



DevCon School

Технологии будущего

# Правильный Xamarin для мобильных приложений

Черников Вячеслав

Binwell

## Что такое Xamarin.Forms

---

Знакомимся с Xamarin и создаем первое приложение на Xamarin.Forms

## Работаем с REST

---

Подключаемся к внешнему REST-сервису, знакомимся с Mock-данными

## Архитектура приложения

---

Отделяем бизнес-логику от UI, упрощаем описание ViewModel, знакомимся с архитектурой

# Что такое Xamarin.Forms

#msdevcon

# Классический Xamarin

Полное покрытие iOS/Android API, нативные приложения

C# для разработки, возможность шарить код

Xamarin.Forms

Унификация и абстрагирование UI

Xaml, MVVM

PCL-библиотека

# Xamarin.Forms

Приложения для работы с данными

Прототипы

Бизнес-приложения

# Xaml 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="XamlSamples.HelloXamlPage"  
    Title="Page Title goes here">  
    <Label Text="{Binding Model.Text}" x:Name="MainLabel"  
        FontAttributes="Bold"  
        TextColor="Aqua" />  
</ContentPage>
```



# Основные элементы

Label – отображение текста

Image – отображение изображений

Button – кнопка

ListView – список элементов

StackLayout – компоновка элементов

ContentPage – простая страница

MasterDetailPage – страница с боковым меню

# Приложение v0.1

Хардкодим и осваиваем UI

## Улучшения

-- *FFImageLoading*

```
<ffimageloading:CachedImage DownsampleToViewSize="true"  
    Source = "http://loremflickr.com/600/600/nature?filename=simple.jpg" >  
</ffimageloading:CachedImage>
```

-- *ListView caching*

```
<ListView CachingStrategy="RecycleElement">
```

-- *XamlC, AssemblyInfo.cs*

```
[assembly: XamlCompilation(XamlCompilationOptions.Compile)]
```



# Работаем с REST API

#msdevcon

# Json.Net и DTO

```
public class Session {  
  
    [JsonProperty("title")]  
    public string Title;  
  
}
```

# Refit

```
public interface IRemoteApi
```

```
{
```

```
    [Get("/sessions")]
```

```
    Task<List<Session>> GetSessions();
```

```
}
```

```
var service =
```

```
RestService.For<IRemoteApi>("https://demo3143189.mockable.io");
```

```
var sessions = await service.GetSessions();
```

# Mock

Ускоряет верстку

Дает возможность проверить различные сценарии и наборы данных

Сохраняем готовые JSON в качестве ресурсов

# DAL

Data Interfaces (ISessionDataService, etc), Data Objects

## Mock

Реализация Data Interface и чтение данных из локальных ресурсов

Локальные JSON

## Remote

Реализация Data Interface и чтение данных с внешнего сервера с помощью Refit

Refit Interface

Кешер

REST Service



# DAL DataServices

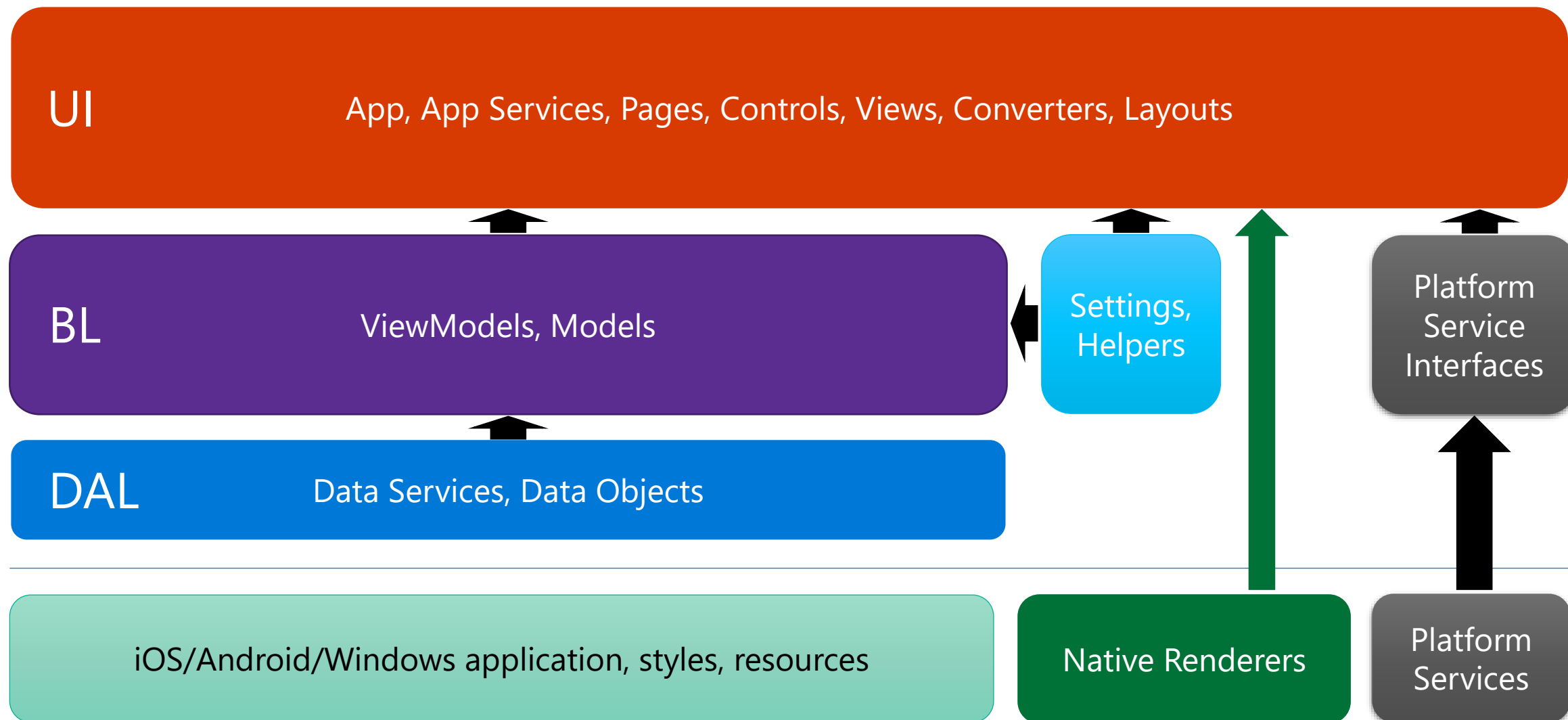
```
public static void Init(bool isMock = false) {  
    if (isMock)  
        DependencyService.Register<ISessionDataService,  
Mock.SessionDataService>();  
    else  
        DependencyService.Register<ISessionDataService,  
Remote.SessionDataService>();  
}  
  
public static ISessionDataService Session =>  
DependencyService.Get<ISessionDataService>();
```

# Приложение v0.2

Подключаем сервер и моки

# Архитектура XF-приложения

# Слои



# MVVM Business Logic (BL)

ViewModel – управление состоянием View, контекст для биндинга (свойства и команды), логика переходов между экранами.

Model – получение доп. информации из базового объекта, часто объединяется с ViewModel

# Структура PCL-проекта

DAL\

Data Objects\

Data Services\

BL\

ViewModels\

Models\

UI\

Pages\

Controls\

Converters\

PlatformServices\

App.cs

Settings.cs

Consts.cs

# Приложение v0.3

Приводим структуру проекта в порядок, подбиваем  
ХВОСТЫ

# Правильный Xamarin для мобильных приложений

1. Знакомство с XF
2. Осваиваем REST
3. Архитектура XF-приложения

Черников Вячеслав  
[slava.chernikoff@binwell.com](mailto:slava.chernikoff@binwell.com)



