new MeetingListViewModel();
        ButtonText = Resources.AppResources.MainPageButton;
    }

    public MeetingListViewModel MeetingListViewModel { get; set; }

    public override async void Start()
    {
        await MeetingListViewModel.SourceToModel();
    }

    public IMvxCommand PressMeCommand =>
        new MvxCommand(() =>
        {
            ButtonText = Resources.AppResources.MainPageButtonPressed;
        });

    public IMvxAsyncCommand GoToSecondPageCommand =>
        new MvxAsyncCommand(async () =>
        {
            var param = new Dictionary<string, string> { { "ButtonText", ButtonText } };

```

```

        await _navigationService.Navigate<SecondViewModel, Dictionary<string, string>>(param);
    });

    public IMvxCommand OpenGithubUrlCommand =>
        new MvxCommand(() =>
        {
            Device.OpenUri(new Uri("https://github.com/JTOne123/XamFormsMvxTemplate"));
        });

    public string ButtonText { get; set; }

    public int SuperNumber
    {
        get { return _settings.SuperNumber; }
        set { _settings.SuperNumber = value; }
    }
}

```

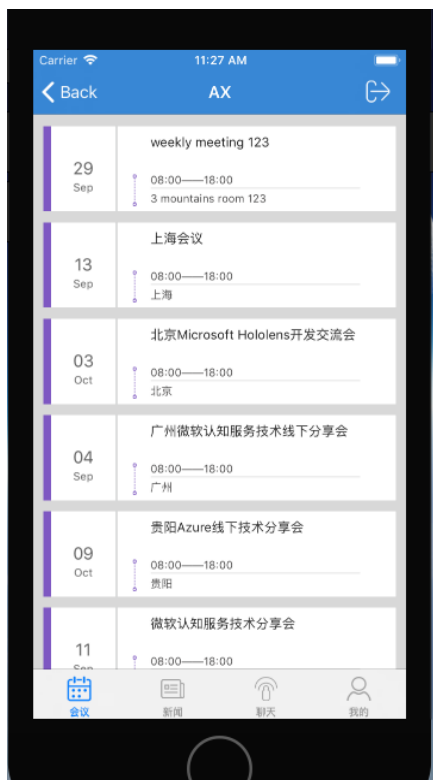
在 AXMobile.Core 文件下找到 Model 文件夹，右击并创建一个 Class(类) 名为 MeetingListItem: 之后在其中添加以下代码:

```

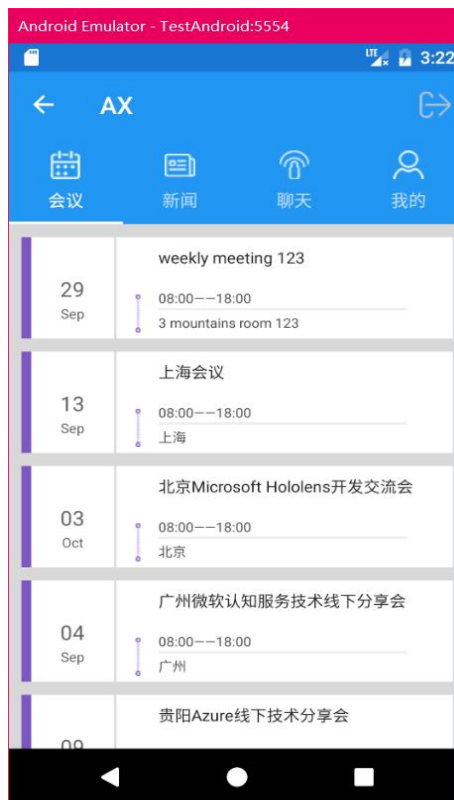
public class MeetingListItem
{
    public string MeetingDateTime { get; set; }
    public int MeetingDateTimeSize { get; set; }
    public string MeetingDateMonth { get; set; }
    public bool MeetingDateMonthVisible { get; set; }
    public string GroupHeaderBackgroundColor { get; set; }
    public string GroupHeaderTextColor { get; set; }
    public string GroupHeaderImageSource { get; set; }
    public int MeetingId { get; set; }
    public string NameText { get; set; }
    public string TimeText { get; set; }
    public string AddressText { get; set; }
}

```

06: 完成所有步骤后，选择对应的设备运行 App 显示结果



(iOS 模拟器运行效果)



(Android 模拟器运行效果)