

Azure & Xamarin Workshop



Azure & Xamarin Workshop

上海 2017 9.12 – 9.13

周岳

jameszhou@highviewsoftware.com

个人介绍

周岳 (jameszhou@highviewsoftware.com)

微软最有价值专家

Xamarin认证开发者

北京视高盛景软件首席架构师(www.highviewsoft.com)

Beijing Xamarin User Group 组织者





微软中国案例中心

[首页](#)[IT趋势](#)[行业类型](#)[产品类型](#)[组织类型](#)[案例搜索](#)[显示更多搜索选项](#)

21 世纪不动产

21 世纪中国不动产利用 Azure 与 Xamarin 以“云+端”方式打造高效的房产 O2O 平台

“通过与微软合作，让我们体验到使用 Xamarin 这种跨平台原生应用程序构建解决方案的优势。我们现有开发人员可以在熟悉的 Visual Studio 环境中利用已有的 C# 经验快速的实现跨平台原生应用的构建、调试、发布，大幅简化新应用的开发周期。同时与 Azure 公有云整合，让我们可以利用更多的云服务为应用提供新的体验，全面实现企业 O2O 平台构建和数字化转型的战略目标。”

——孔明 研发总监 21 世纪中国不动产

业务背景



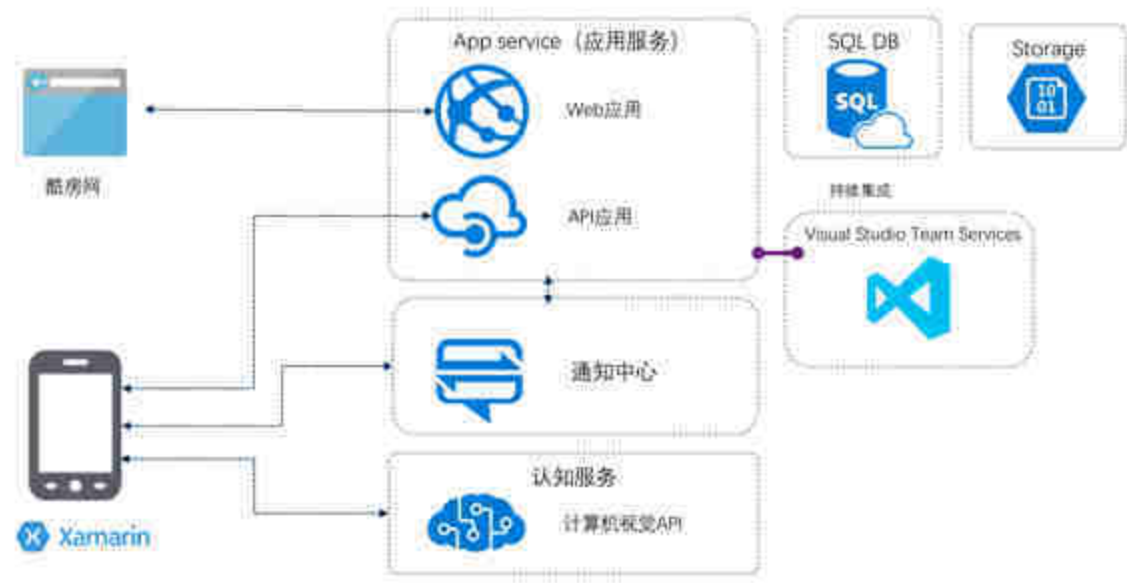
行业：建筑和房地产

地区：中国

运用产品：

- Development Tools

[➔ 了解更多相关案例](#)[➔ 购买产品](#)



解决方案系统架构图



Workshop介绍

2天

09 : 00 – 12 : 00

13 : 00 – 16 : 00



Day 1

- Azure介绍
- Workshop AX项目介绍
- AX项目服务端开发
- AX项目Azure部署

Day 2

- Xamarin 介绍
- AX项目移动端开发
- AX项目调用Azure服务

Azure介绍

全球42个区域



72个产品

Microsoft Azure

为什么选择 Azure? 解决方案 产品 文档 价格 培训 应用商店 合作伙伴 博客 资源

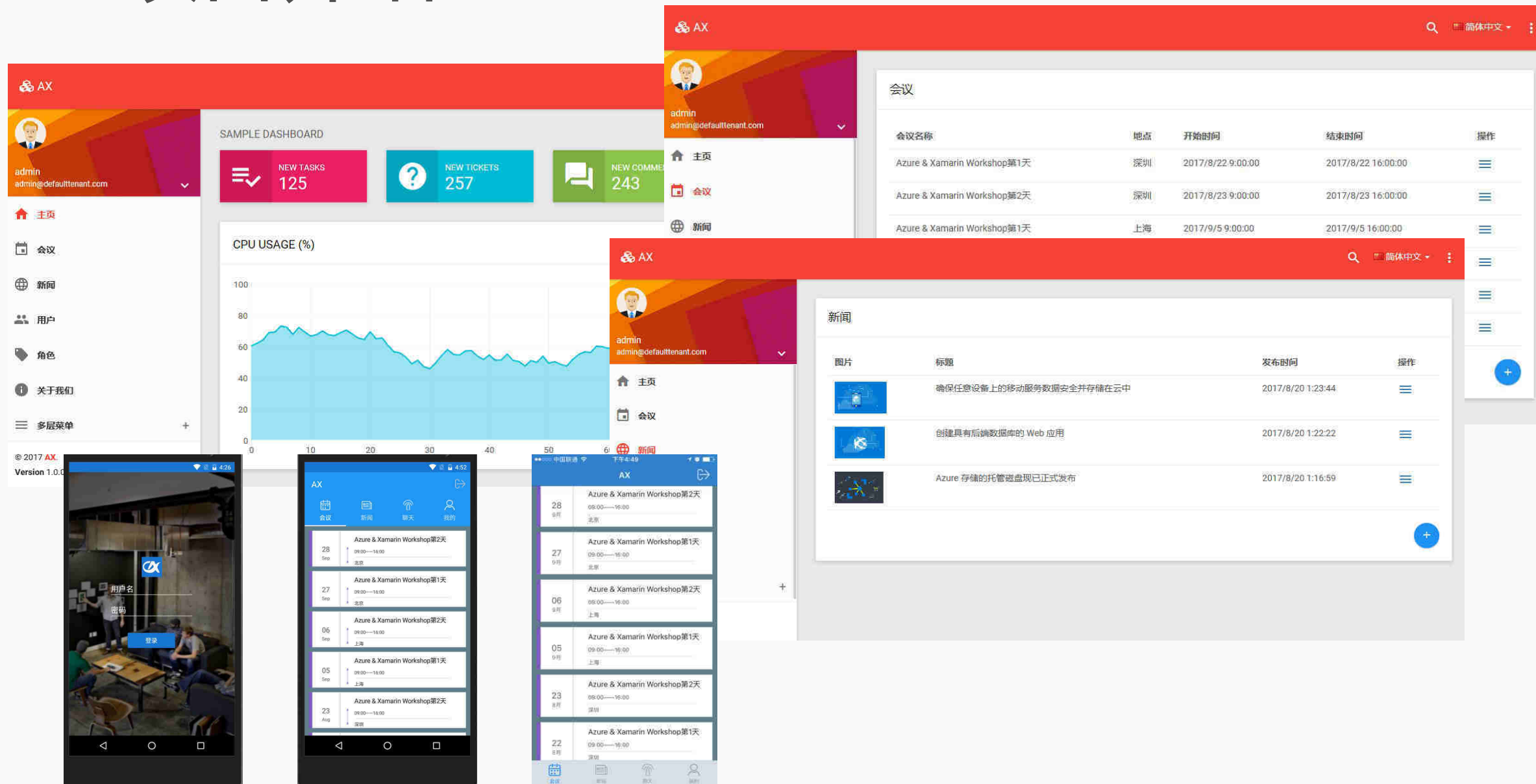
- 计算
 - 虚拟机
 - 在几秒钟内预配好 Windows 和 Linux 虚拟机
 - 虚拟机规模集
 - 管理并扩展到数千台 Linux 和 Windows 虚拟机
 - 应用服务
 - 快速创建适用于 Web 和移动的强大云应用
 - 函数
 - 使用无服务器代码处理事件
 - 批处理
 - 云规模的作业计划和计算管理
 - Service Fabric
 - 在 Windows 或 Linux 上开发微服务和安排容器
 - 云服务
 - 创建高度可用且可无限伸缩的云应用程序和 API
- 联网
- 存储
- Web + 移动
- 容器
- 数据库
- 数据 + 分析
- AI 和认知服务
- 物联网
- 企业集成
- 安全性 + 标识
- 开发人员工具
- 监视 + 管理
- Microsoft Azure Stack

<p>虚拟机</p>  <p>在几秒钟内预配好 Windows 和 Linux 虚拟机</p>	<p>应用服务</p>  <p>快速创建适用于 Web 和移动的强大云应用</p>	<p>SQL 数据库</p>  <p>托管关系 SQL 数据库即服务</p>
<p>存储</p>  <p>持久的、高度可用的、高度可伸缩的云存储器</p>	<p>云服务</p>  <p>创建高度可用且可无限伸缩的云应用程序和 API</p>	<p>Azure Cosmos DB</p>  <p>试着将 Azure Cosmos DB 用于全球分布的多模式数据库</p>

<https://azure.microsoft.com/zh-cn/services/>

Workshop项目AX介绍

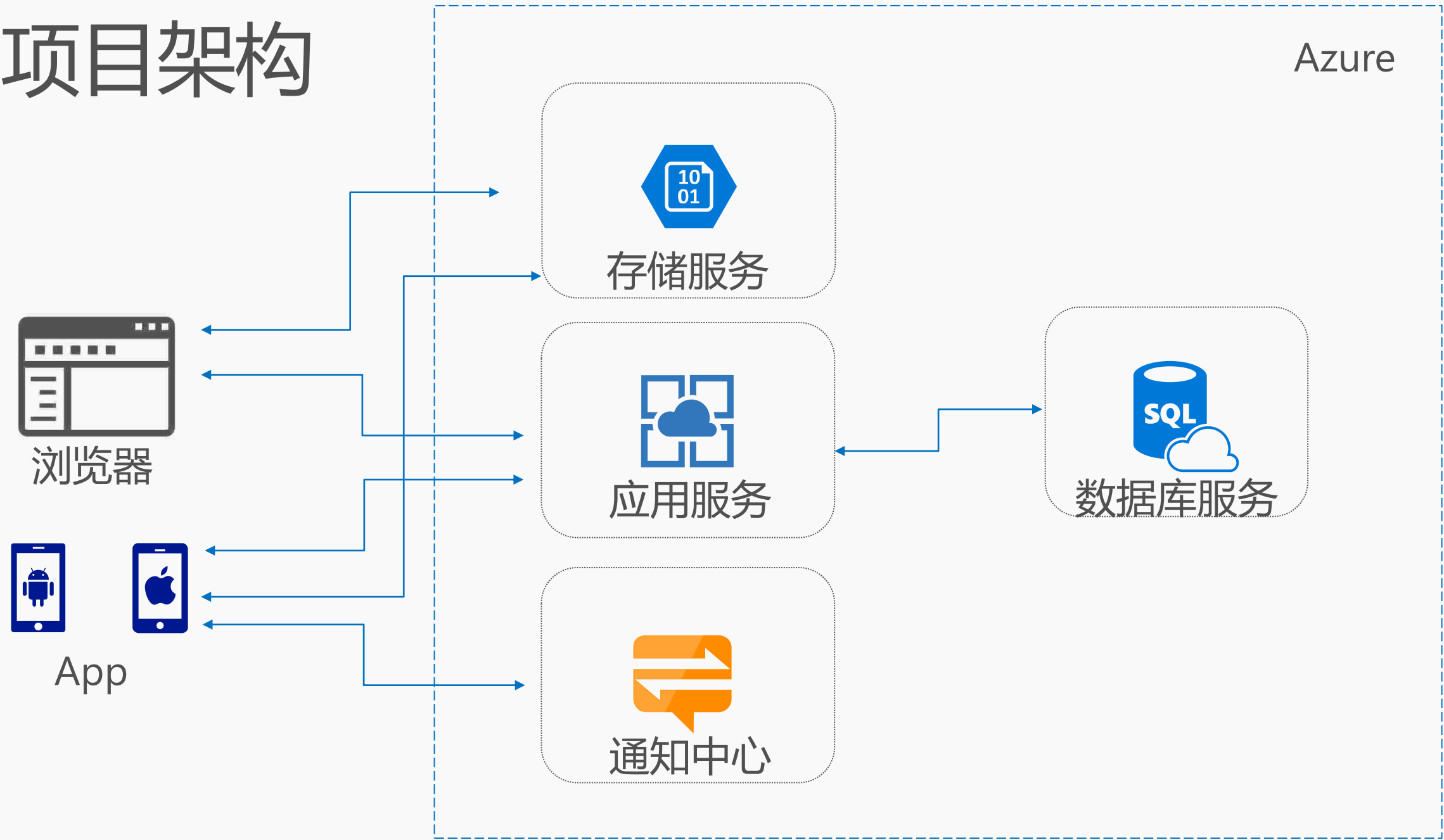
AX 项目介绍



AX项目演示

Photo used under CC <http://www.wocintechchat.com/>

AX项目架构



AX 项目开发工具及语言

- Visual Studio 2017
- ASP.NET MVC
- Xamarin
- C#







AX – 服务端开发

关于Abp框架

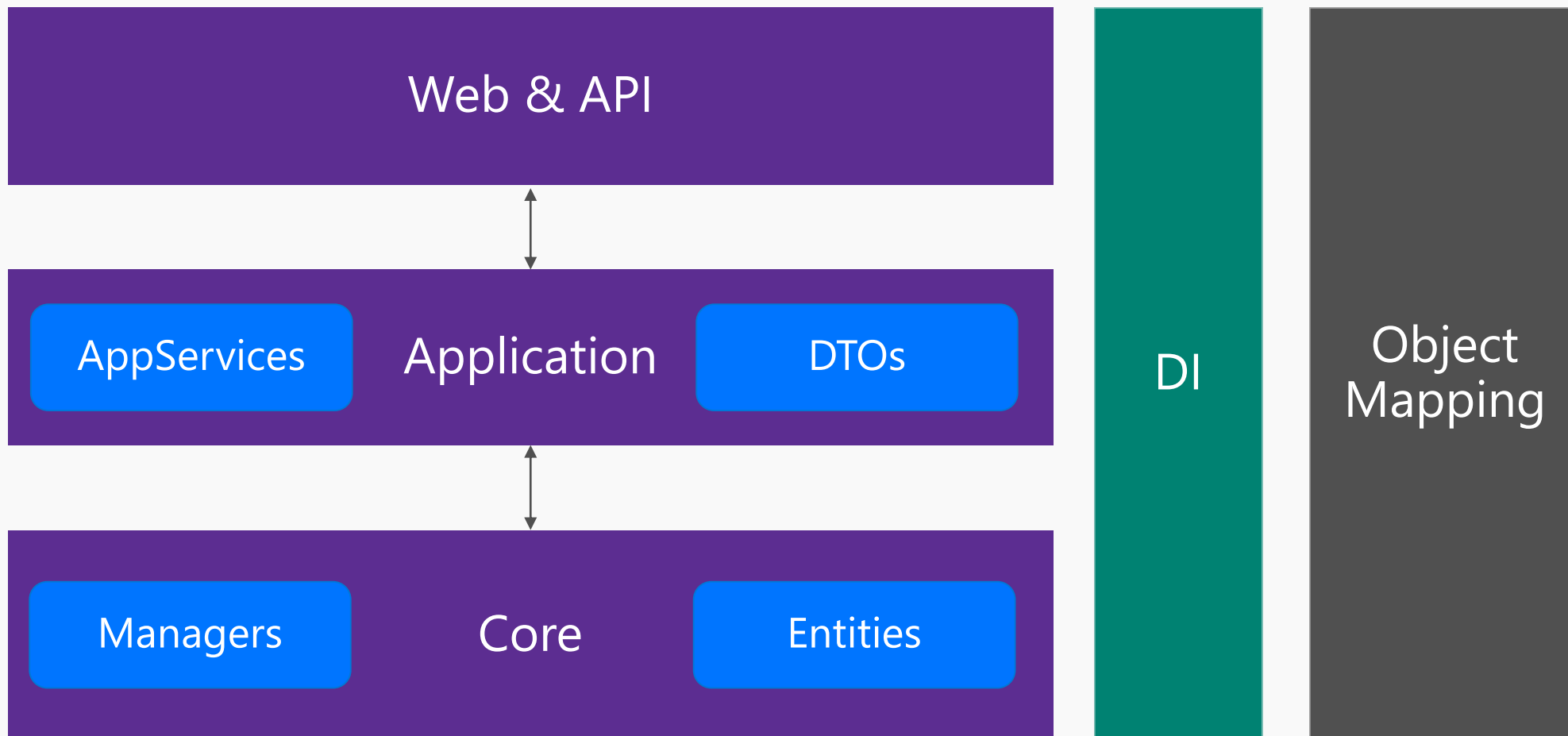
- Abp: <https://aspnetboilerplate.com/>

- 多层结构
- 模块化
- DDD

ASP.NET Boilerplate is a general purpose application framework especially designed for new modern web applications. It uses already familiar tools and implements best practices around them to provide you a SOLID development experience.

	Layered Architecture Provides a layered architectural model based on Domain Driven Design.		Modular Design Designed as modular and extensible. Provides infrastructure to build your own modules too.
	Multi Tenancy SaaS applications made easy! Integrated multi-tenancy from database to UI.		Domain Driven Design Designed and developed DDD in mind. Provides a SOLID model for your application.
	Well Documented Comprehensive documentation and jump start tutorials.		Free & Open Source Actively developed on Github (MIT license) and open for contributions .

Abp框架多层结构



演示

AX功能1：会议



Photo used under CC <http://www.wocintechchat.com/>

演练一：创建AX项目

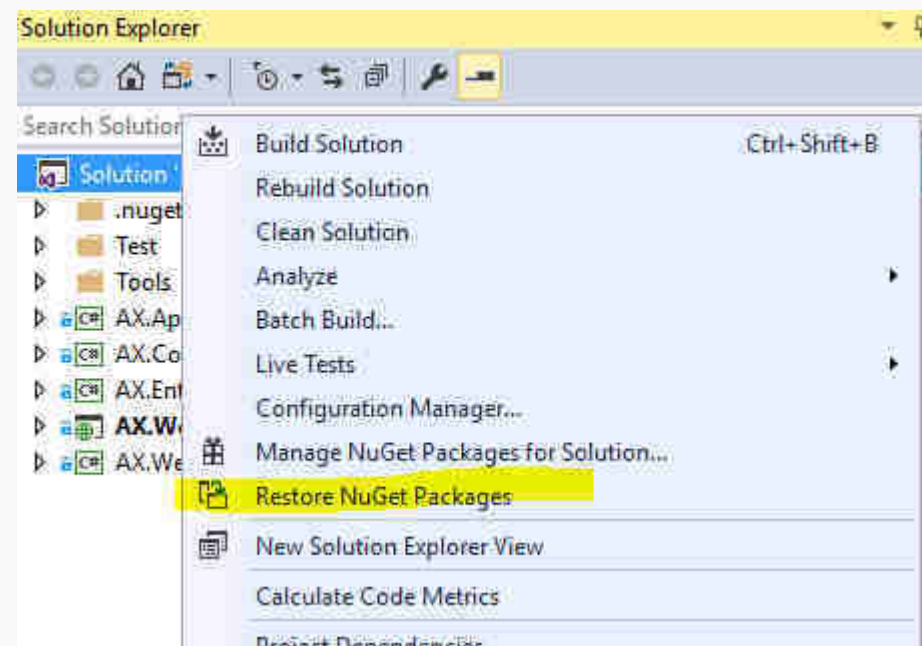
01 : 创建Abp项目模板

- 创建Abp项目模板或使用Workshop提供的模板

https://projectax.blob.core.chinacloudapi.cn/src/AX_01Start.zip

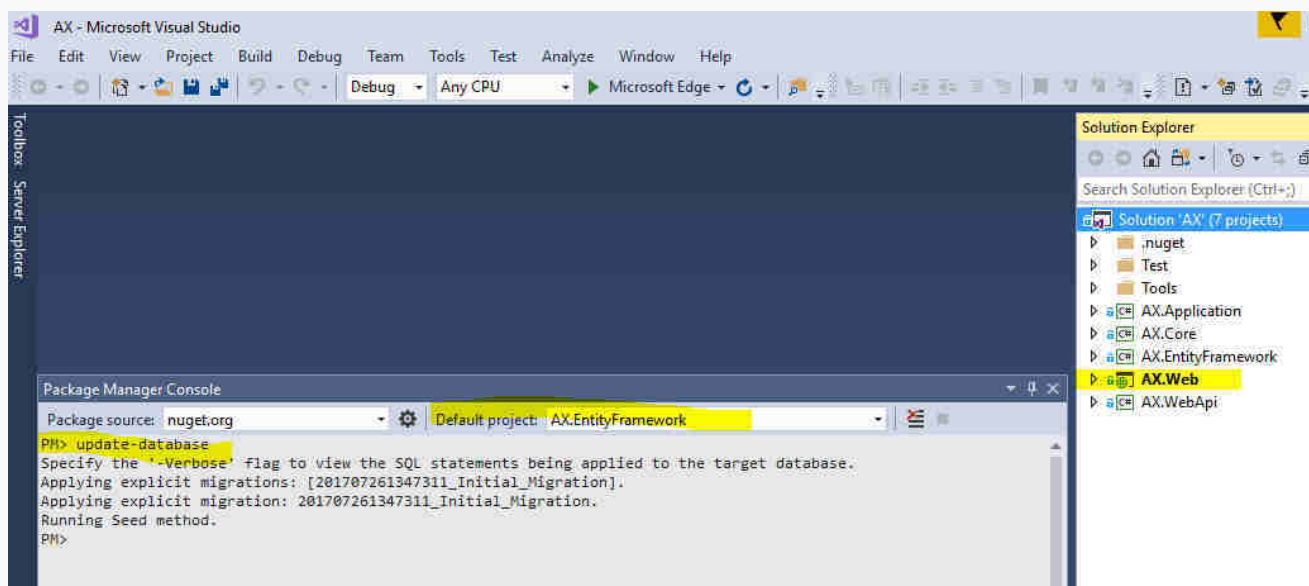
https://projectax.blob.core.chinacloudapi.cn/src/AX_02Full.zip

- 恢复nuget 包，编译解决方案



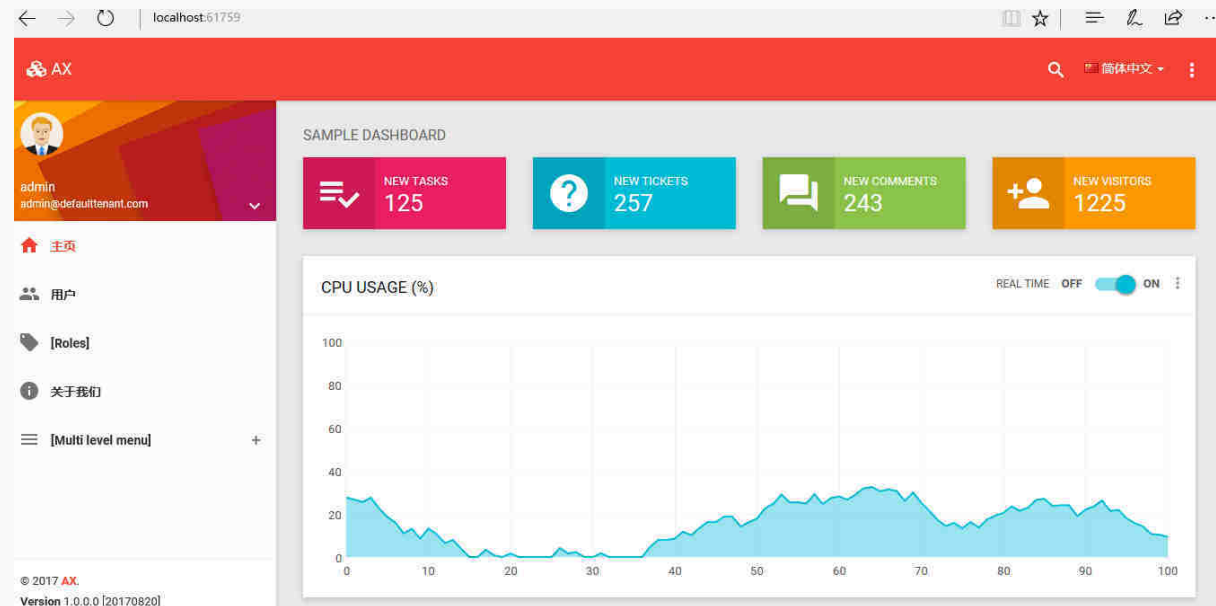
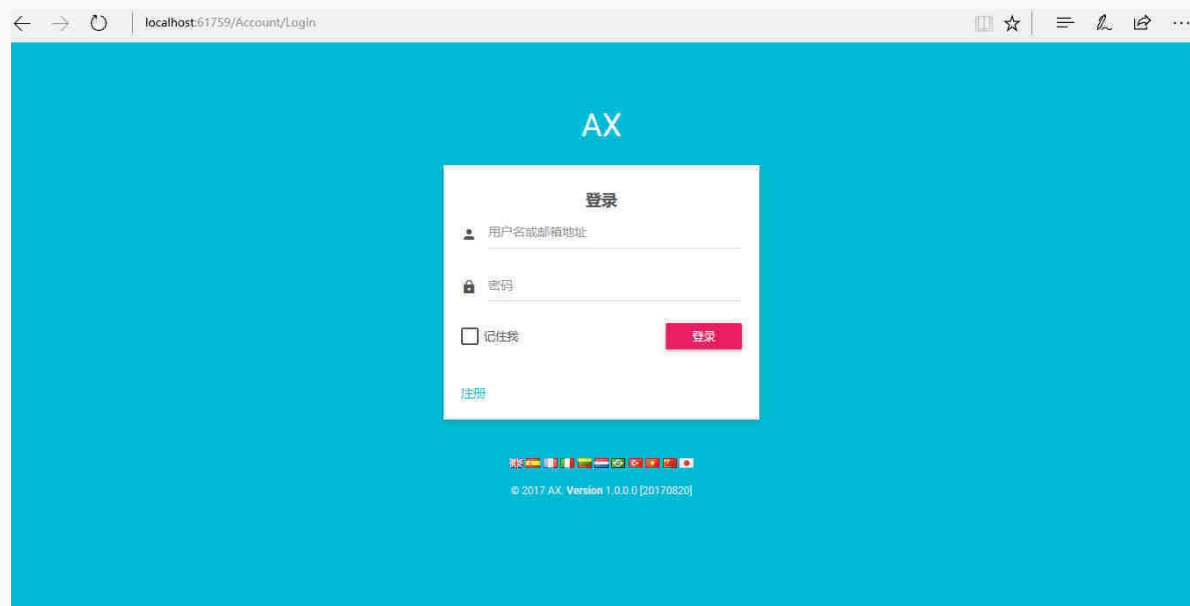
02：初始化数据库

1. 创建SQL数据库AX
2. 修改Web.Config 配置数据库连接字符串
3. 在VS的Package Manager Console窗口中使用 update-database初始化数据库



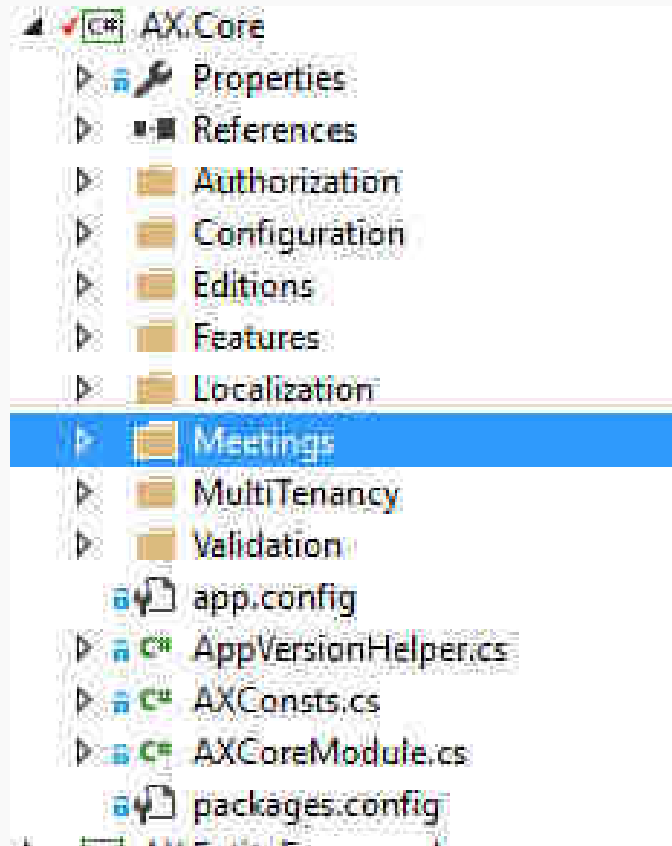
03：运行项目

- 登录用户名:admin
- 密码: 123qwe



演练二：实现AX.Core

01 : 在AX.Core项目新建文件夹“Meetings”



02 : 在“Meetings”目录下新建Meeting.cs

```
public class Meeting:FullAuditedEntity<int>
{
    public const int MaxSubjectLength = 50;
    public const int MaxLocationLength = 200;
    public const int MaxMoreInfoLength = 500;

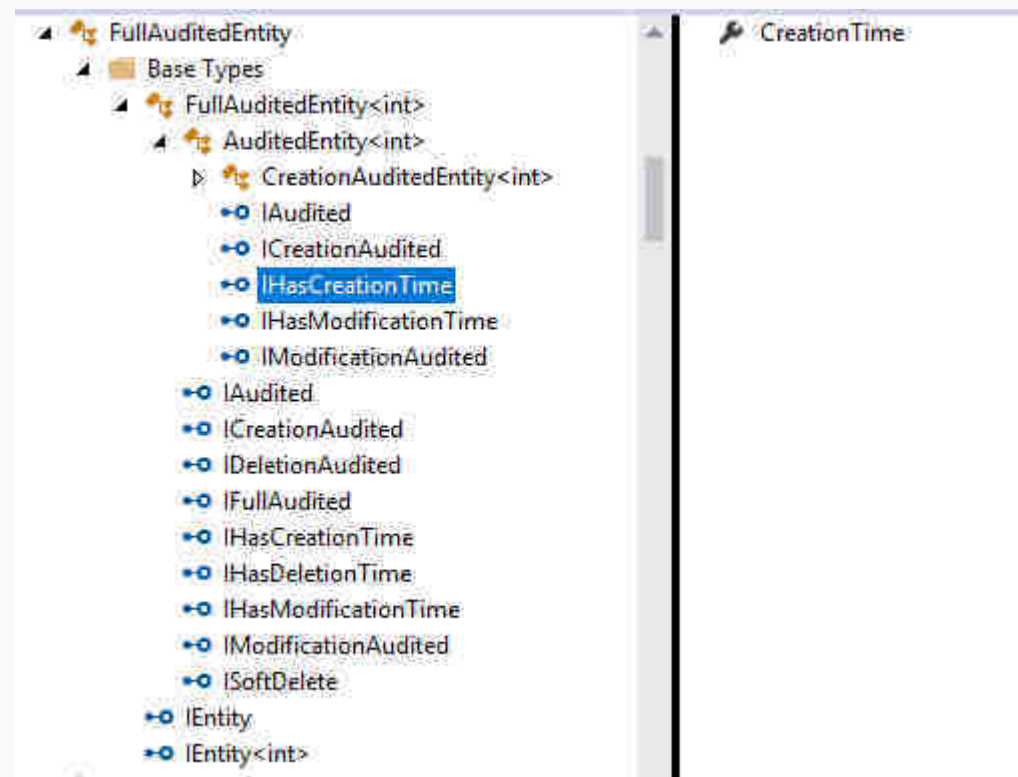
    [Required]
    [StringLength(MaxSubjectLength)]
    public virtual string Subject { get; set; }

    [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; }
}
```



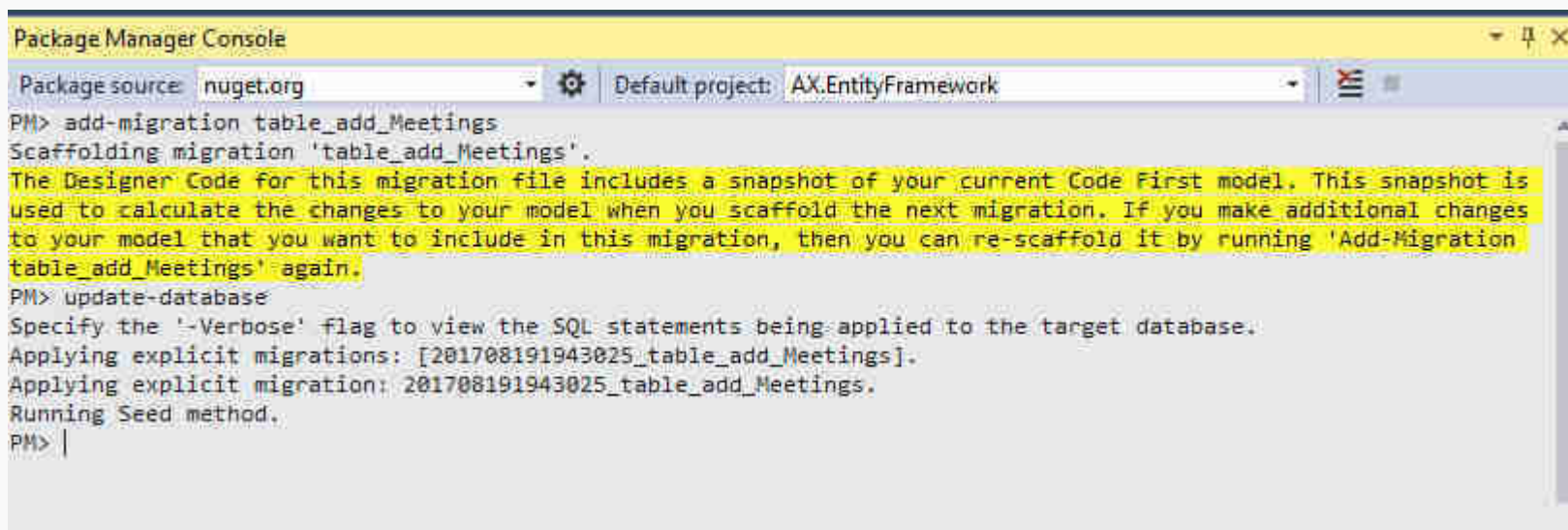
03: 更新AX.EntityFramework

1. 在AX.EntityFramework项目中打开
EntityFramework\AXDbContext.cs
2. 添加如下代码：

```
public IDbSet<Meetings.Meeting> Meetings { get; set; }
```

04：更新数据库

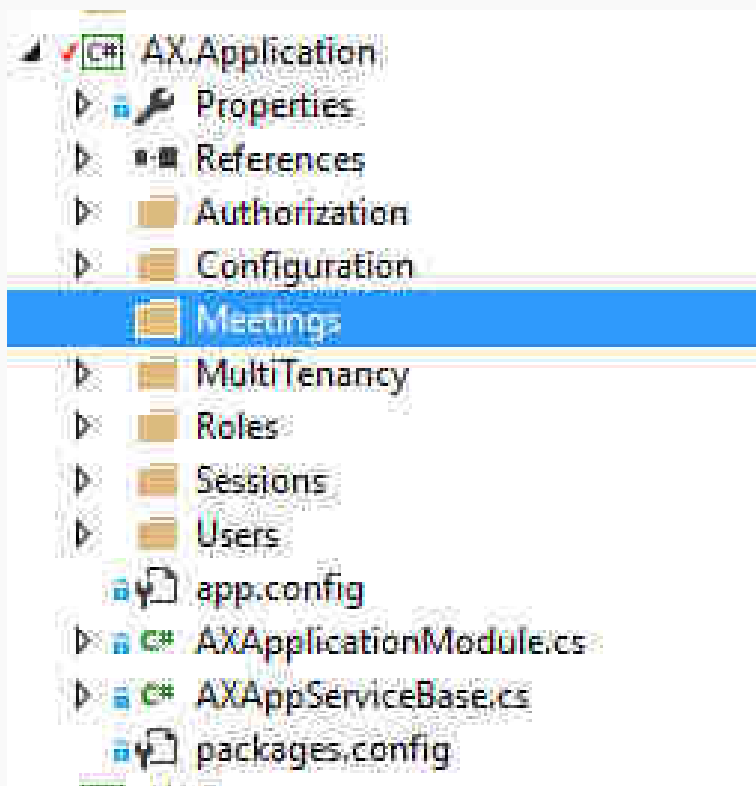
1. 在VS的Package Manager Console窗口中使用
add-migration table_add_Meetings命令
2. 然后运行update-database命令



```
Package Manager Console
Package source: nuget.org
Default project: AX.EntityFramework
PM> add-migration table_add_Meetings
Scaffolding migration 'table_add_Meetings'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is
used to calculate the changes to your model when you scaffold the next migration. If you make additional changes
to your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration
table_add_Meetings' again.
PM> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201708191943025_table_add_Meetings].
Applying explicit migration: 201708191943025_table_add_Meetings.
Running Seed method.
PM> |
```


演练三：AX.Application

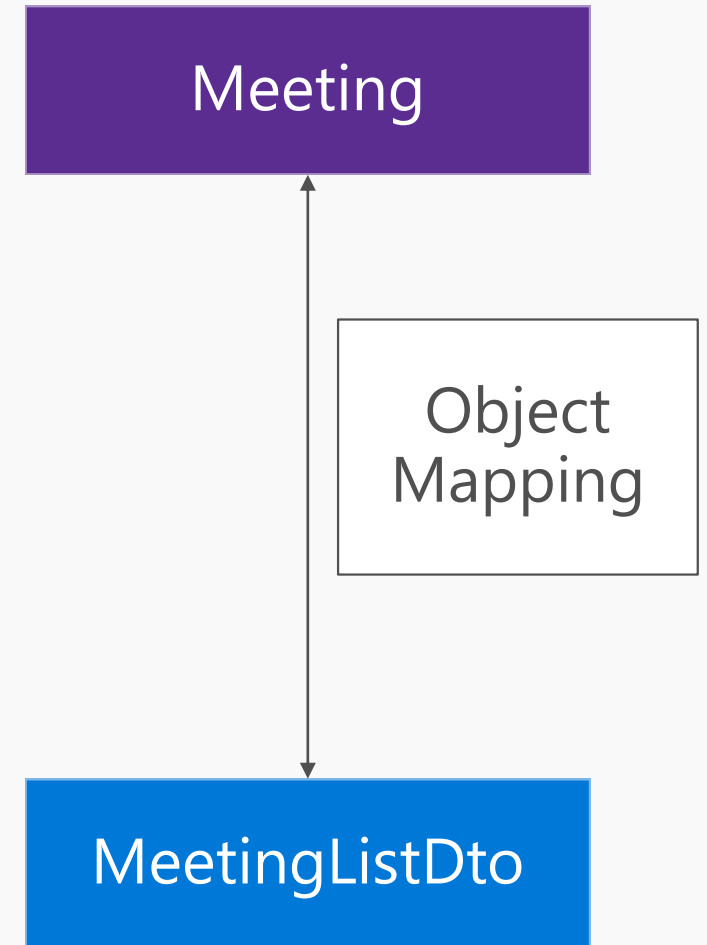
01：在AX.Application项目中新建文件夹“Meetings”



02：在“Meetings” 目录下创建DTOs

1. 新建“Dto”目录
2. 在“Dto”目录下新建MeetingListDto.cs及MeetingDto.cs

```
[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; }  
}
```



```
[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; }  
}
```

03：定义IMeetingAppService接口

在Meetings目录下，新建IMeetingAppService.cs

```
public interface IMeetingAppService:ICrudAppService<MeetingDto>
{
    PagedResultDto<MeetingListDto> GetAllList(PagedAndSortedResultRequestDto
input);
    Task SendNotification(int id);
}
```

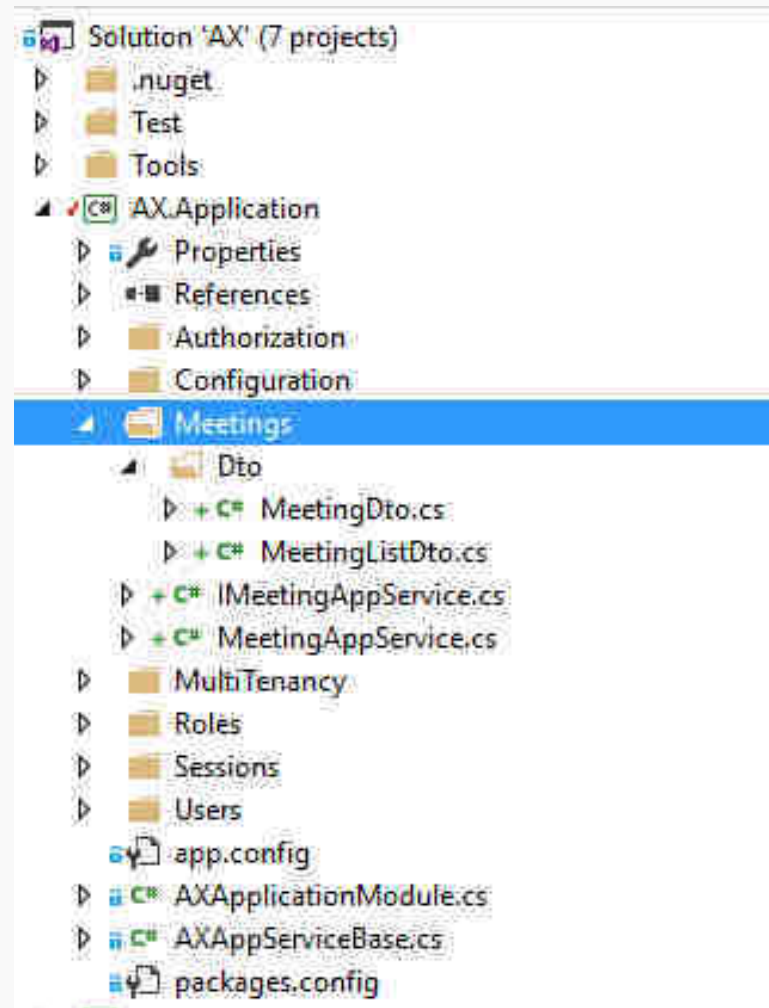
04：定义MeetingAppService

在Meeting目录下，新建MeetingAppService.cs

```
public class MeetingAppService : CrudAppService<Meeting, MeetingDto>, IMeetingAppService
{
    public MeetingAppService(IRepository<Meeting> repository) : base(repository)
    {
    }

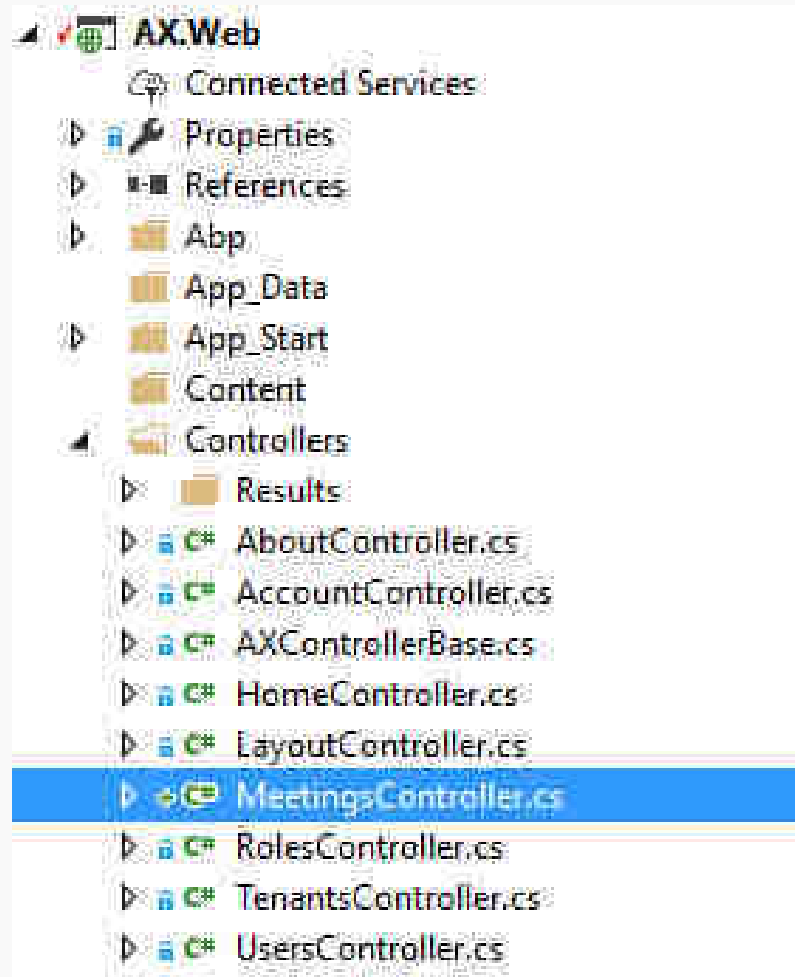
    public PagedResultDto<MeetingListDto> GetAllList(PagedAndSortedResultRequestDto input)
    {
        var list = this.Repository.GetAll().OrderBy(x =>
x.CreationTime).Skip(input.SkipCount).Take(input.MaxResultCount);
        return new PagedResultDto<MeetingListDto>
        {
            Items = Mapper.Map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

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

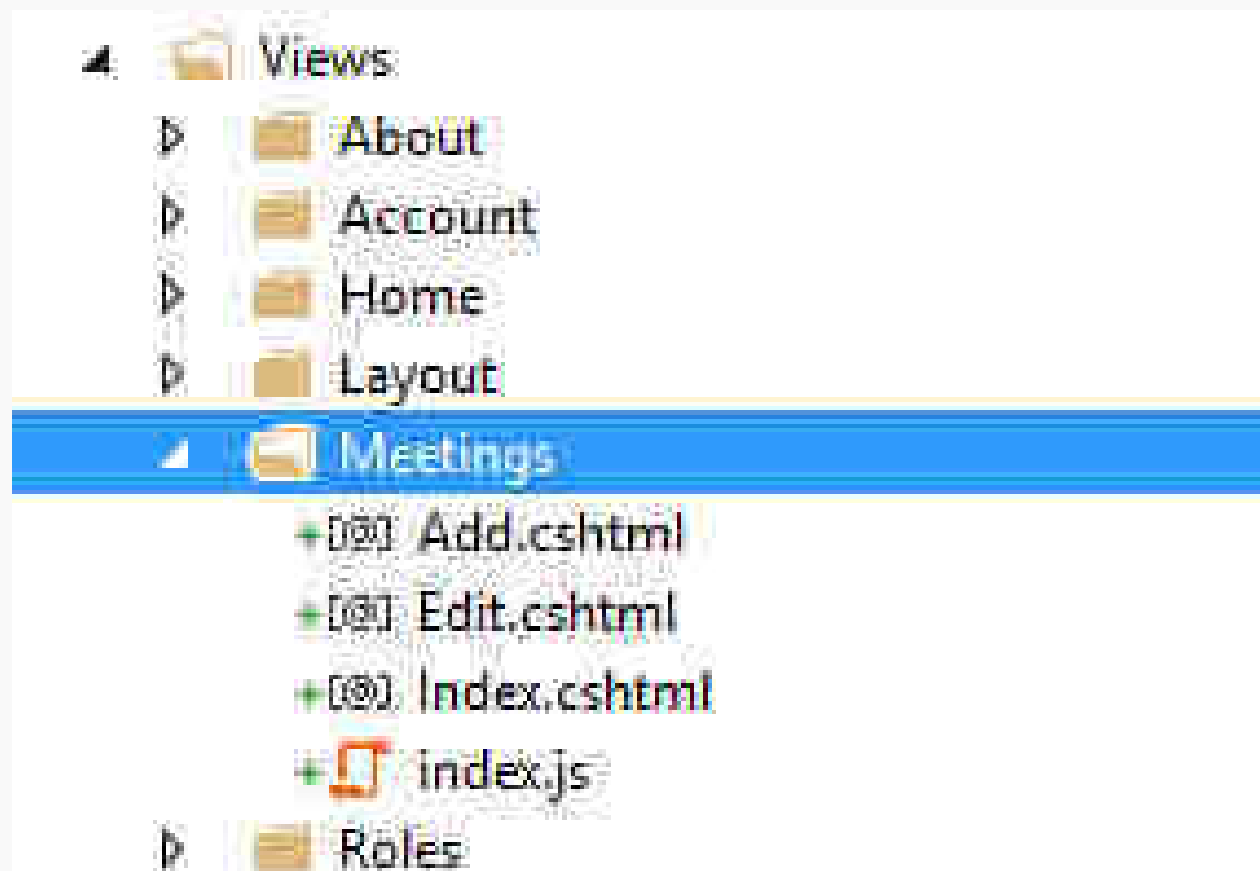



演练四：AX.Web

01：新建MeetingsController.cs



02 : 新建Meetings Views



03：添加页面导航


1. 打开AX.Web\App_Start\AXNavigationProvider.cs
2. 添加如下代码：

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

04：运行程序

- 添加会议数据

AX



admin
admin@defaulttenant.com

主頁

Meetings



用戶


Roles

關於我們

[Multi level menu]

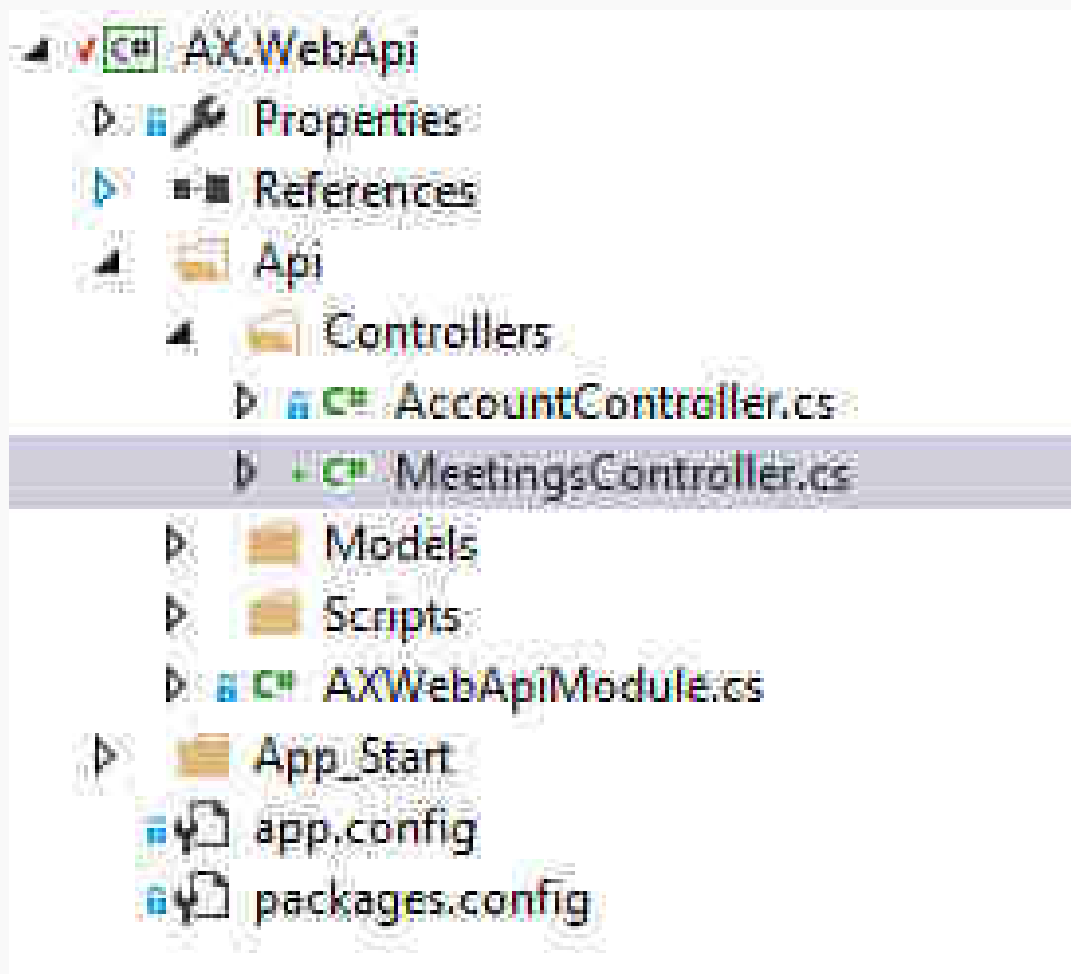
會議

會議名稱	地點	開始時間	結束時間	[Actions]
Azure & Xamarin Workshop第1天	深圳	2017/8/22 9:00:00	2017/8/22 16:00:00	
Azure & Xamarin Workshop第2天	深圳	2017/8/23 9:00:00	2017/8/23 16:00:00	



演练五：AX.WebApi

01 : 在AX.WebApi中添加MeetingsController



02：运行程序，通过swagger测试api

- <http://localhost:61759/swagger/ui/index>

The screenshot displays the Swagger UI interface. At the top, there is a green header bar with the Swagger logo, a text input field containing 'http://localhost:61759/swagger/docs/v1', another text input field labeled 'api_key', and an 'Explore' button.

On the left side, under the heading 'AX', there is a list of API endpoints. Each endpoint has a 'Show/Hide' link and a 'List Operator' link. The endpoints listed are:

- AbpCache
- AbpServiceProxies
- Account
- app_account
- app_configuration
- app_meeting
- app_role
- app_session
- app_tenant
- app_user
- Meetings

On the right side, the 'Request URL' is set to 'http://localhost:61759/api/meeting/list'. Below it, the 'Response Body' is displayed as a JSON object:

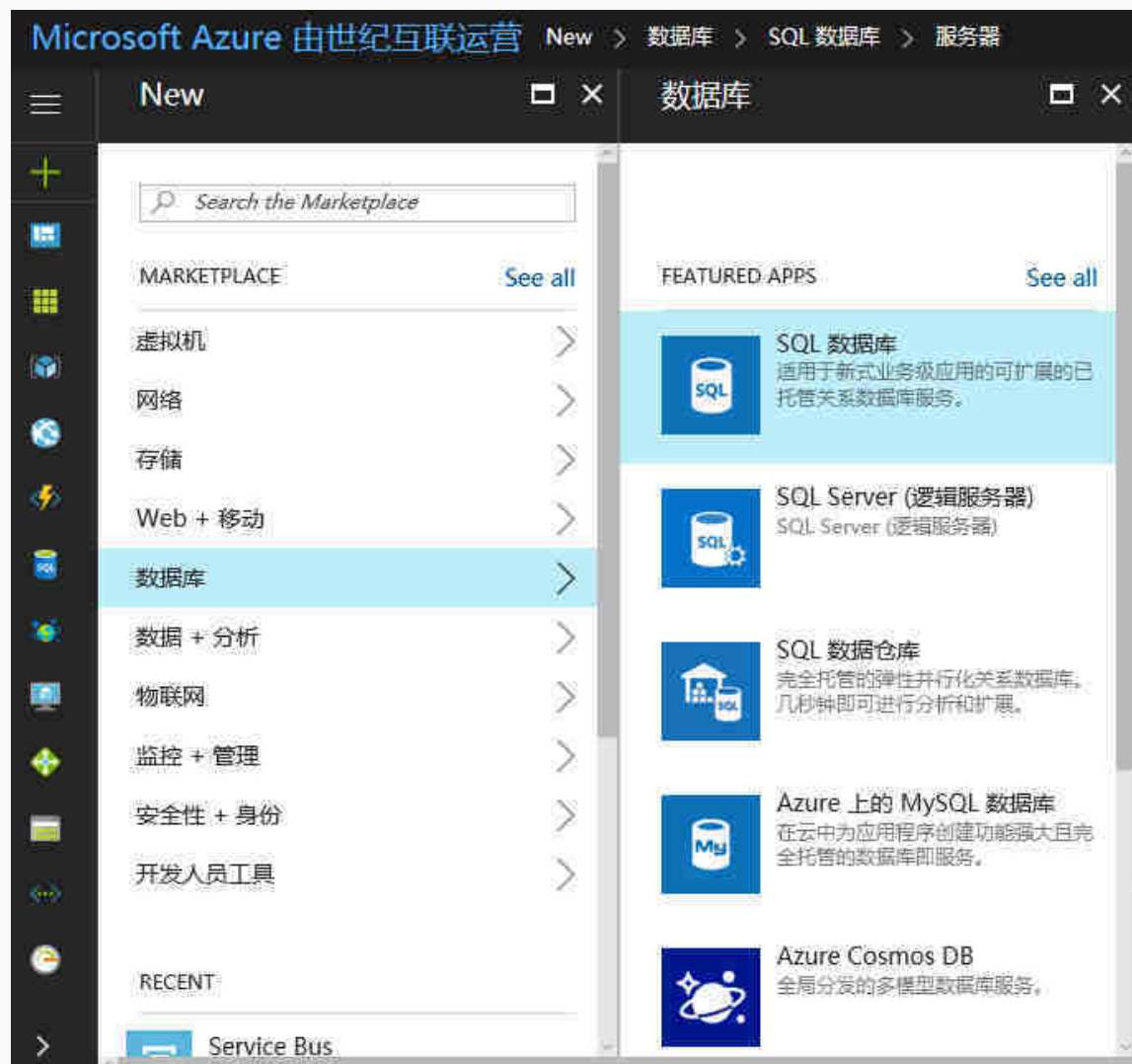
```
{
  "totalCount": 2,
  "items": [
    {
      "subject": "Azure & Xamarin Workshop第2天",
      "location": "深圳",
      "startTime": "2017-08-23T09:00:00",
      "endTime": "2017-08-23T16:00:00",
      "moreInfo": null,
      "id": 2
    },
    {
      "subject": "Azure & Xamarin Workshop第1天",
      "location": "深圳",
      "startTime": "2017-08-22T09:00:00",
      "endTime": "2017-08-22T16:00:00",
      "moreInfo": null,
      "id": 1
    }
  ]
}
```

At the bottom right, the 'Response Code' is shown as '200'.

将AX部署到Azure

<https://portal.azure.cn>

01 创建Azure SQL数据库



SQL 数据库

数据库名称

ax

订阅

Pay-In-Advance

资源组

新建

使用现有项

project_ax

选择源

空白数据库

服务器

配置所需的设置

想要使用 SQL 弹性池?

是

不是现在

定价层

Standard S2: 50 DTU, 250 GB

固定到仪表板

创建

自动化选项

服务器

创建一个新服务器

新服务器

服务器名称

projectax

服务器管理员登录

axuser

密码

.....

确认密码

.....

位置

中国北部

允许 Azure 服务访问服务器

选择

记住此用户名和密码

SQL 数据库

数据库名称

ax

订阅

Pay-In-Advance

资源组

新建

使用现有项

project_ax

选择源

空白数据库

服务器

projectax (中国北部)

想要使用 SQL 弹性池?

是

不是现在

定价层

Basic: 5 DTU, 2 GB

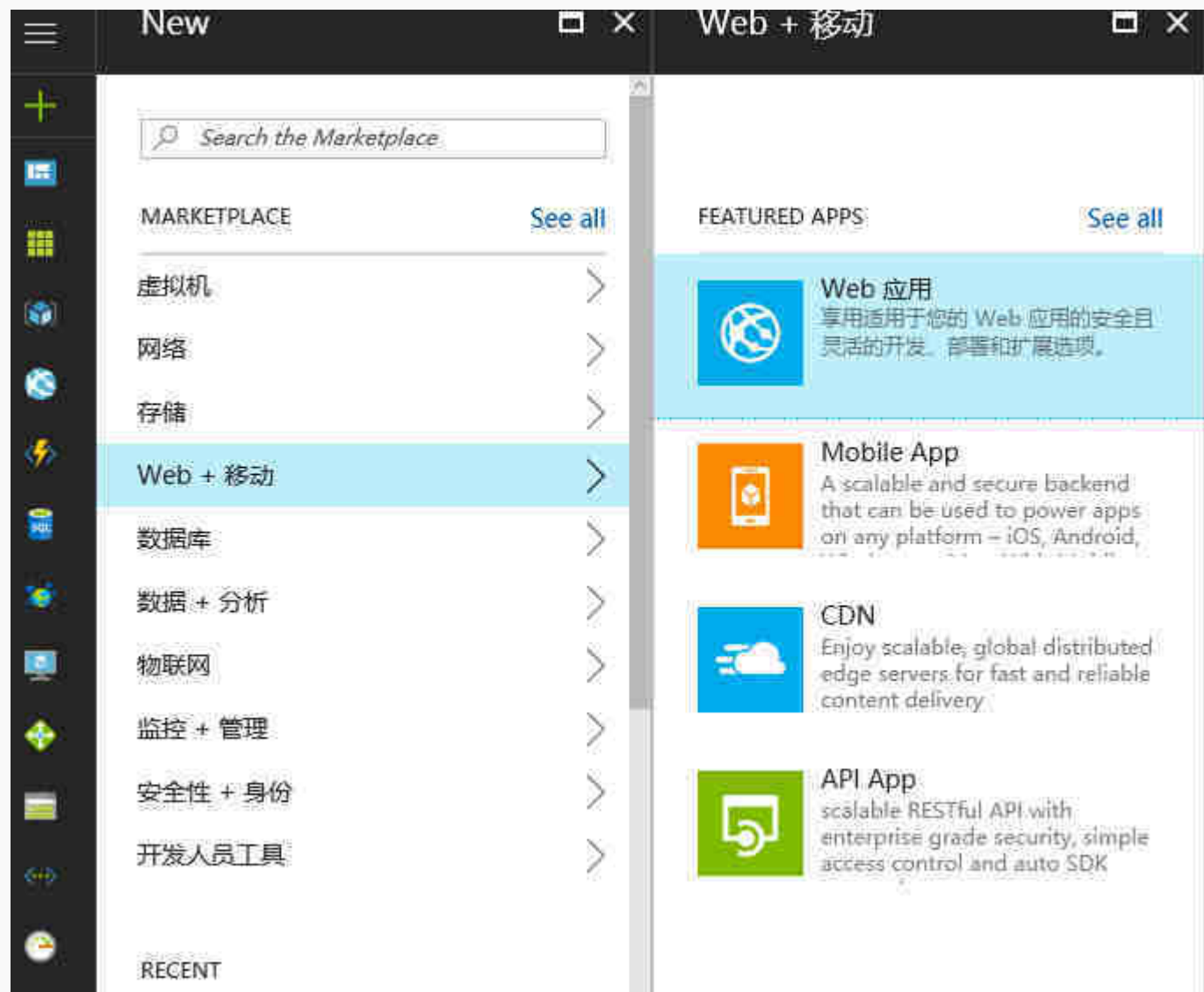
固定到仪表板

创建

自动化选项

- 获取数据库连接字符串，配置Web.config

02 创建Azure应用服务



Web 应用

创建

- * 应用名称
projectax ✓
chinacloudsites.cn
- * 订阅
Pay-In-Advance
- * 资源组 ⓘ
☐ 新建 ☒ 使用现有项
project_ax
- * 应用服务计划/位置
ServicePlan9211d0fa-ab1e(Chi... >

☐ 固定到仪表板

[创建](#) [自动化选项](#)

新建应用服务计划

创建一个 Web 应用计划

* 应用服务计划

ServicePlanProjectAX

* 位置

China North

* 定价层

S1 标准

确定

选择你的定价层

浏览可用计划及其功能

 每日 50 次备份	 每日 50 次备份	 每日 50 次备份
 流量管理器 地区可用性	 流量管理器 地区可用性	 流量管理器 地区可用性
1,466.87 CNY/月(估计值)	2,933.74 CNY/月(估计值)	5,867.48 CNY/月(估计值)
S1 标准	S2 标准	S3 标准
1 核心	2 核心	4 核心
1.75 GB RAM	3.5 GB RAM	7 GB RAM
 50 GB 存储	 50 GB 存储	 50 GB 存储
 自定义域/SSL SNI Ind 和 IP SSL 支持	 自定义域/SSL SNI Ind 和 IP SSL 支持	 自定义域/SSL SNI Ind 和 IP SSL 支持
 多达 10 个实例 自动缩放	 多达 10 个实例 自动缩放	 多达 10 个实例 自动缩放
 每天备份	 每天备份	 每天备份
 5 个槽 Web 应用暂存	 5 个槽 Web 应用暂存	 5 个槽 Web 应用暂存
 流量管理器 地区可用性	 流量管理器 地区可用性	 流量管理器 地区可用性
443.20 CNY/月(估计值)	886.40 CNY/月(估计值)	1,772.95 CNY/月(估计值)
B1 基本	B2 基本	B3 基本
1 核心	2 核心	4 核心

选择

Web 应用

创建

* 应用名称

projectax

.chinacloudsites.cn

* 订阅

Pay-In-Advance

* 资源组

☐ 新建 ☒ 使用现有项

project_ax

* 应用服务计划/位置

ServicePlanProjectAX(China N...

☐ 固定到仪表板

创建

自动化选项

03 : 创建数据存储服务



- 获取存储服务访问密钥并配置Web.config



```
<add key="AzureBlobAccessKey" value="" />
<add key="AzureBlobAccount" value="" />
<add key="AzureBlobUrlPostfix" value="core.chinacloudapi.cn/" />
```

04：创建通知中心服务

- <https://manage.windowsazure.cn/>





projectax



通知中心

配置

新建



媒体服务



服务总集



自动化



CDN



计划程序



ACTIVE DIRECTORY

队列

主题

中继

通知中心

事件中心



快速创建

通知中心名称

ax



区域

中国东部



命名空间

projectax



servicebus.chinacloudapi.cn

创建新的通知中心





ax

☐ 启用未经身份验证的推送通知。

apple 通知设置

证书指纹

上传

删除



google cloud messaging 设置

GCM API 密钥



amazon device messaging 设置



客户端 ID

- 配置Web.config

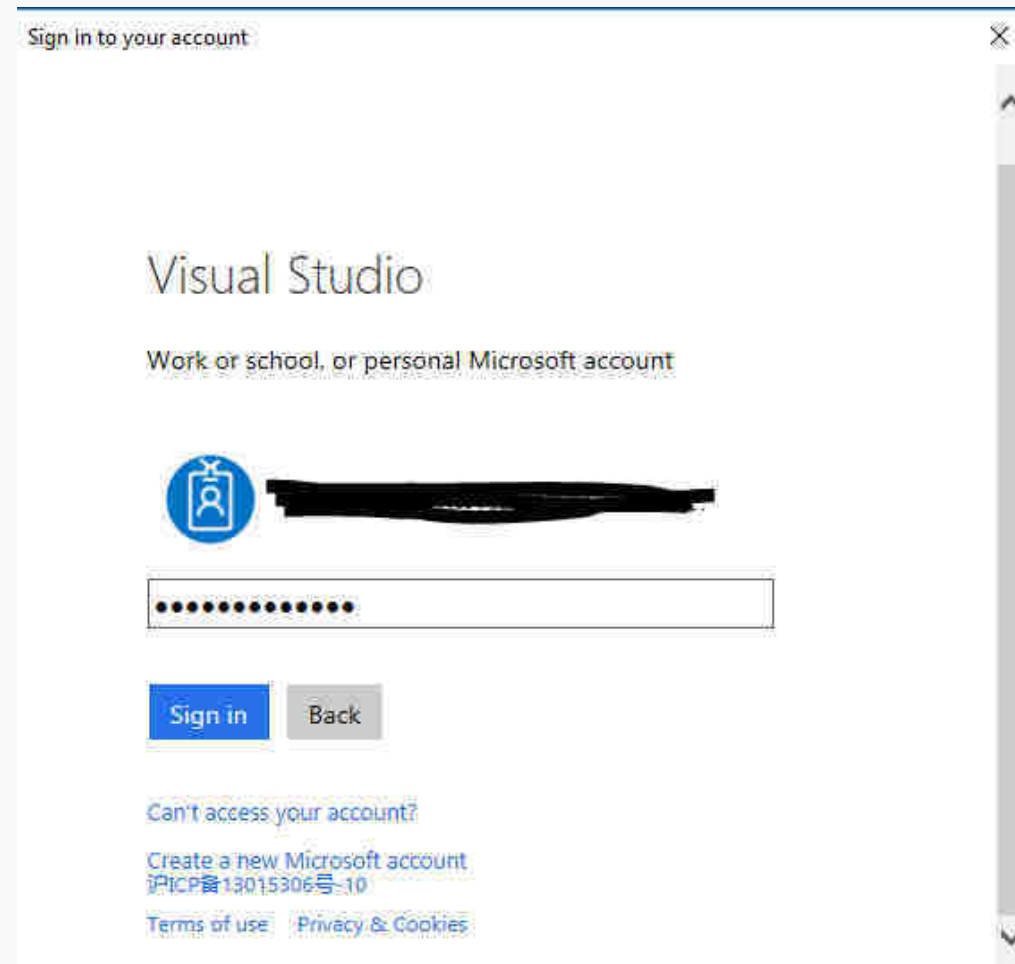
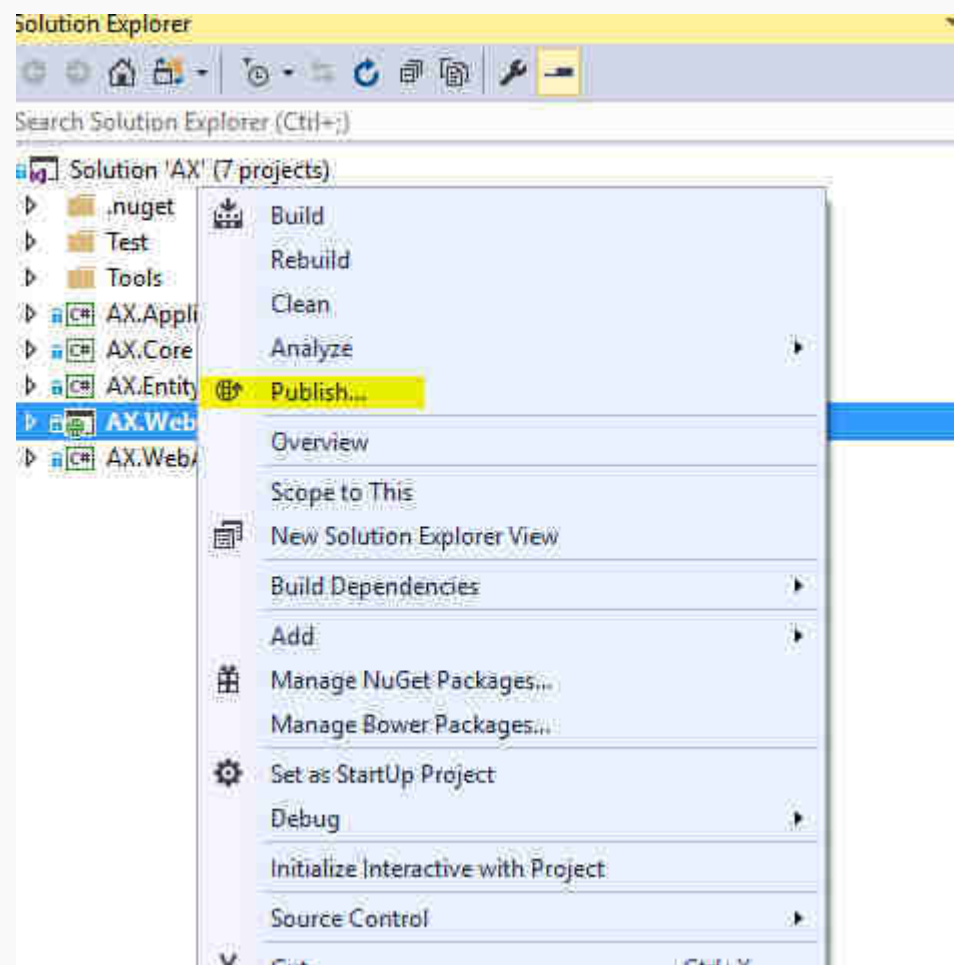
```
<add key="AzureNotificationHubConnectionString" value="" />  
<add key="AzureNotificationHubPath" value="" />
```

05：部署数据库到SQL Azure

- 修改SQL Azure防火墙设置
- 确认Web.config连接字符串
- 运行update-database命令

06：使用VS2017登录连接Azure (中国)

- <https://docs.azure.cn/zh-cn/articles/others/aog-portal-management-qa-vs2017-login>



Overview

Connected Services

Publish

Publish

Publish your app to Azure or another host. [Learn more](#)



Microsoft Azure
App Service



IIS, FTP, etc



Folder



☐ Create New

☒ Select Existing

Publish

App Service

Host your web and mobile applications, REST APIs, and more in Azure



Subscription

Pay-In-Advance

View

Resource Group

Search



OK

Cancel



Azure & Xamarin Workshop

Day 2

- Xamarin 介绍
- AX项目移动端开发
- AX项目调用Azure服务

Xamarin介绍

Xamarin Native approach



iOS C#



Android C#



Windows C#

Shared C# Logic

3 Native User Interfaces
Shared App Logic



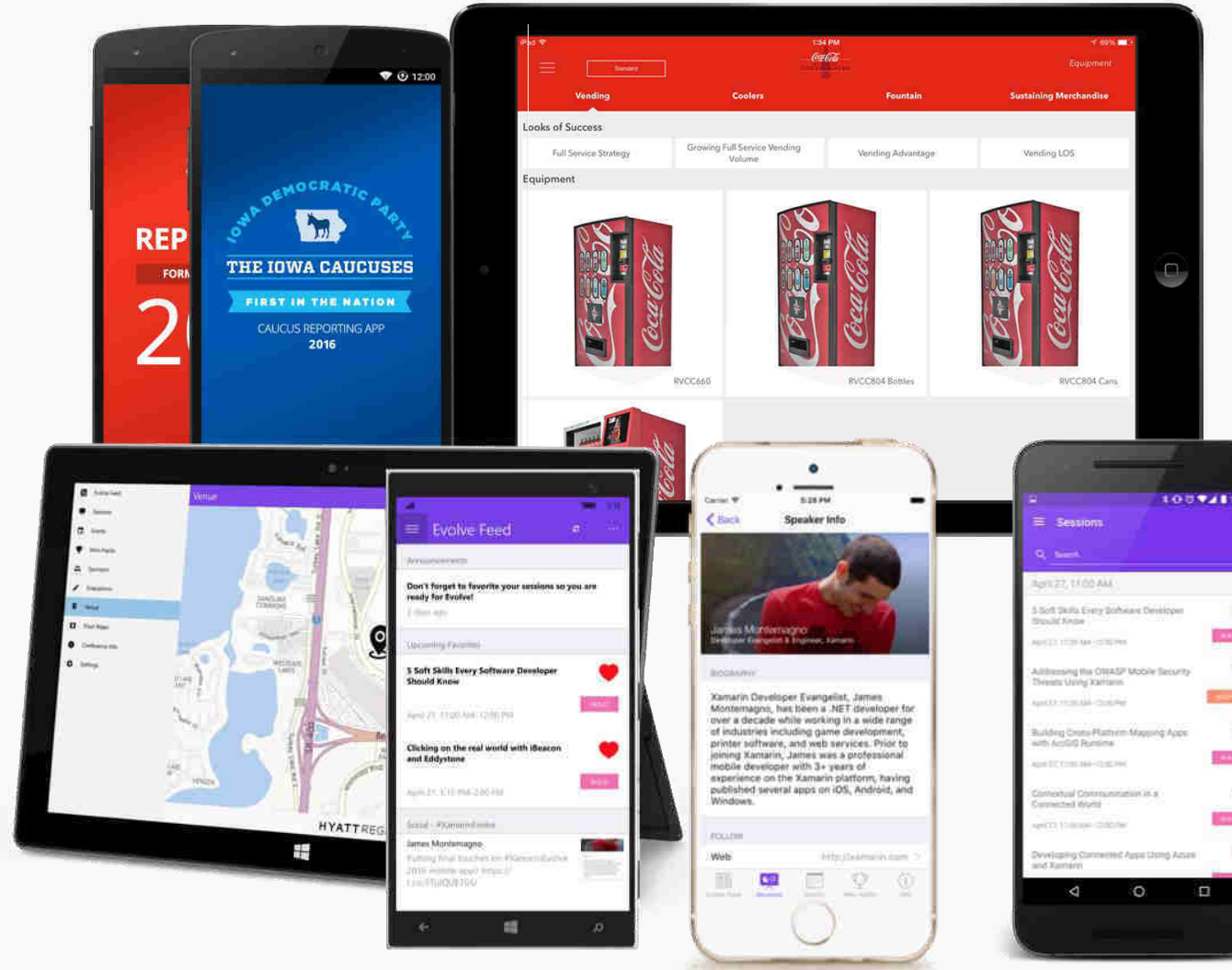
Xamarin.Forms approach



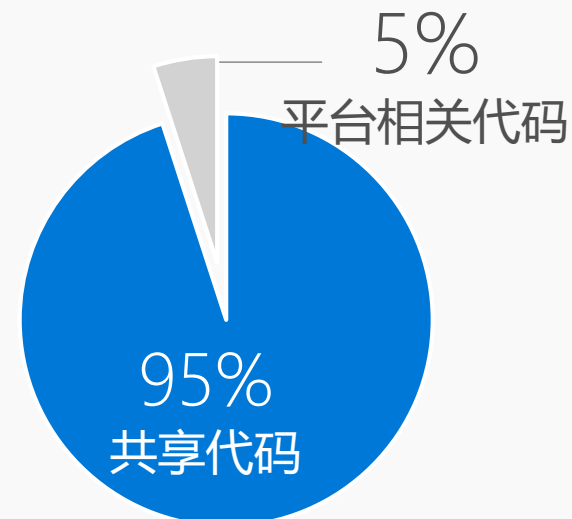
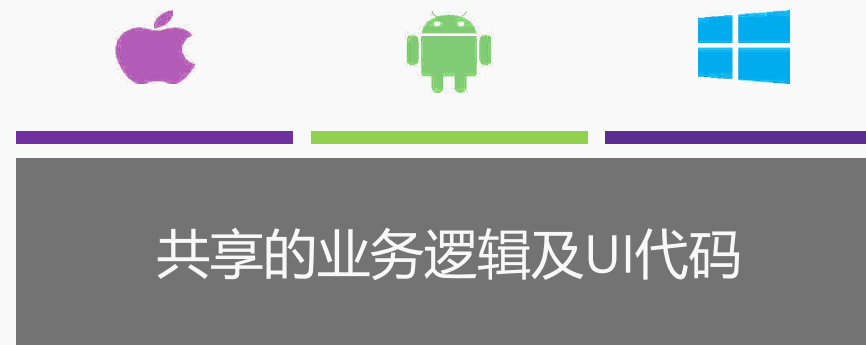
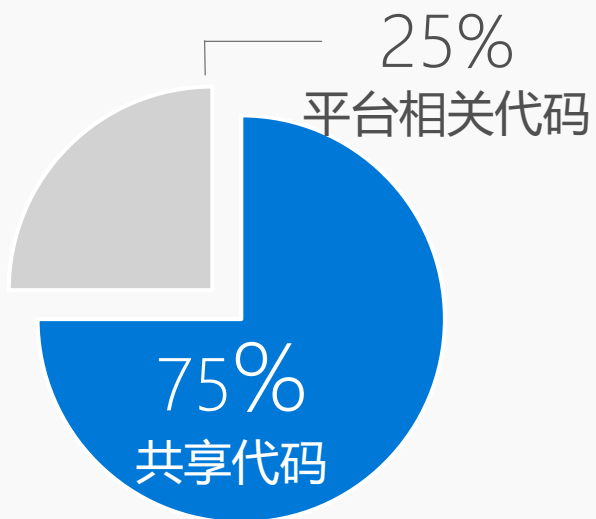
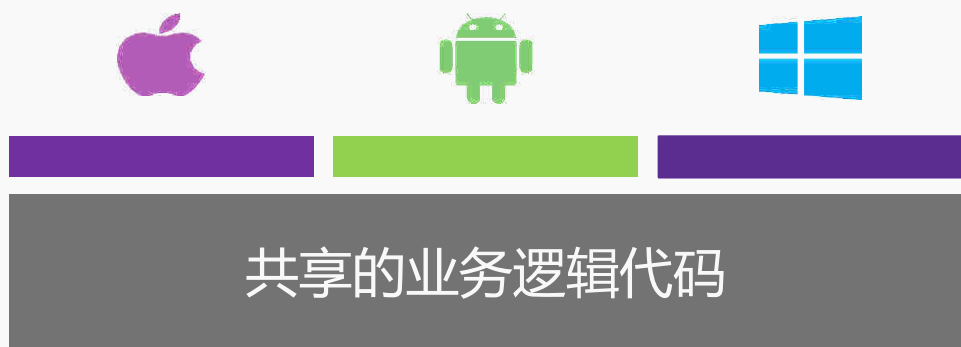
Shared UI Code

Shared C# Logic

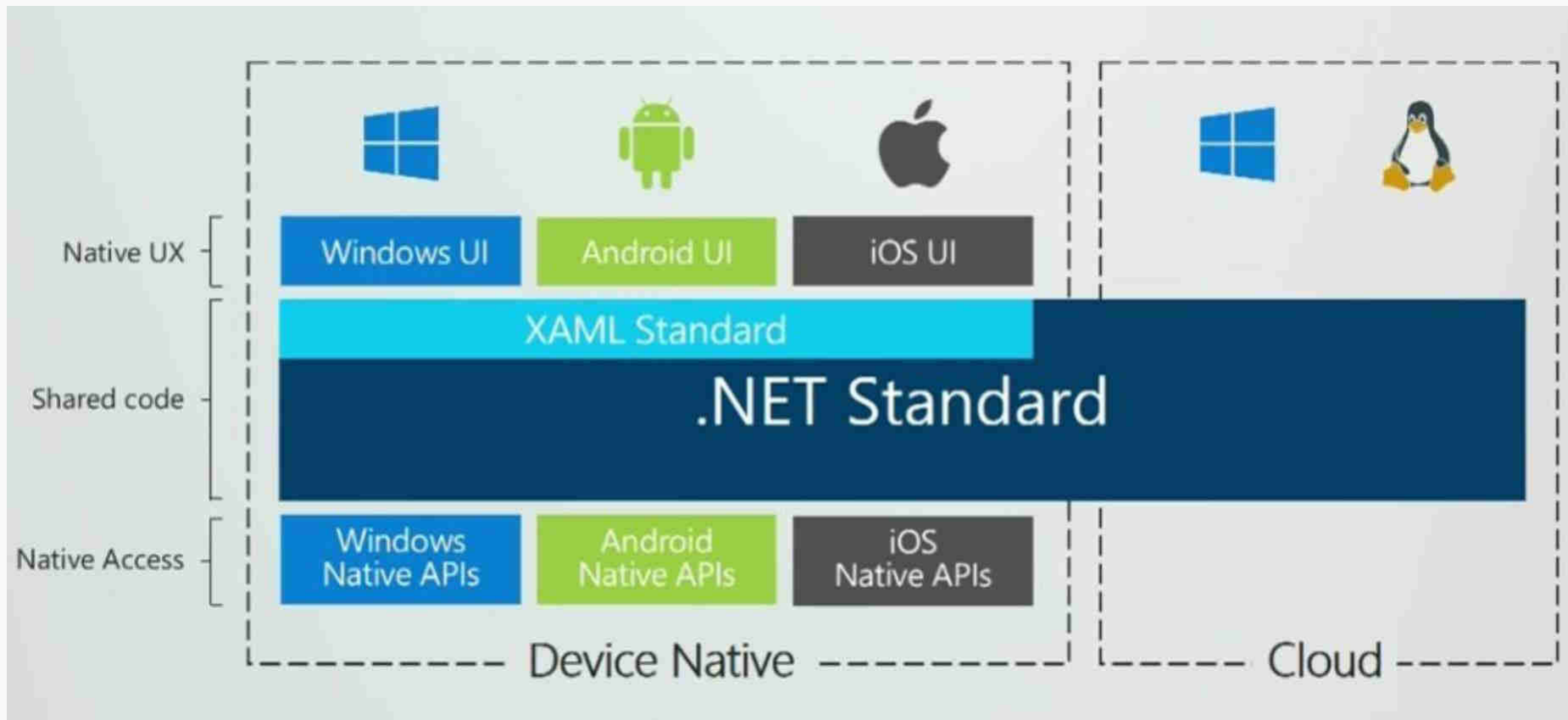
Shared (Native) User Interfaces
Shared App Logic



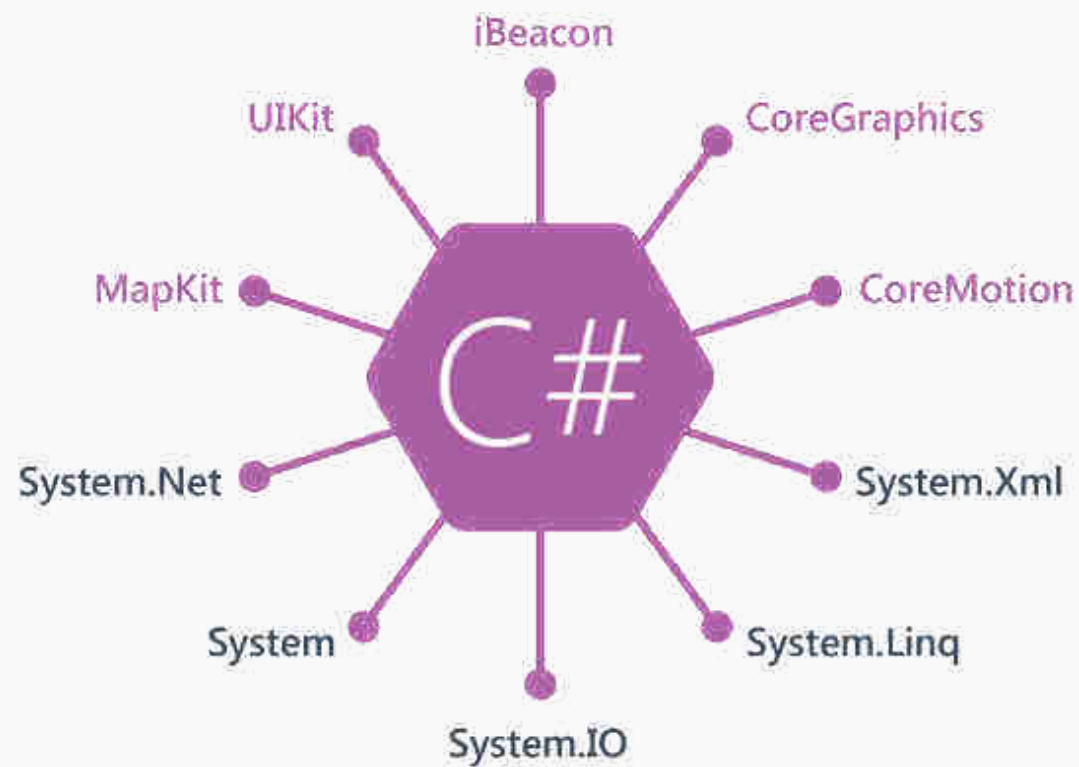
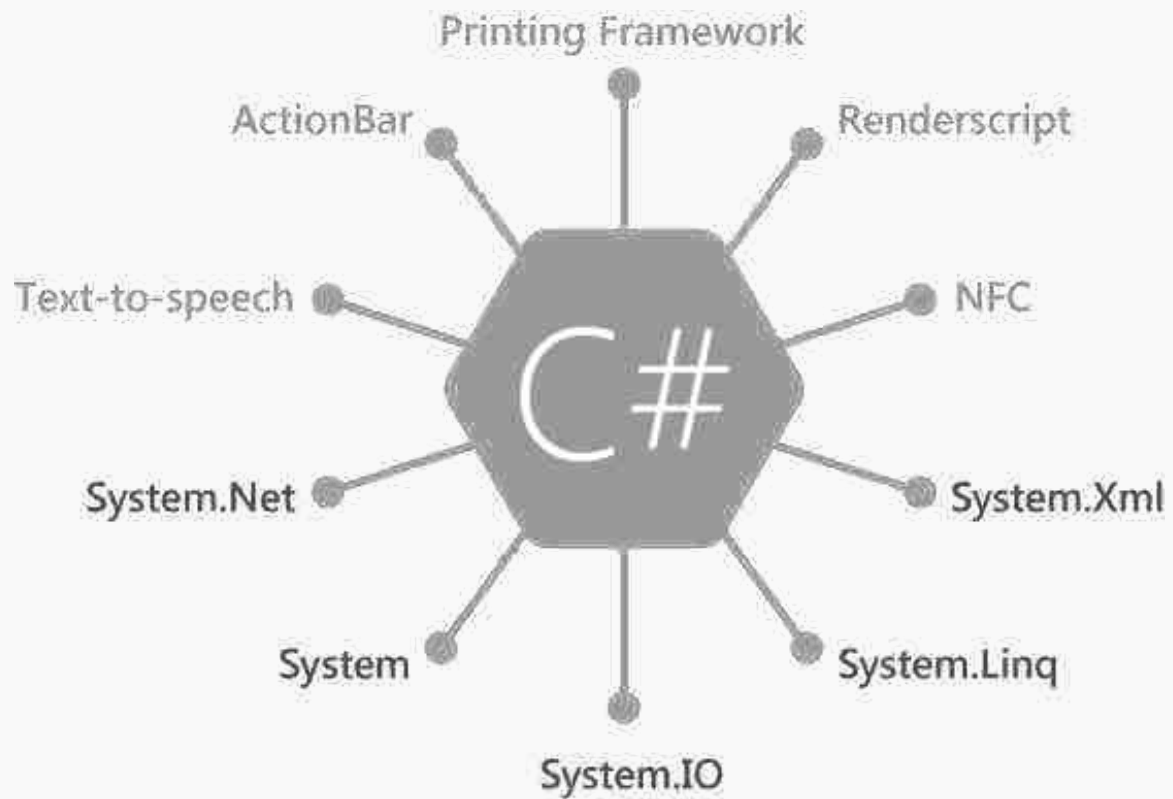
Xamarin Native, Xamarin Forms 代码共享



.NET (C#) 前端、后端全栈跨平台



100%原生API

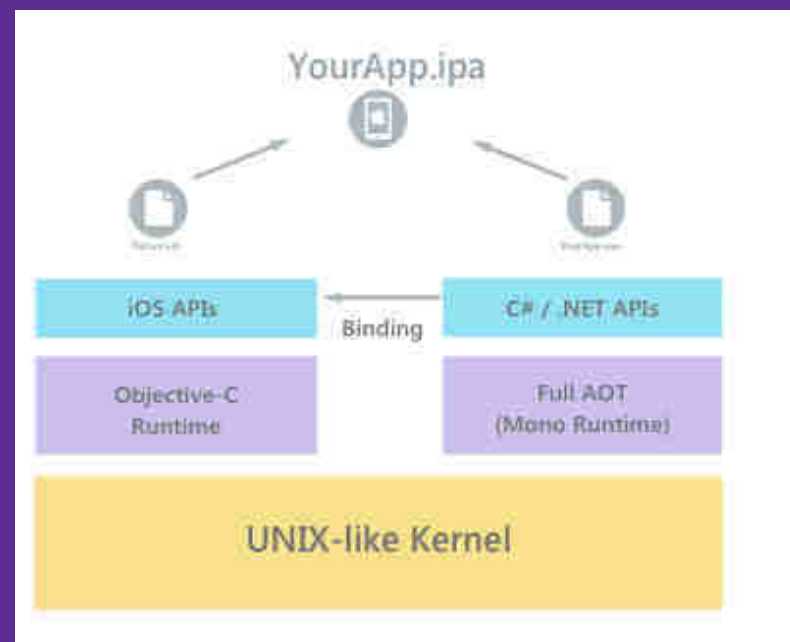
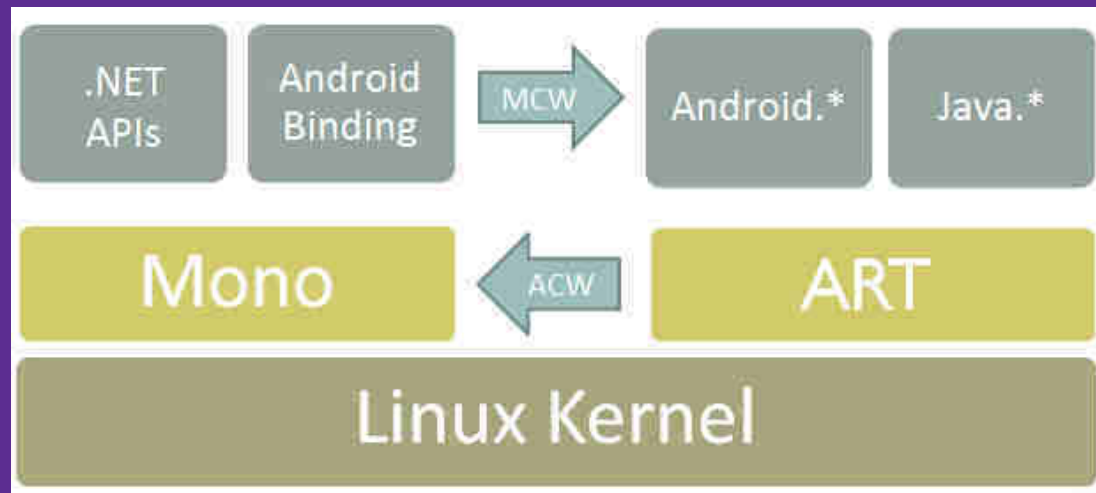


Xamarin底层技术

Mono

Binding (绑定)

P/Invoke (平台调用)



原生编译方式



Java, Objective-C / Swift 能做的

C#也能做！

Azure Mobile Center

 Test

 Analytics

 Build

 Distribute

 Live Update

 Crash Reporting

 Push Notifications



 Storage

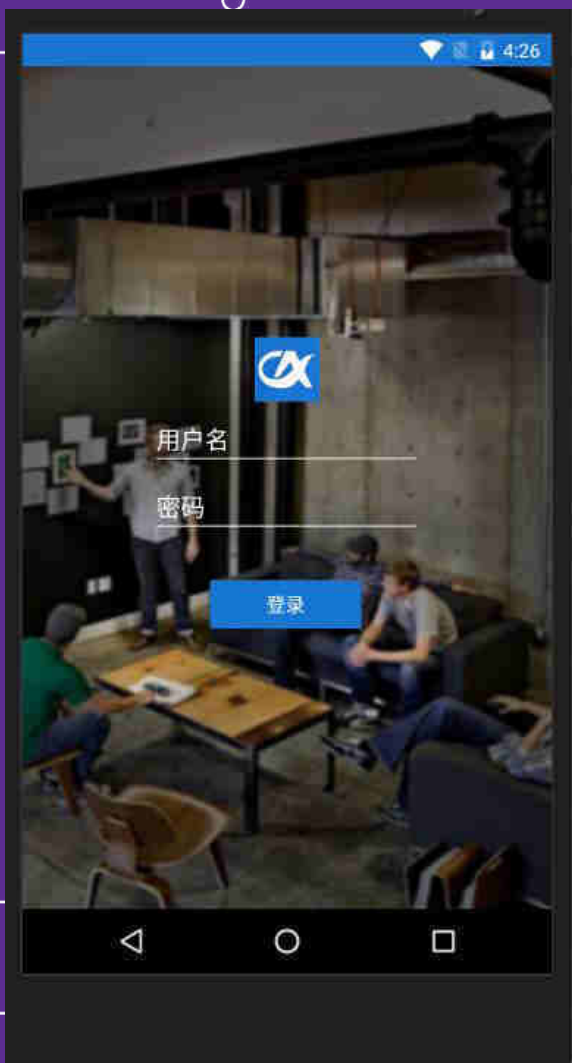
 Tables

 Identity

Azure Mobile Apps



Workshop项目 AX App介绍



AX App演示

推送

Photo used under CC <http://www.wocintechchat.com/>

演练一：创建AX App 项目

01：使用AXMobile项目模板

1. 下载项目模板

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

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

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

2. 恢复Nuget包，编译项目

演练二：创建App登录页

01 : AXMobile.Core

1. 在Pages目录中添加Login.xaml, 确认此文件属性中的Build Action = "EmbeddedResource"
2. 在Helpers目录下添加ClearStackHelper.cs
3. 在ViewsModels目录中添加LoginViewModel.cs
4. 修改MvxApp.cs代码如下：

```
RegisterAppStart<ViewModels.LoginViewModel>();
```


02 : AXMobile.Droid

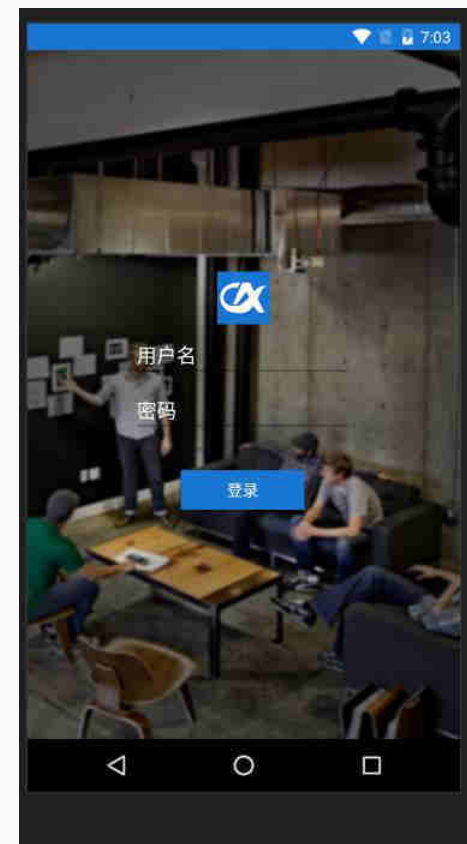
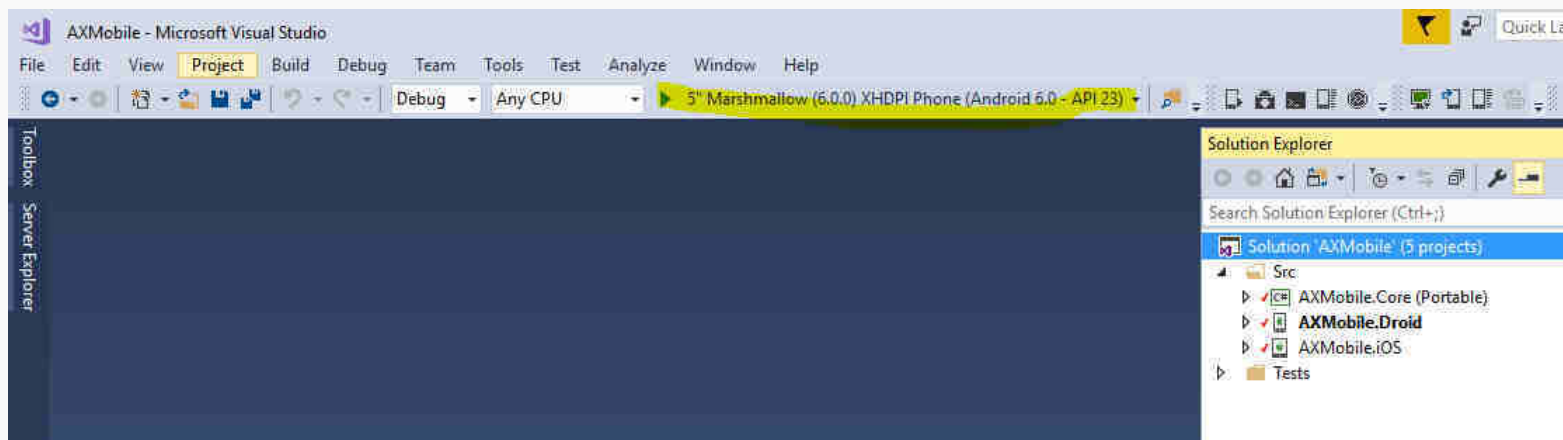
1. 在Resources/drwable目录下添加backgroundimage.png
2. 确认文件属性中的Build Action = "AndroidResource"

03: AXMobile.iOS

1. 在Resources目录下添加backgroundimage.png
2. 确认文件属性中的BuildAction = "BundleResource"

04：在VS中启动AXMobile.Droid

- 选择AXMobile.Droid为启动项
- 选择模拟器或真机，启动



05 : 在VS for Mac中启动AXMobile.iOS

演练三：创建App首页

01: AX.Core

1. 修改Pages目录下的MainPage.xaml
2. 修改ViewModels目录下的MainViewModel.cs

```
<TabbedPage.Children>
    <ContentPage Title="会议" Icon="news.png"/>
    <!--<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>
```

02 : AX.Droid

1. 在Resources/drawable目录下添加图片：

back.png, news.png, communication.png, information.png

2. 确认文件属性中的Build Action = "AndroidResource"

03: AXMobile.iOS

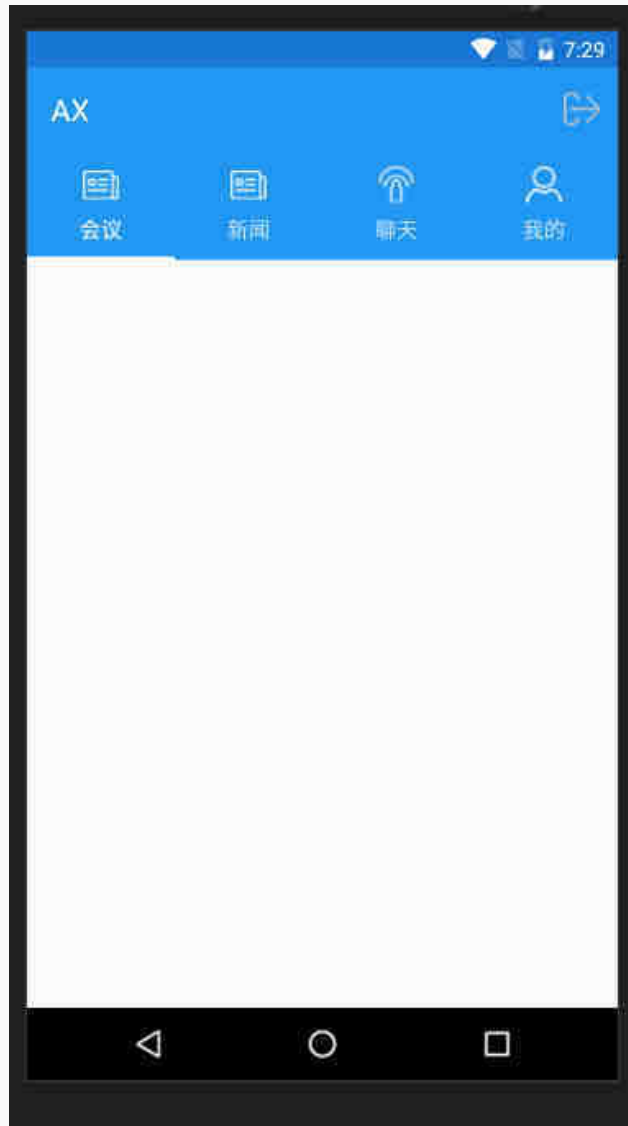
1. 在Resources目录下添加

back@2x.png, news@2x.png, communication@2x.png, information @2x.png

back@3x.png, news@3x.png, communication@3x.png, information@3x.png

2. 确认文件属性中的BuildAction = "BundleResource"

04：运行App



演练四：创建App新闻页

01: AX.Core

- 在Pages目录下添加MeetingListPage.xaml, 确认此文件属性中的Build Action = "EmbeddedResource"
- 在ViewModels目录下添加MeetingListViewModel.cs
- 在Helpers目录下添加HttpClientHelper.cs
- 添加AppConstants.cs, 配置如下常量为AX API根路径

```
public const string Base_API_Url = "";
```

- 在Services目录下添加BaseErrorDto.cs和CommonErrorDto.cs
- 在Services目录下添加“Meetings”目录，在其中添加IMeetingService.cs, MeetingService.cs, MeetingListItemDto.cs, MeetingsDto.cs
- 在Model目录下添加MeetingListItem.cs

- 修改MainPage.xaml及MainPageViewModel.cs

```
<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>
```

02 : AXMobile.Droid

- 在Resources/drawable/中添加blue.png, purple.png
- 确认文件属性中的Build Action = "AndroidResource"

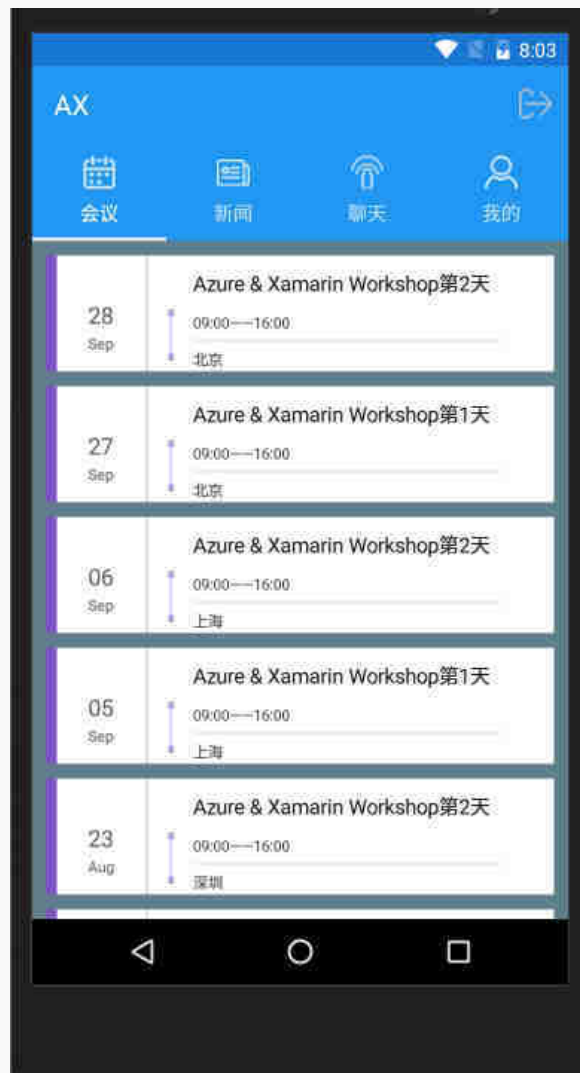
03: AXMobile.iOS

- 在Resources中添加

blue@2x.png, purple@2x.png, blue@3x.png,
purple@3x.png

- 2.确认文件属性中的BuildAction =
"BundleResource"

运行App



演练五：推送

01: 所有项目

- 安装nuget: ksemenenko.AzurePushNotifications

02 : AXMobile.Core

- 配置AppConstants.cs的常量

```
public const string Azure_Notification_Hub_Name = "";  
public const string Azure_Notification_Hub_ListenConnectionString = "";
```

03: AXMobile.iOS

- 在AppDelegate.cs中添加如下代码

```
public override void ReceivedRemoteNotification(UIApplication application,
NSDictionary userInfo)
{
    CrossAzurePushNotifications.Platform.ProcessNotification(userInfo);
}

public override void RegisteredForRemoteNotifications(UIApplication
application, NSData deviceToken)
{
    CrossAzurePushNotifications.Platform.RegisteredForRemoteNotifications(deviceToken
);
}
```

感谢

请填写反馈表

联系 jameszhou@highviewsoftware.com

Xamarin 企业开发D2D培训

讲师由微软MVP (Visual Studio Xamarin 方向),
Xamarin认证移动开发者、具备多年Xamarin技术经验的开发者 (Dev)担任,
培训对象为具备.NET基础的开发者(Dev),
特别针对企业现有的ASP.NET/Win Form/WPF等技术团队量身打造。



<http://www.highviewsoft.com/service/training>

Photo used under CC <http://www.wocintechchat.com/>