



Red FabriQ

Mobile & web solutions

Xamarin Mobile Development

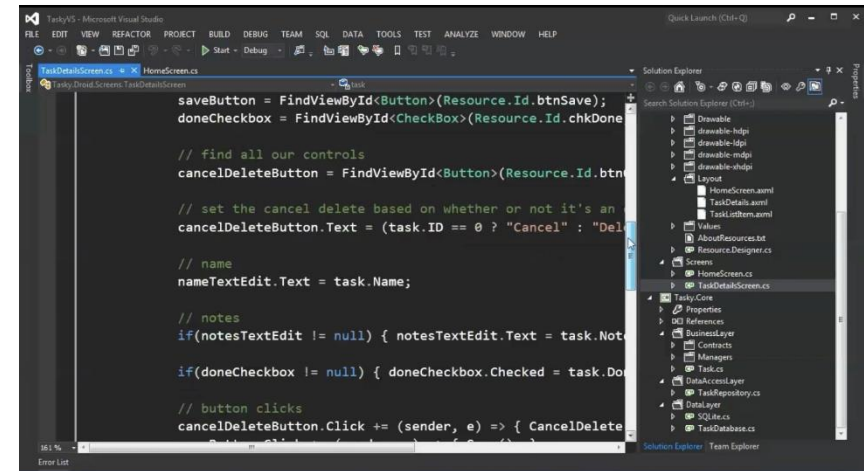
- Introducing Xamarin
- Standard Approach
- Xamarin Approach
- APIs Coverage
- Xamarin Forms
- Xamarin Tools
- Demo
- Q&A



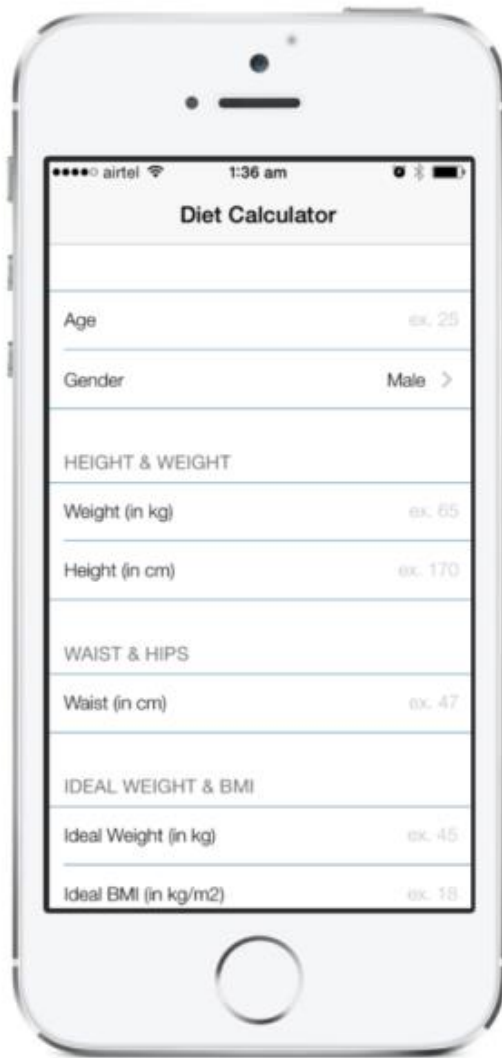
■ ■ ■ Introducing Xamarin

- Plateforme de développement mobile
- Unifie le développement des App en C# pour les plateformes majeurs
- Génère des App entièrement natives – UI & Performances
- Accélère le Time-To-Market par le biais de partage et réutilisation du code

- Les entreprises peuvent exploiter les compétences, outils, équipes C#



□ ■ ■ Native UI : No Compromises



□ ■ ■ **Standard Development Approach**



iOS App

Objective-C
XCode



Android App

Java
Eclipse



Windows App

C#
Visual Studio





iOS C# UI

Android C# UI

Windows C# UI

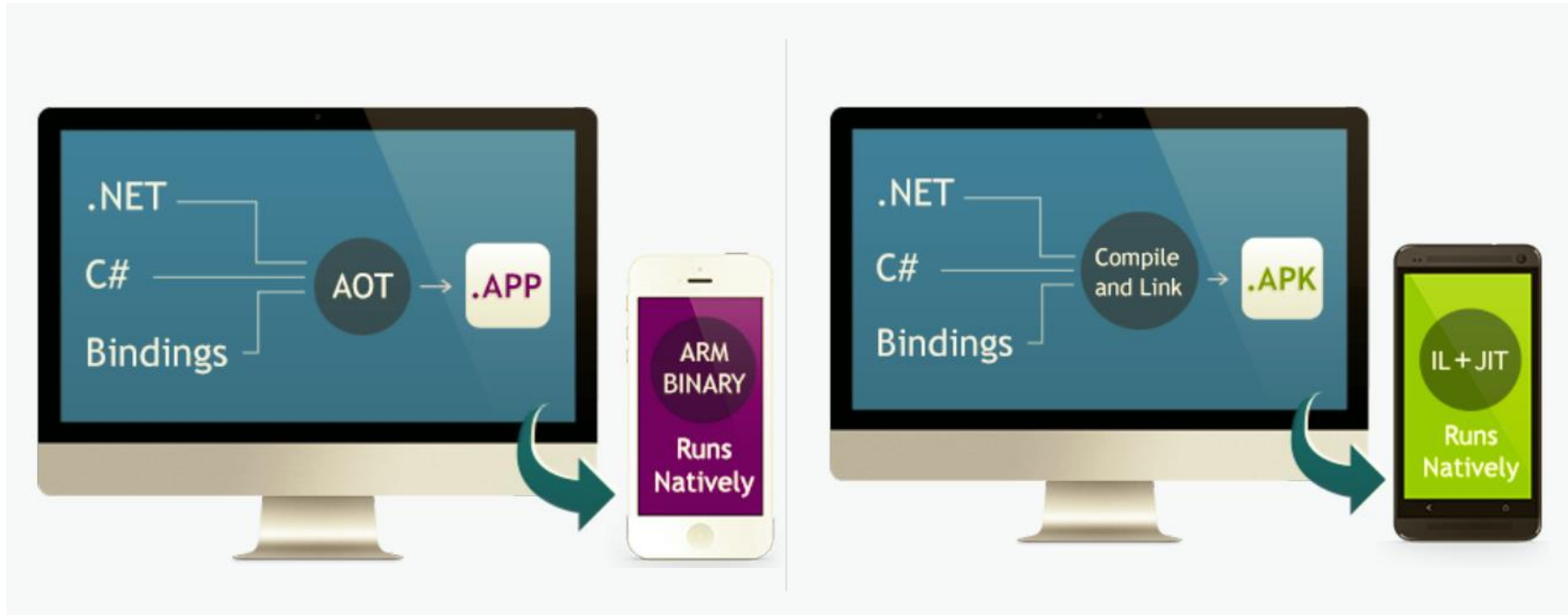
Shared C# Backend



□ ■ ■ Xamarin Approach : Code Sharing



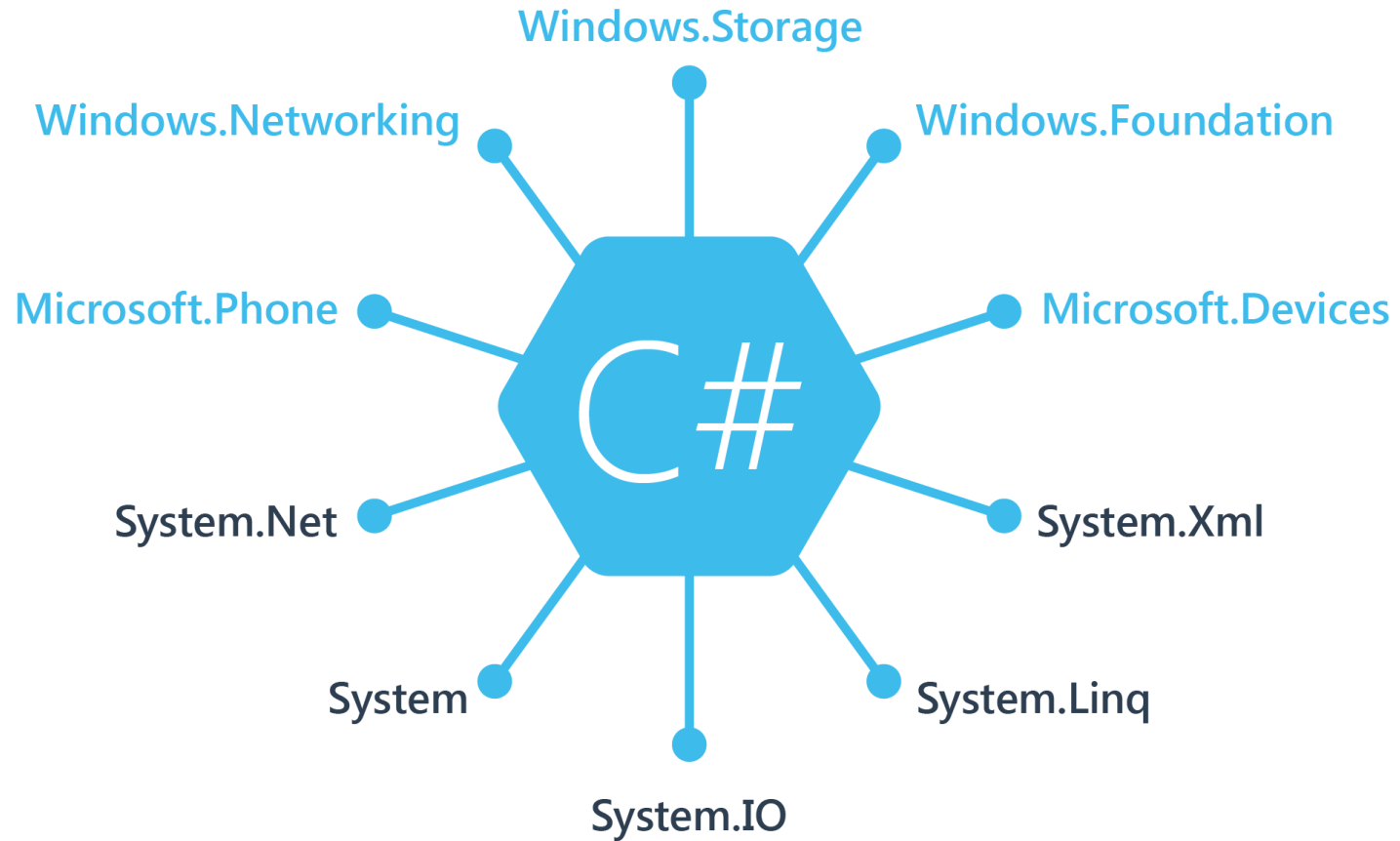
□ ■ ■ Xamarin Approach : How it works ?

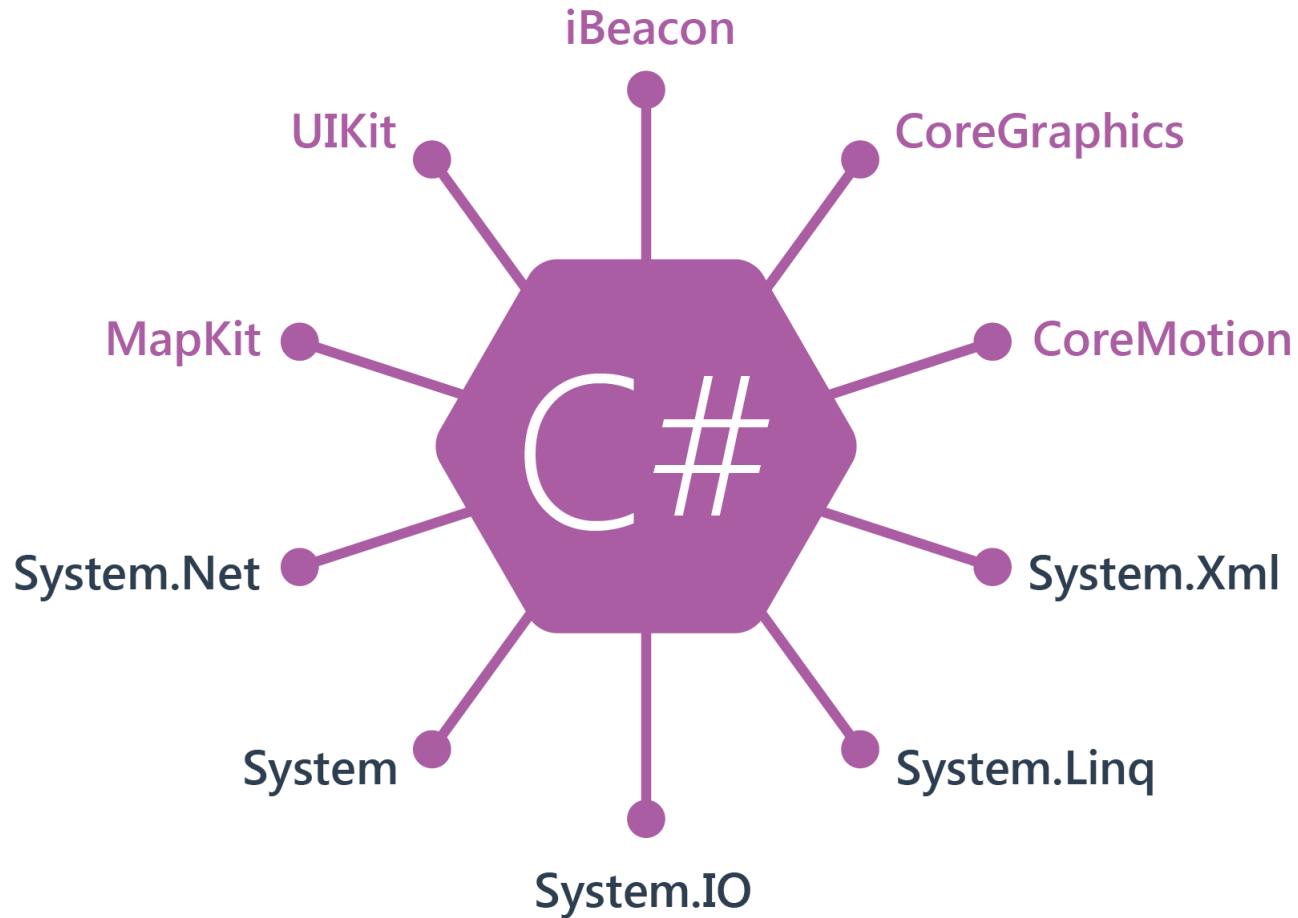


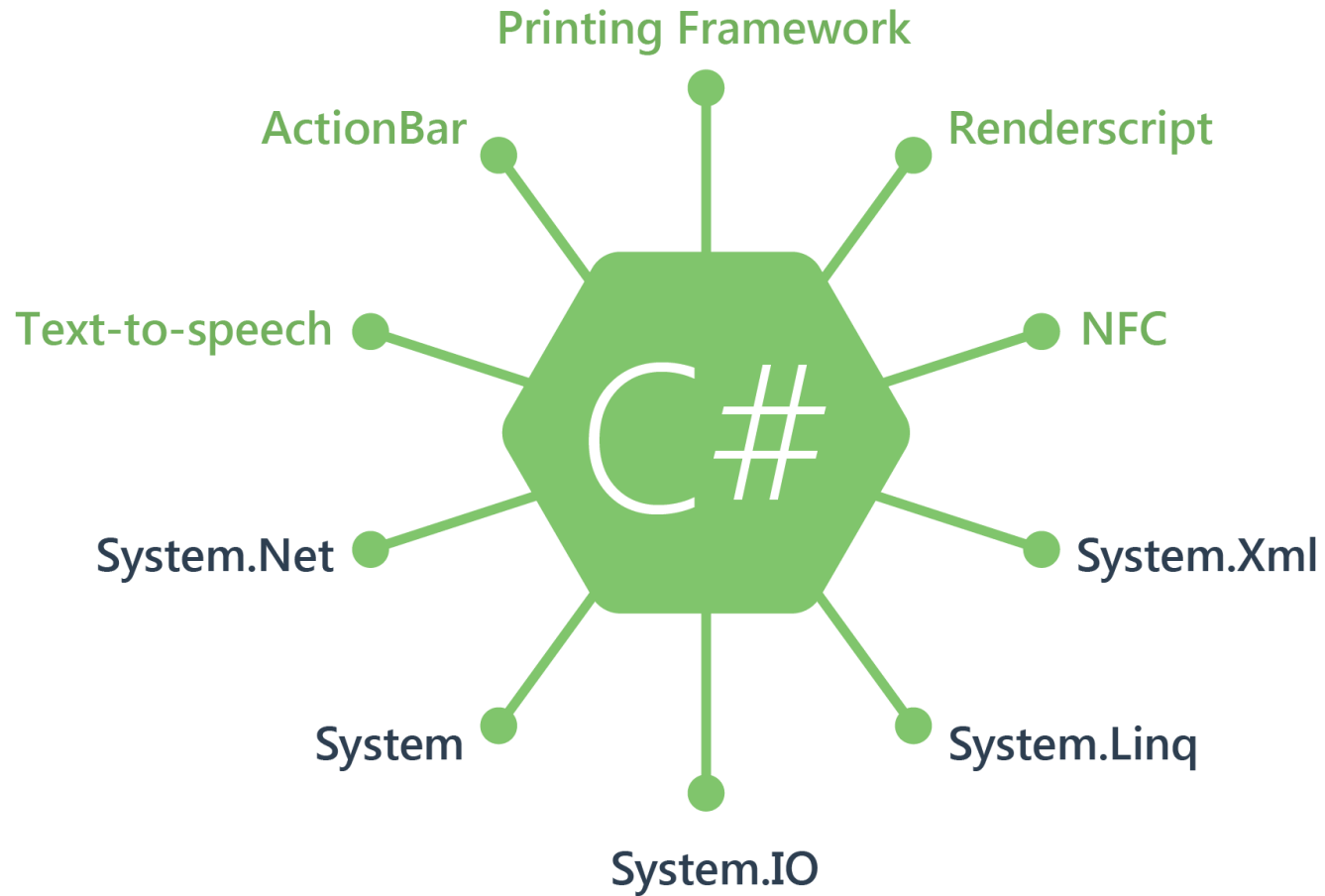
Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store

Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device



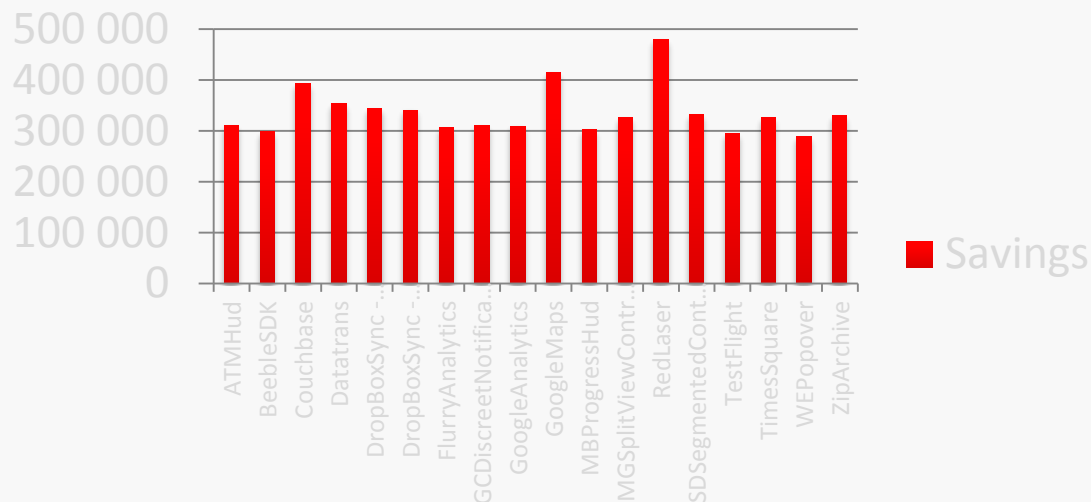






■ SmartLink

- ▶ Permet d'intégrer dans votre binaire le strict nécessaire du Framework Xamarin



■ Binding : Consommer des API natives depuis votre code C#

- ▶ Intégrer du code existant
- ▶ Obtenir des performances significatives
- ▶ Adopter des librairies tiers
- ▶ Adopter des contrôles et framework tiers



■ Sous IOS

```
login = UIButton.FromType (UIButtonType.Custom);
login.SetTitle ("Login", UIControlState.Normal);
login.SetBackgroundImage (UIImage.FromBundle ("login_btn.png").CreateResizableImage (new UIEdgeInsets (8, 8, 8, 8)),
    UIControlState.Normal);
login.TouchUpInside += delegate {
    Login ();
};
login.
AddCen TextInputMode
help = TintAdjustmentMode
help.S TintColor
help.T TintColorDidChange
int UIView.Tag Title
TitleColor
AddCen TitleEdgeInsets
//Adju titleLabel
var fr titleLabelRect
frame.X = login.Frame.Right + 8;
help.Frame = frame;

login.SetBackgroundImage (UIImage.FromBundle ("login_btn.png"), UIControlState.Normal);
login.Alert ("Please enter your name or password to login.", null, "Ok").Show ();
```

■ Sous Android

```
protected override void OnCreate (Bundle bundle)
{
    base.OnCreate (bundle);

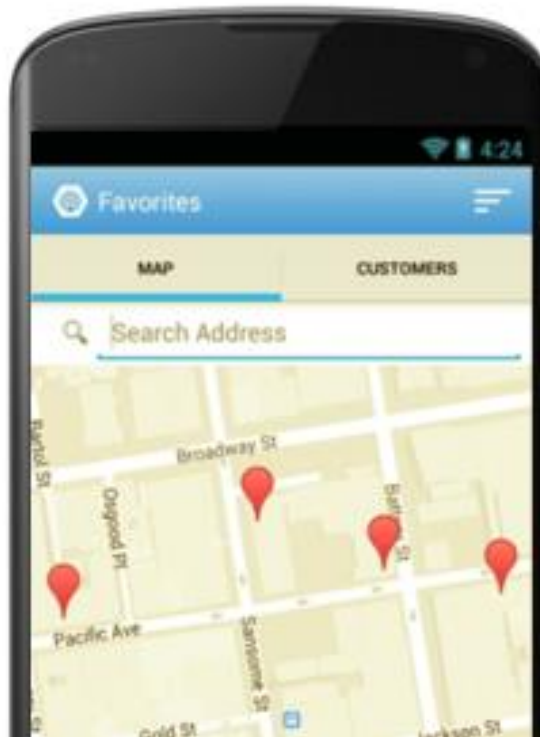
    // Set our view from the "main" layout resource
    SetContentView (Resource.Layout.Login);

    // Get our controls from the layout resource,
    // and attach an event to it
    login = FindViewById<Button> (Resource.Id.logIn);
    login.

    userNa StartDrag Id.userName);
    passwo SystemUiVisibility Id.password);
    progre SystemUiVisibilityChange Id.loginProgress);
    var lo SystemUiVisibilityChange resource.Id.loginQuestion);
    Tag
    //Set Text : & go buttons on the input keyboard to interact with login.
    userNa TextChanged
    passwo TextColors
    userNa TextFormatted
    lo TextScaleX
};
password.TextChanged += (sender, e) => {
    loginViewModel.Password = password.Text;
};
```



Construire des UIs pour iOS, Android et Windows Phone depuis un code c# unique et partagé



□ ■ ■ Xamarin Forms

- Créer rapidement et facilement des interfaces utilisateurs en partageant du code
- Les éléments Xamarin.Forms sont mappés aux contrôles et comportements natives
- Plus de code partagé : Comportement, User interaction, ViewModels, ...Etc.
- **Contrôles** : Button, BoxView, DatePicker, Editor, ListView, Image, ..etc.
- **Pages** : Content, MasterDetail, Navigation, Tabbed, Carousel
- **Layouts** : Stack, Absolute, Relative, Grid, ContentView, ScrollView, Frame
- Ecrans créés à partir du code-behind ou du Xaml
- Data Binding bidirectionnel
- Contrôle de navigation
- API d'animation
- Dependency Services
- Messagin Center





Shared UI Code

Shared C# Backend



■ ■ ■ Xamarin Forms : ViewModel Example

```
public class LoginViewModel : INotifyPropertyChanged
{
    private string username = string.Empty;
    public string Username
    {
        get { return username; }
        set { username = value; OnPropertyChanged ("Username"); }
    }

    private string password = string.Empty;
    public string Password
    {
        get { return password; }
        set { password = value; OnPropertyChanged ("Password"); }
    }

    public Command LoginCommand
    {
        get {
            return new Command (() => {
                //Log into Server here
            });
        }
    }
}
```



■ ■ ■ Xamarin Forms : Code Behind Example

```
public class LoginPage : ContentPage
{
    public LoginPage()
    {
        //set binding context
        this.BindingContext = new LoginViewModel ();

        //create UI & bind to properties
        var username = new Entry { Placeholder = "Username" };
        username.SetBinding (Entry.TextProperty, "Username");

        var password = new Entry { Placeholder = "Password", IsPassword = true };
        password.SetBinding (Entry.TextProperty, "Password");

        var loginButton = new Button {
            Text = "Login",
            TextColor = Color.White,
            BackgroundColor = Color.FromHex("77D065")
        };

        loginButton.SetBinding (Button.CommandProperty, "LoginCommand");

        //set main content of page
        Content = new StackLayout{
            VerticalOptions = LayoutOptions.Center,
            Padding = 50, Spacing = 10,
            Children = { username, password, loginButton }
        };
    }
}
```



■ ■ ■ Xamarin Forms : XAML Example

```
<?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="LoginExampleForms.LoginPageXAML">
  <ContentPage.Content>

    <StackLayout VerticalOptions="Center" Padding="50" Spacing="10">

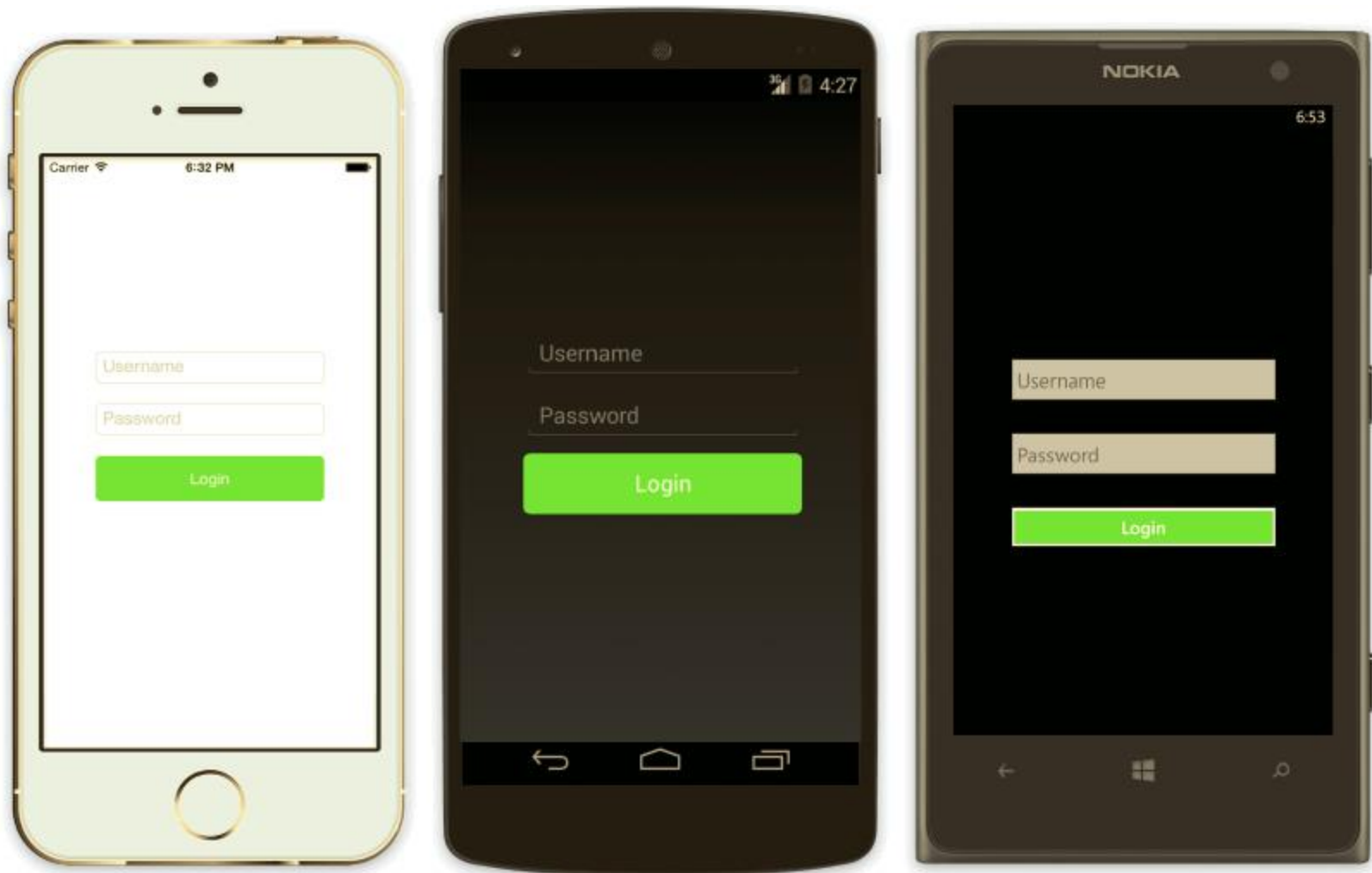
      <Entry Placeholder="Username" Text="{Binding Username}"/>
      <Entry Placeholder="Password" Text="{Binding Password}"/>
      <Button Text="Login"
        TextColor="#FFFFFF"
        BackgroundColor="#77D065"
        Command="{Binding LoginCommand}"/>

    </StackLayout>

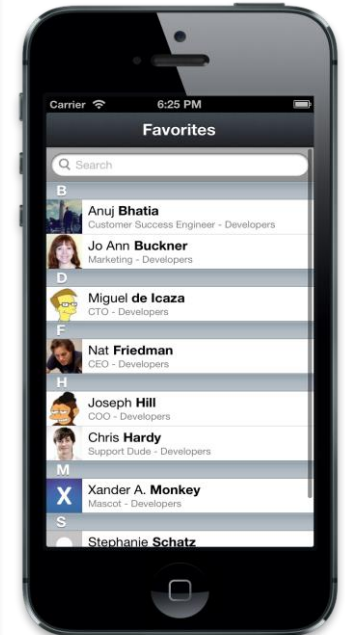
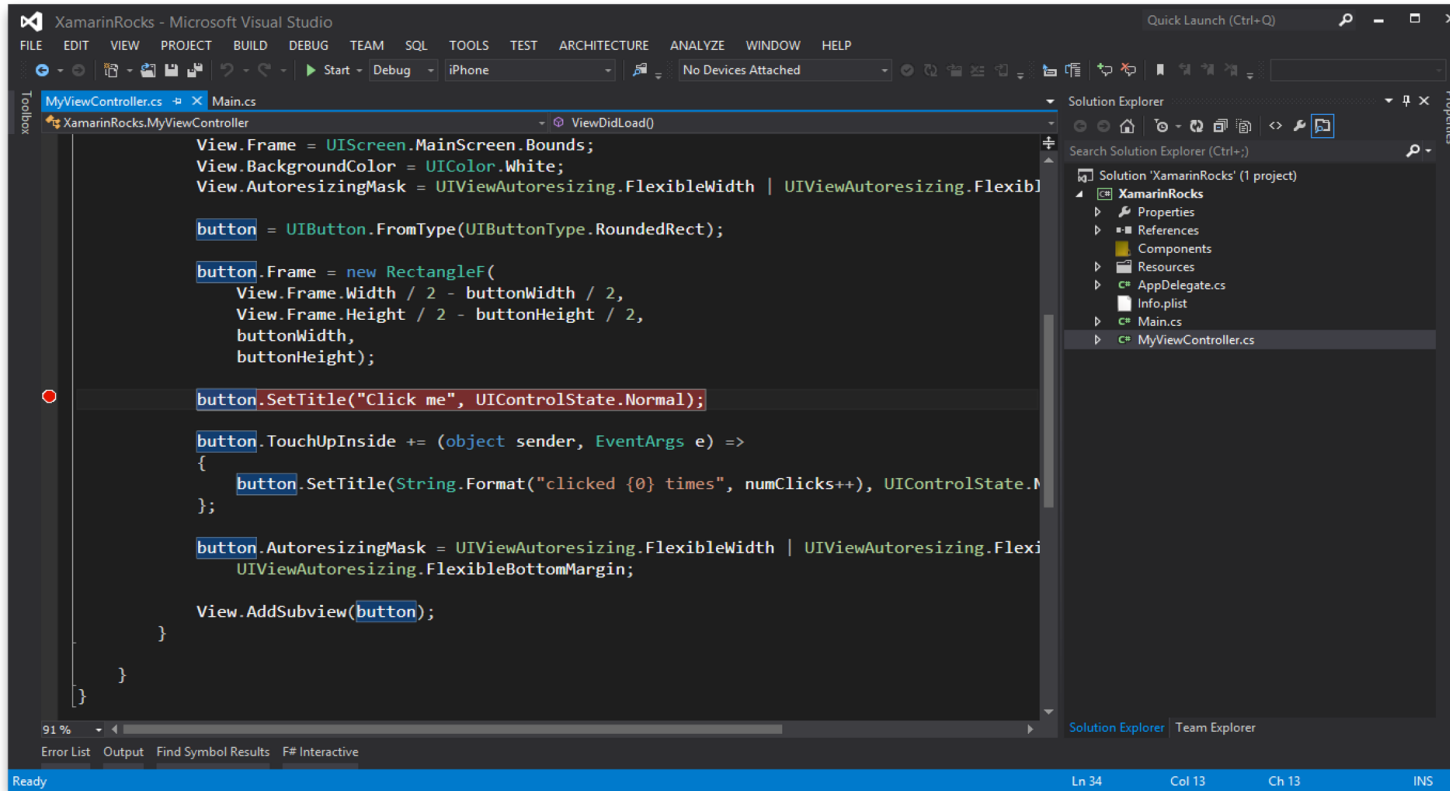
  </ContentPage.Content>
</ContentPage>
```



■ ■ ■ Xamarin Forms : Rendering Example



Xamarin Tools : Visual Studio Extension



Une seule solution pour iOS, Android et Windows

C# and F# support

Reuse your Visual Studio and .Net Experience

Préserver votre écosystème Microsoft :

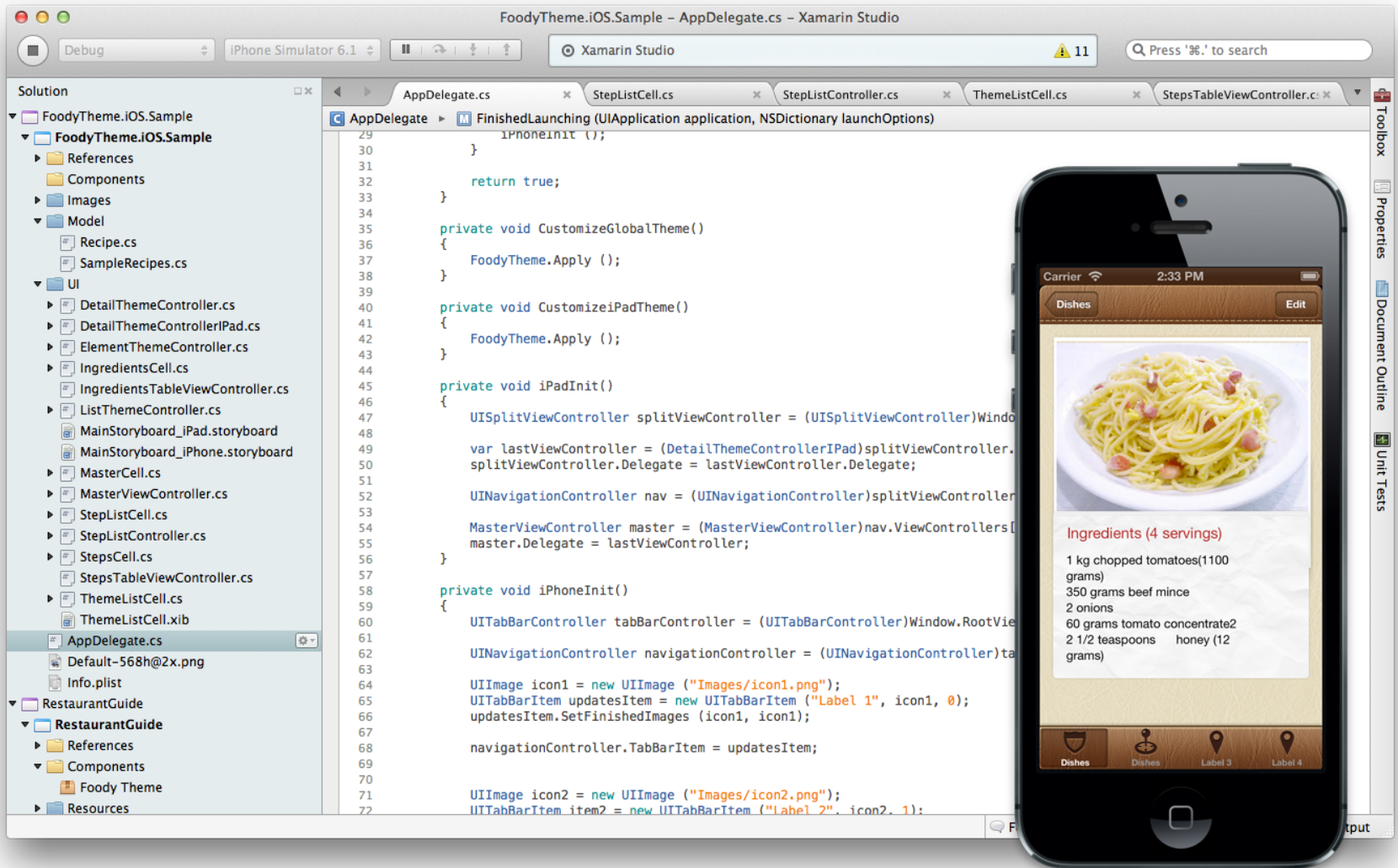
ReSharper

Team Foundation Server (TFS)

Your favorite code coverage and profiling tools

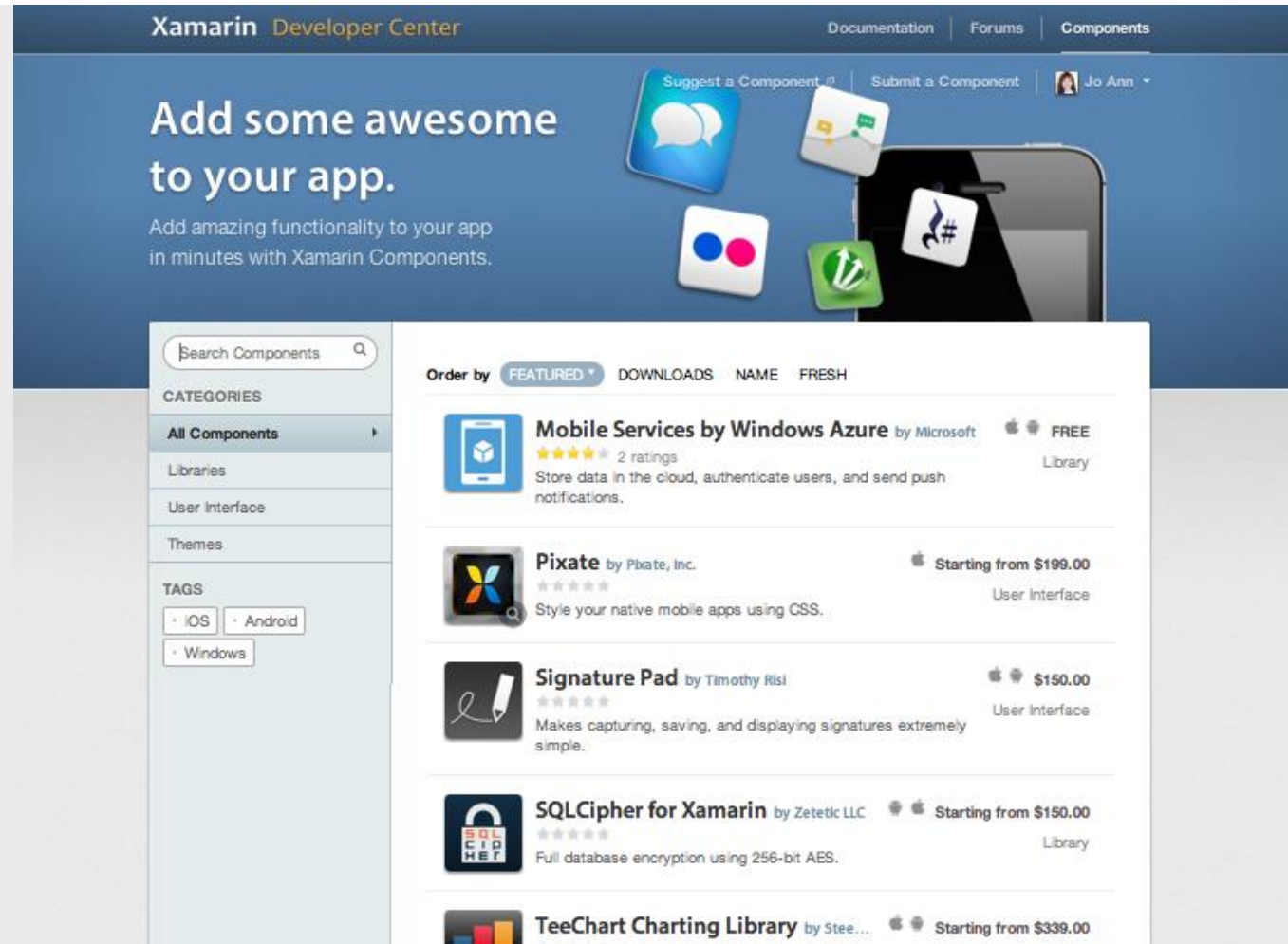


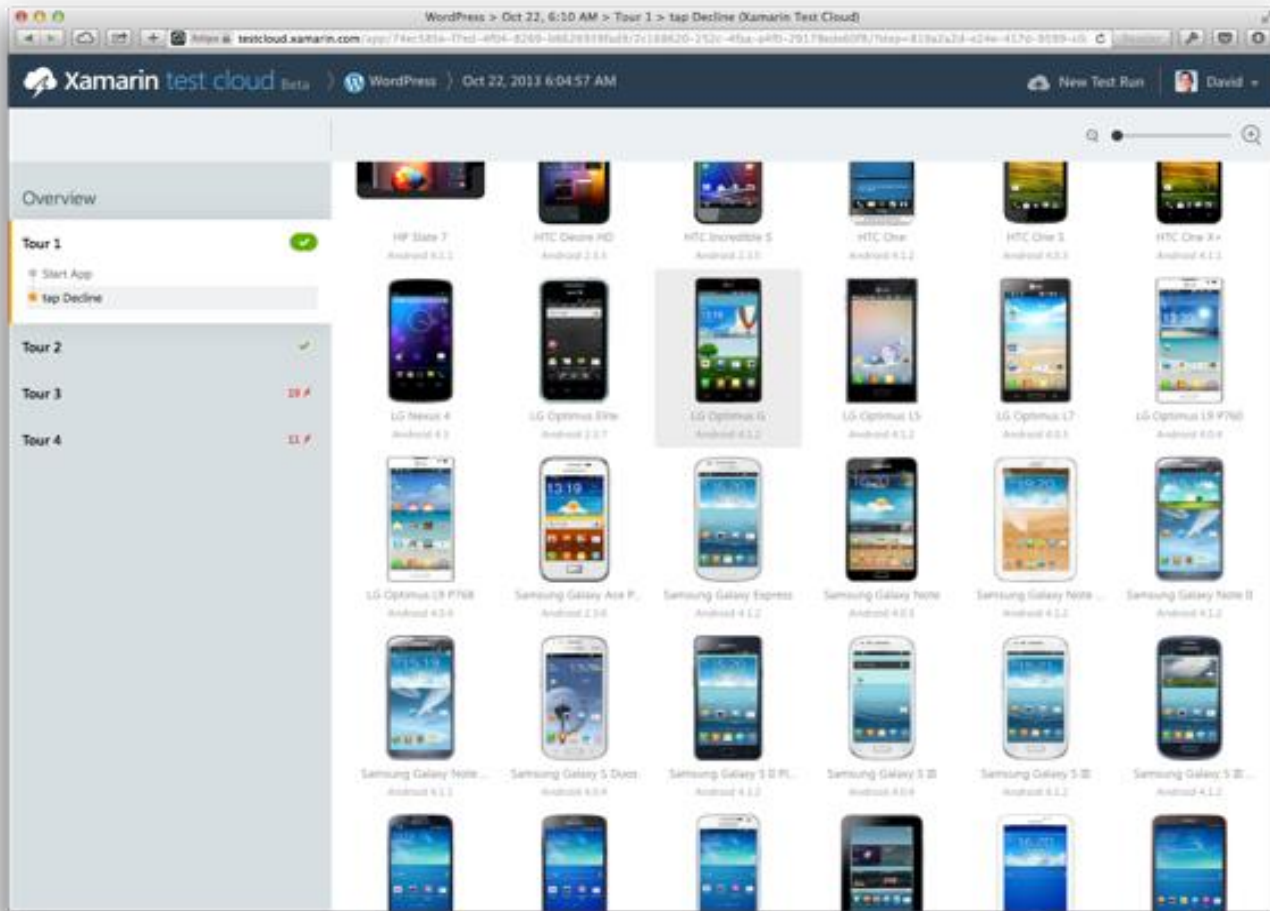
Xamarin Tools : Xamarin Studio



Pre-Build de
composants, Contrôles
UI, Themes & Web
Services de haute
qualité.

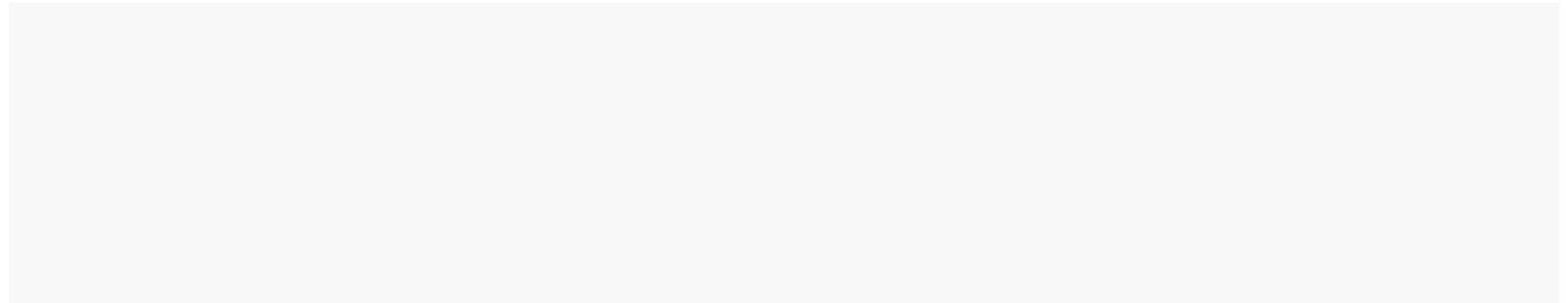
Full IDE Integration.





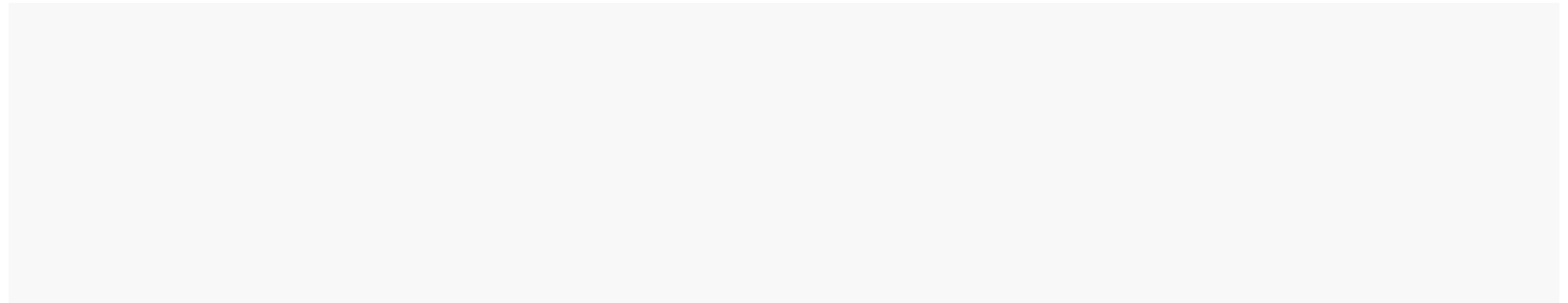
- *Test de votre application et plus particulièrement des IHM*
 - *Exécution très rapide de vos tests*
- *Exécution de vos tests sur de vrais matériels*
 - *Résistant aux charges*
 - *Interface accessible et simple d'usage*





DEMO





Q&A

