



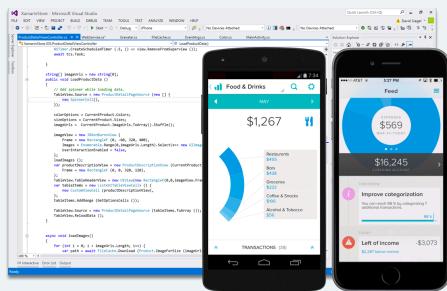
Développez des
applications natives
pour iOS, Android, Mac
et Windows en C#

Julien Mialon

Xamarin Student Ambassador

mialon.julien@gmail.com

Xamarin est un moyen de



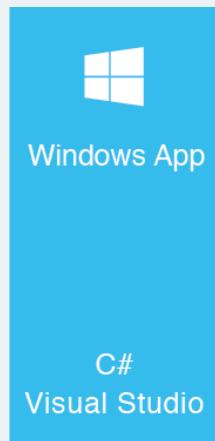
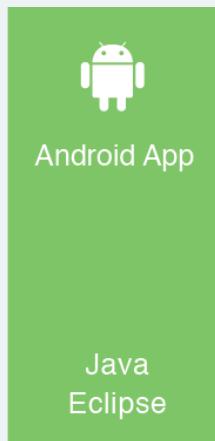
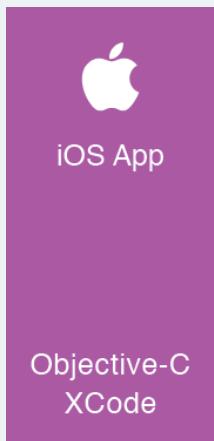
Créer

Tester

Monitorer

Différentes approches pour le Développement Mobile

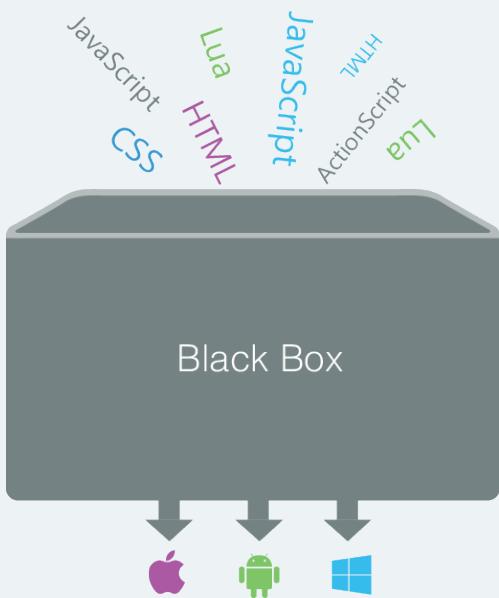
L'approche native



Créer l'application plusieurs fois

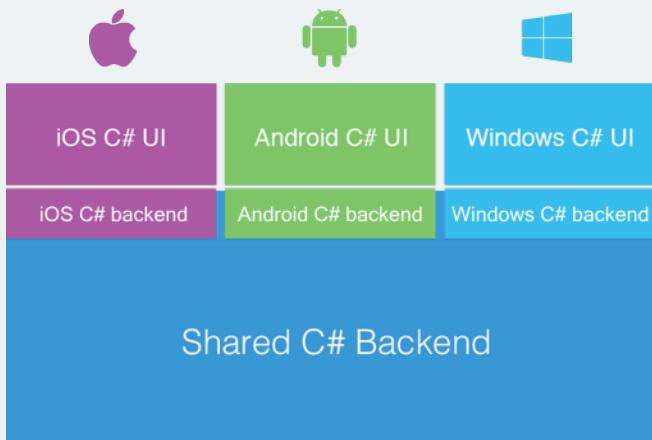
- Plus de développeurs
- Plus de maintenance
- Différents environnements

L'approche Write Once, Run Anywhere



- Limitation au dénominateur commun
- Fragmentation des web browsers
- Développement & design pour 1 plateforme, fonctionne aussi sur les autres

L'approche de Xamarin



- UI native
- Performance native
- Code partagé
- C# & .NET Framework
- Accès à toutes les APIs de chaque plateforme

Pourquoi C# ?

C# is Awesome

```
from p in Table<Person> ()  
    where p.ID == id  
    select p;
```

- LINQ Support

```
var doc = XDocument.Load(url);  
foreach(var item in doc.Root.Elements()) {  
    var text = item.Value;  
}
```

- XML support

```
button.TouchUpInside += (s, o) => {  
    message.Text = "Hello!";  
};
```

- Event Handling & Delegates

C# is Awesome - JSON

Json.NET permet de simplifier la conversion entre vos données JSON et vos objets .NET.

```
public class Person
{
    public string Name { get; set; }
    public DateTime Birthday { get; set; }
}
var person = new Person { Name = "Bob", Birthday = new DateTime (1987, 2, 2) };
var output = Newtonsoft.Json.JsonConvert.SerializeObject (person);

person = Newtonsoft.Json.JsonConvert.DeserializeObject<Person> (output);
Console.WriteLine ("{0} - {1}", person.Name, person.Birthday);
```

La différence – Classes et Méthodes

```
// Objective-C
@interface Person : NSObject
@property (strong, nonatomic) NSString *name;
@end

@implementation Person
- (id)initWithName:(NSString *)name {
    self = [super init];
    if (self) {
        self.name = name;
    }
    return self;
}

+ (NSArray *)getNames {
    NSArray *people = @[
        [[Person alloc] initWithName:@"David"],
        [[Person alloc] initWithName:@"Vinicius"],
        [[Person alloc] initWithName:@"Serena"],
    ];
    NSMutableArray *names = [NSMutableArray array];
    [people enumerateObjectsUsingBlock:^(Person *person,
                                         NSUInteger idx,
                                         BOOL *stop) {
        [names addObject:person.name];
    }];
    return names;
}
@end
```

C# avec Xamarin

```
// C# with Xamarin
class Person : NSObject {
    public string Name { get; set; }

    public static string[] GetNames() {
        var people = new[] {
            new Person { Name="David" },
            new Person { Name="Vinicius" },
            new Person { Name="Serena" },
        };
        return people.Select(person => person.Name).ToArray();
    }
}
```

Simplification des API avec Async/Await

Objective-C

```
[UIView animateWithDuration:0.2  
    animations:^{view.alpha = 0.0;}  
    completion:^(BOOL finished){ [view removeFromSuperview]; }];
```

C# avec Xamarin

```
//Animate alpha to 0 asynchronously.  
//await animation and then remove from superview  
bool success = await UIView.AnimateAsync(2, () => { UIView.Alpha = 0;});  
view.RemoveFromSuperview();
```

La différence – Android ItemClick

Java

```
listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // Value of item
        String itemValue = (String) listView.getItemAtPosition(position);

        // Show Toast
        Toast.makeText(getApplicationContext(),"Position :" + position + " ListItem : "
                + itemValue , Toast.LENGTH_LONG).show();
    }
});
```

C# avec Xamarin

```
listView.ItemClick += (sender, args) => {
    // Value of item
    var itemValue = (string)listView.GetItemAtPosition(args.Position);

    //Show Toast
    Toast.MakeText(this, string.Format("Postition: {0} ListItem: {1}",
        args.Position, itemValue), ToastLength.Long).Show();
};
```

C# & Async avec Xamarin

```
listView.ItemClick += async (sender, args) => {
    // Value of item
    var itemValue = (string)listView.GetItemAtPosition(args.Position);

    //Show Toast
    Toast.MakeText(this, string.Format("Postition: {0} ListItem: {1}",
        args.Position, itemValue), ToastLength.Long).Show();
};
```

Async/Await

```
public async Task ExecuteGetPodcastsCommand()
{
    if (IsBusy)
        return;

    try
    {
        IsBusy = true;
        var client = new HttpClient();
        // Request from server podcast xml
        var podcastString = await client.GetStringAsync(PodcastUrl);

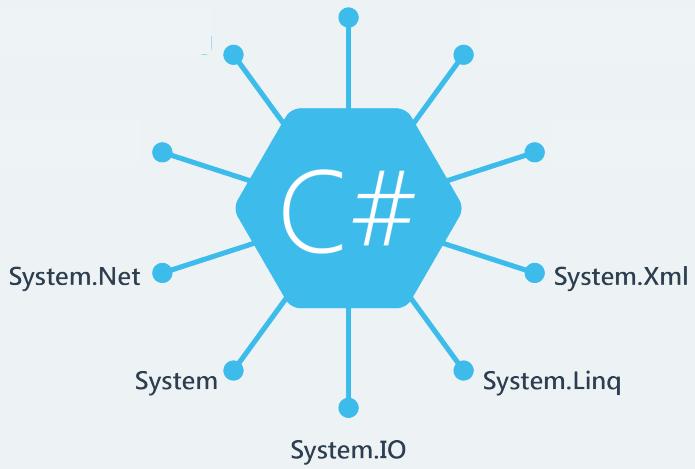
        // Parse xml into data model and load into list
        var casts = await ParseXml(podcastString);

        foreach (var cast in casts)
        {
            Podcasts.Add(cast);
            FilteredPodcasts.Add(cast);
        }
    }
}
```

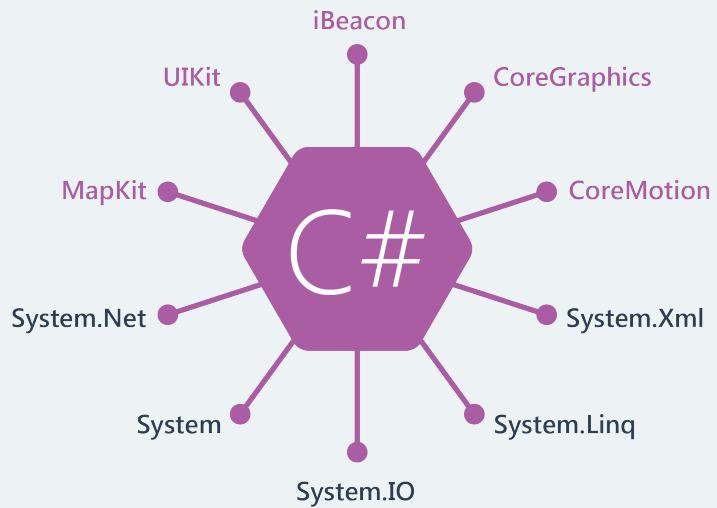
Écrivez du code simple & maintenable

Comment Xamarin Fonctionne

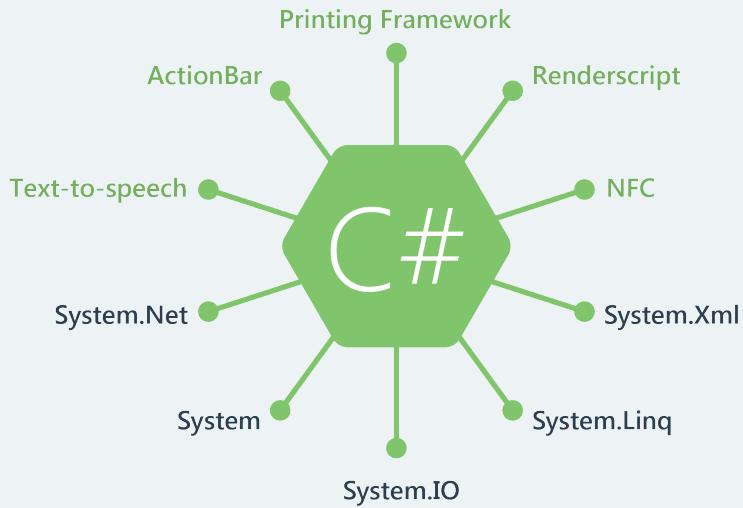
.NET + Windows APIs



.NET + iOS APIs | 100% Coverage



.NET Android APIs | 100% Coverage



Tout ce que vous pouvez faire en Objective-C, Swift ou Java peut être fait en C# avec Xamarin et Visual Studio.

Performance native



Xamarin.iOS compile votre application en binaire ARM pour l'App Store d'Apple.



Xamarin.Android utilise un système Just In Time lors de l'exécution.

Toujours à jour



Support le jour de la sortie :
iOS 5, iOS 6, iOS 7, iOS 7.1, iOS 8



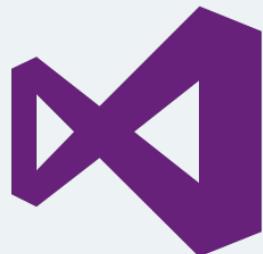
Support complet :

- Google Glass
- Android Wear
- Amazon Fire TV
- Et plus !

Environnement de développement

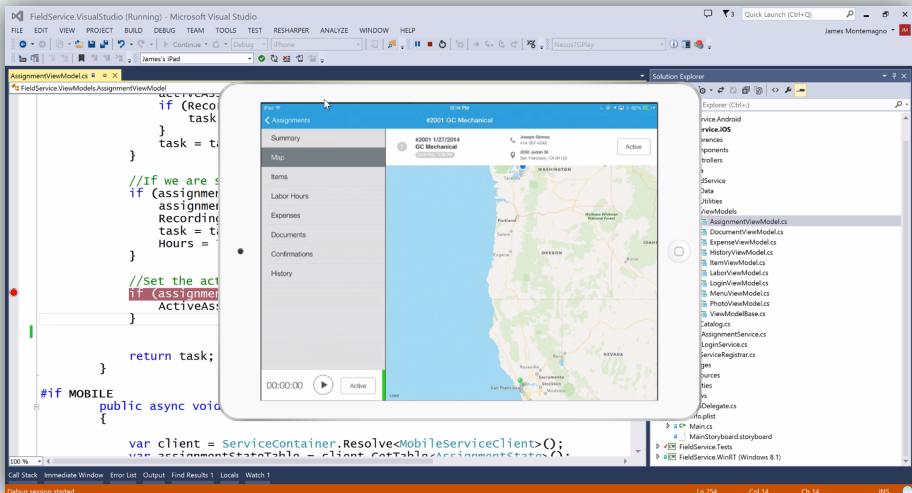


Xamarin Studio
PC ou Mac



Visual Studio Plugin
VS 2010 et Plus

Intégration dans Visual Studio



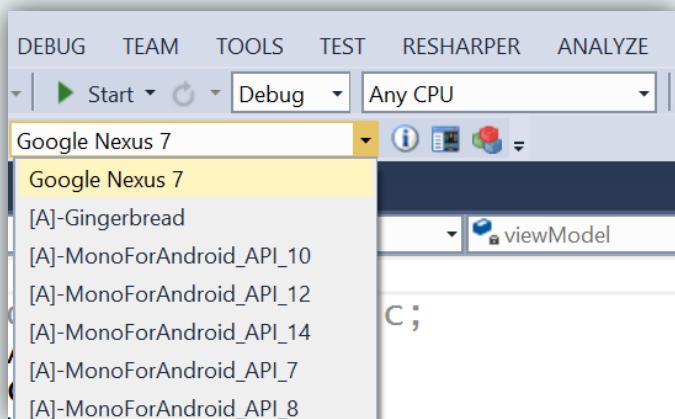
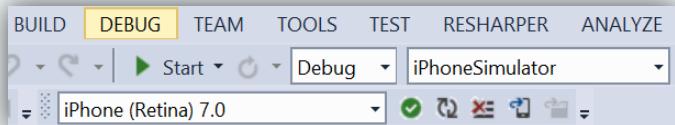
Une seule solution :

- iOS
- Android
- Windows Phone
- Windows Store

Utilisez
l'écosystème
Microsoft :

- ReSharper
- Team Foundation Server
- Tous vos outils préférés

Intégration dans Visual Studio

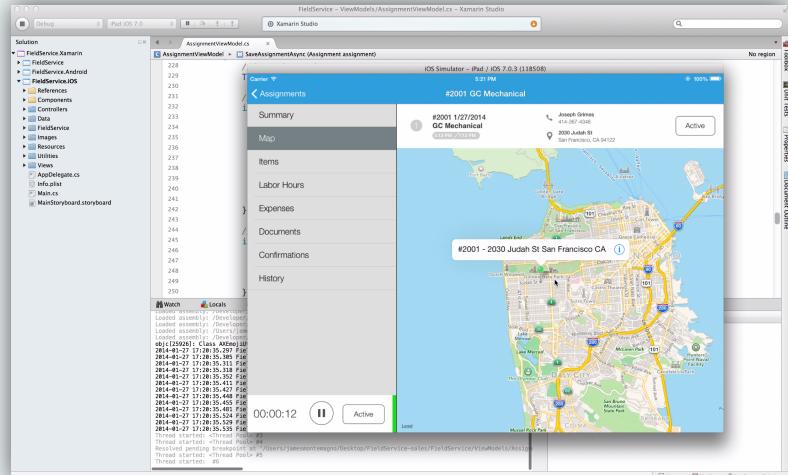


Déboguez sur :

- Émulateurs
- Devices

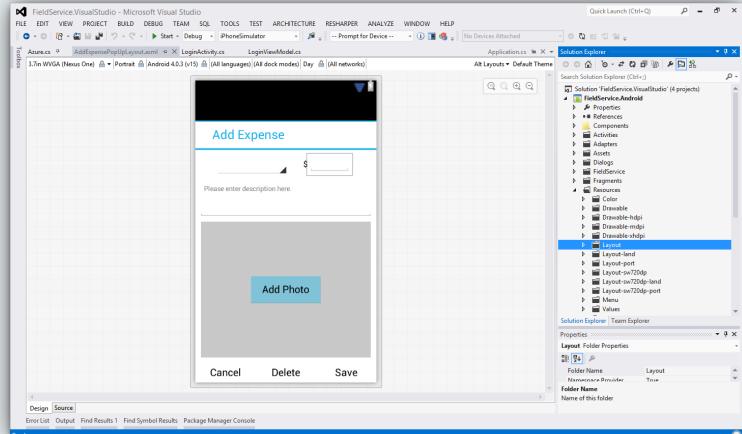
Xamarin Studio

- Optimisé pour le développement mobile cross-platform
- Explorez les APIs avec l'auto complétion
- Designers pour Android et iOS
- Débogage puissant sur émulateur ou device

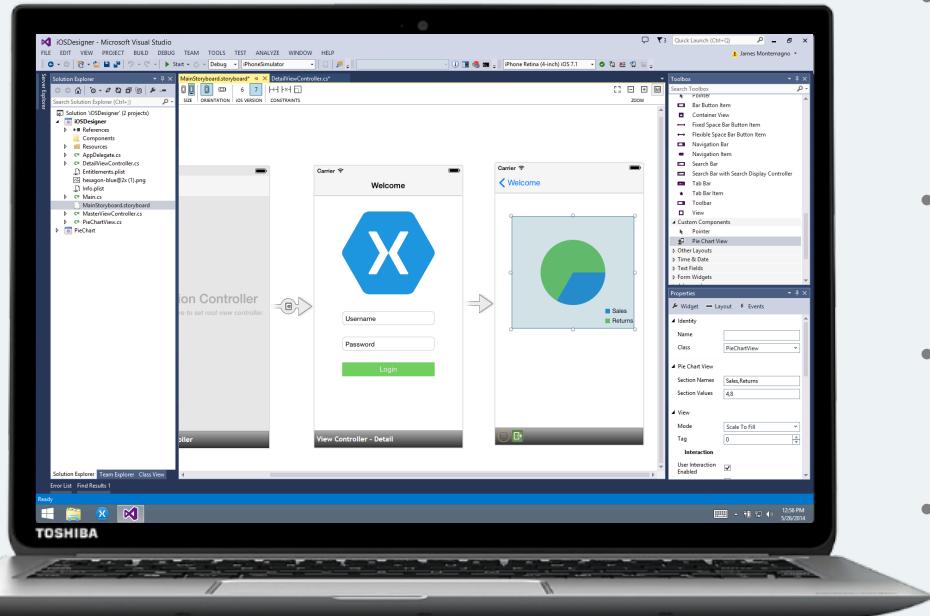


Designer Android

- Disponible dans :
 - Xamarin Studio
 - Visual Studio
- Créez votre UI avec un simple drag & drop
- Ciblez des écrans, des résolutions et des versions d'Android différentes
- Les layouts sont au format XML standard d'Android



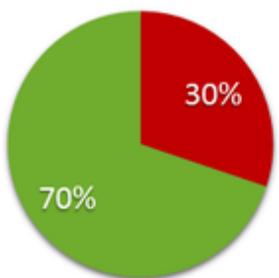
Xamarin Designer pour iOS



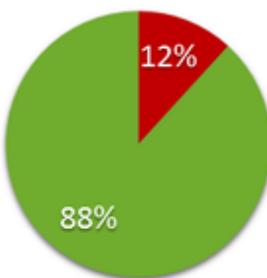
- Disponible dans Xamarin Studio et Visual Studio
- Support de tous les éléments UIKit
- Support de composants tiers
- Aperçu des changements en temps réel

Partage de code : Accélérer le développement

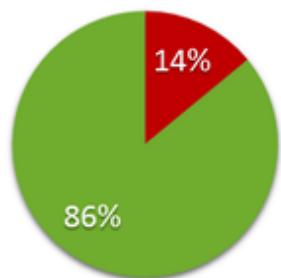
iOS



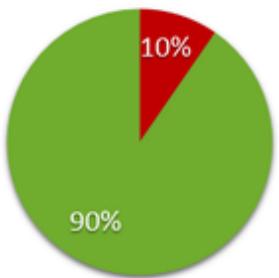
OS X



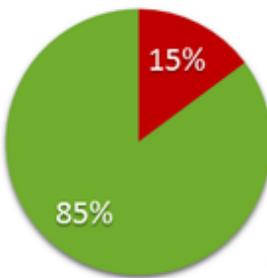
Android



Windows Phone 7



Windows RT



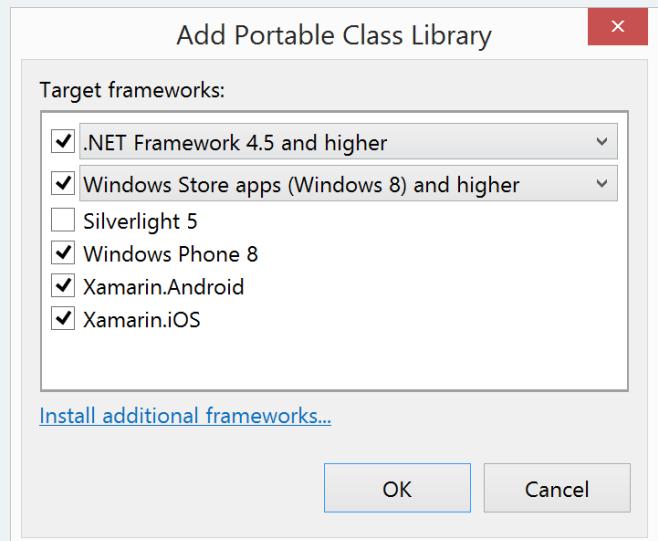
iCircuit Code Reuse
February 5, 2013

Copyright 2013 Krueger Systems, Inc.

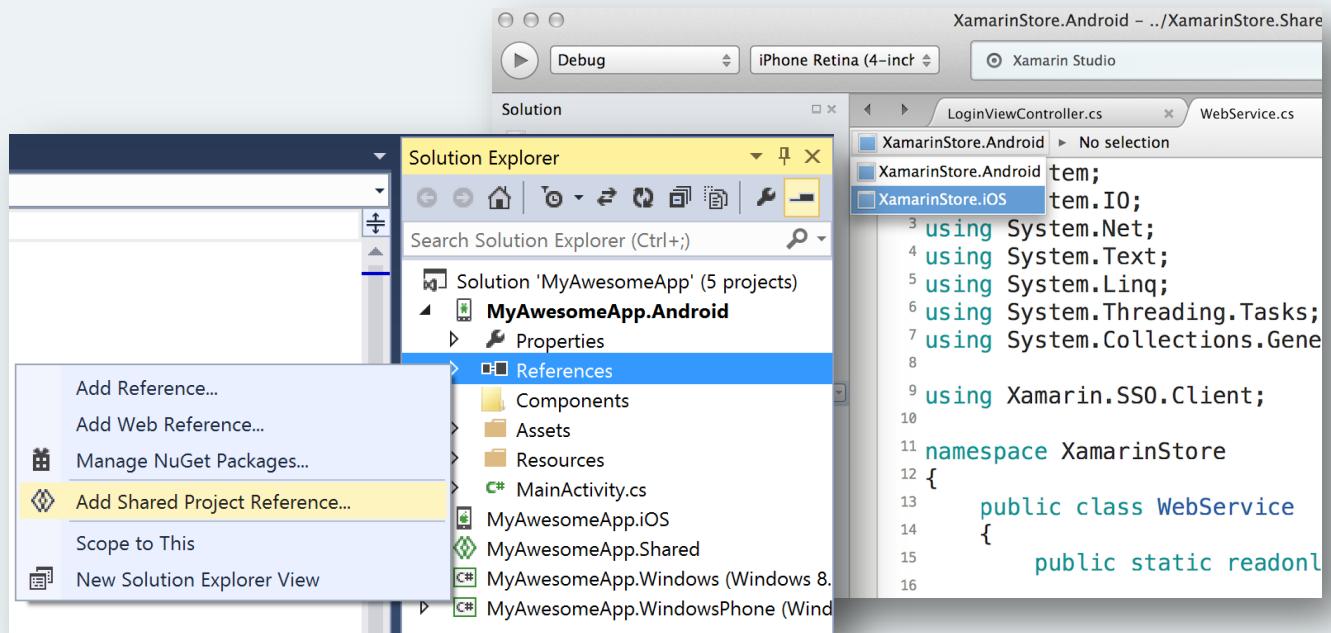
Source: <http://praeclarum.org/post/42378027611/icircuit-code-reuse-part-cinq>

Portable Class Libraries

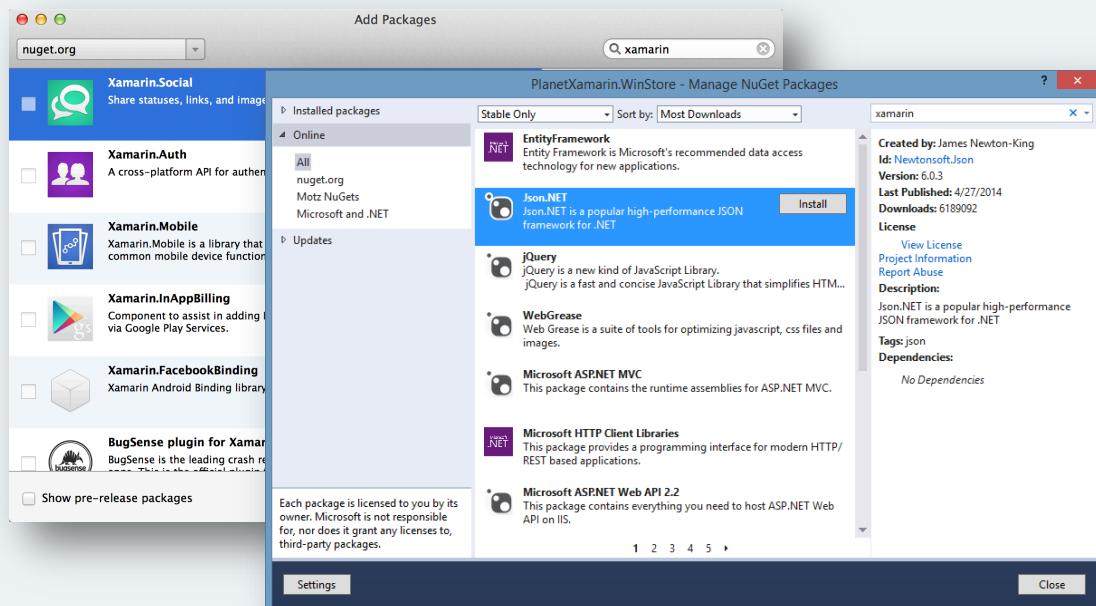
- 1 Assembly
- Plusieurs plateformes
- Incluant :
 - **Xamarin.Android**
 - **Xamarin.iOS**



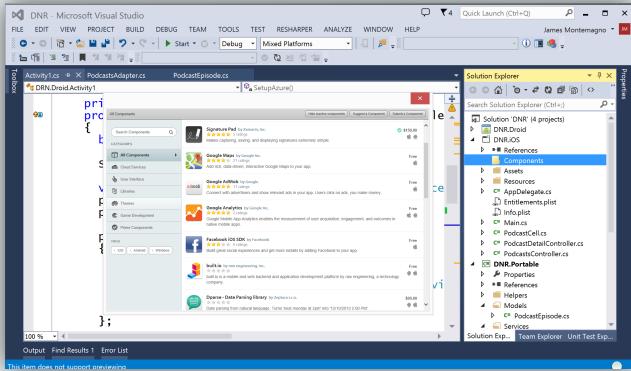
Shared Projects



NuGet

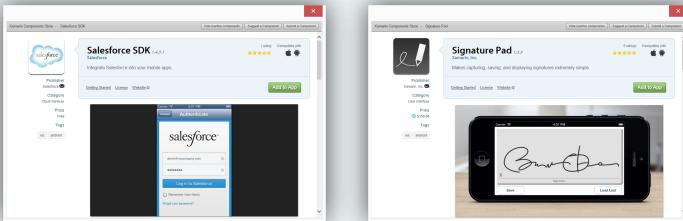


Xamarin Component Store



Créez plus vite

- Ajoutez des composants existants depuis Visual Studio ou Xamarin Studio



- Des éléments d'UI, d'intégration avec un service cloud et d'autres sont disponibles

Distribuable partout

Une application Xamarin peut être distribuée partout



Pre-built T-Shirt Store

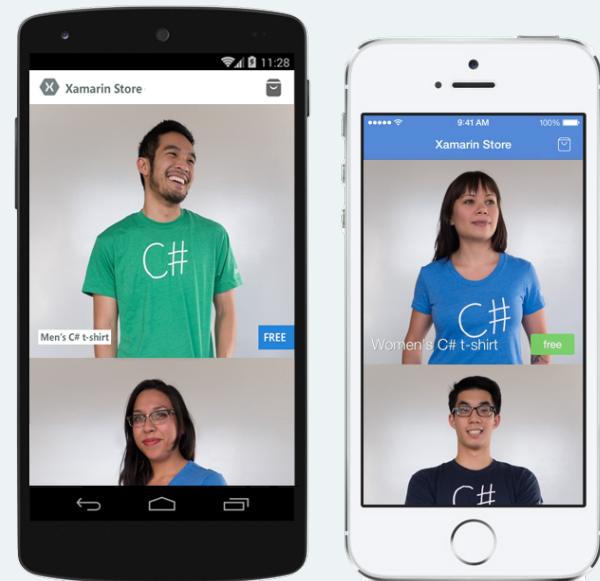
Features:

- Shopping Cart
- Social Share
- Gravatar Integration
- Beautiful Animations

Technical Details

- Code sharing with Shared Project
- Android Fragments
- Local image caching
- Cross platform image downloading
- Modular/self-contained screens

<http://xamarin.com/prebuilt/sharp-shirt>



Pre-built Employee Directory

Features:

- LDAP-ready for easy integration into your corporate directory
- Multiple search options
- Ability to favorite contacts
- Ability to call or email from contact's listing
- Gravatar Integration

Technical Details

- iPhone, Android, and Windows native user interfaces
- Uses MVVM, shared view-models across platforms
- SQLite data store with a .NET-based SQLite ORM on all platforms
- Ability to hook up your enterprise backend



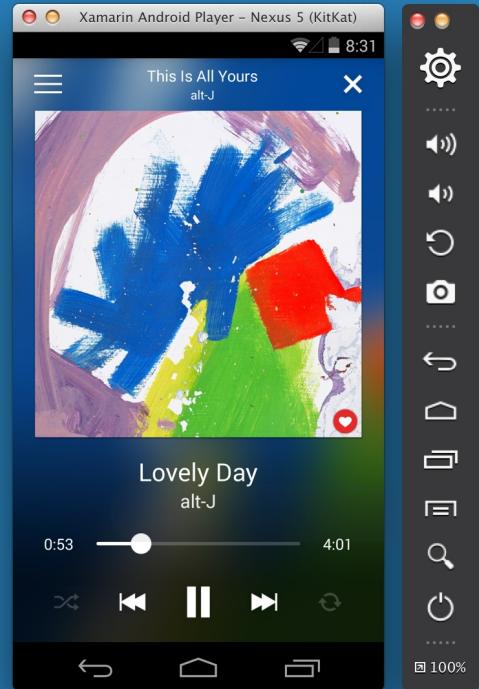
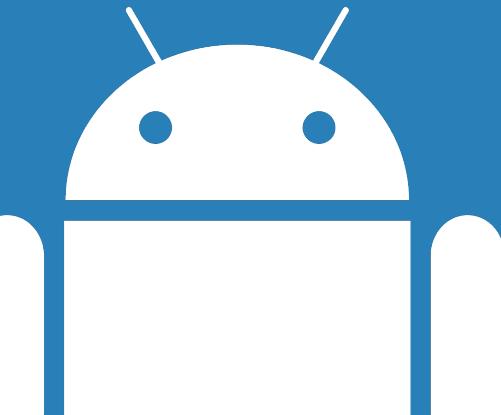
<http://xamarin.com/prebuilt/employee-directory>

Xamarin Android Player

High Speed Android Emulator

- Mac ou PC

- Xamarin.com/Android-Player

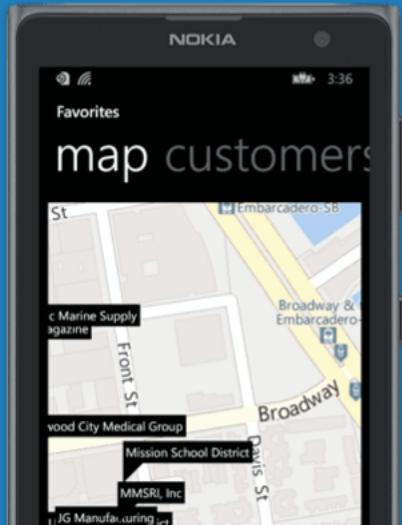
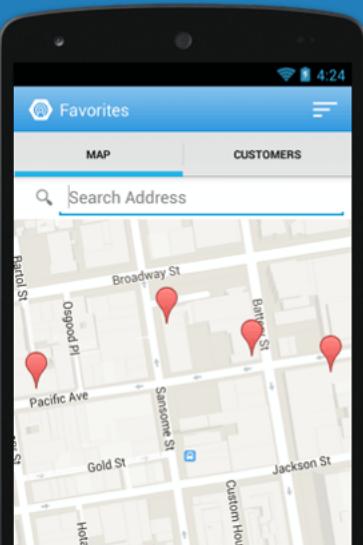
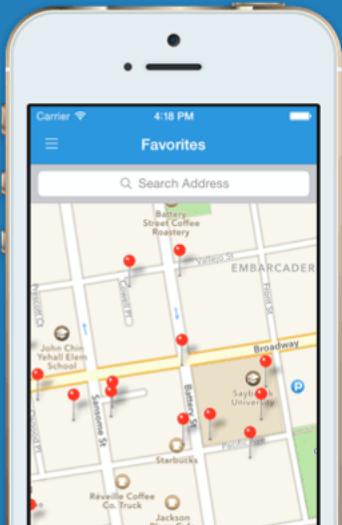


Démo

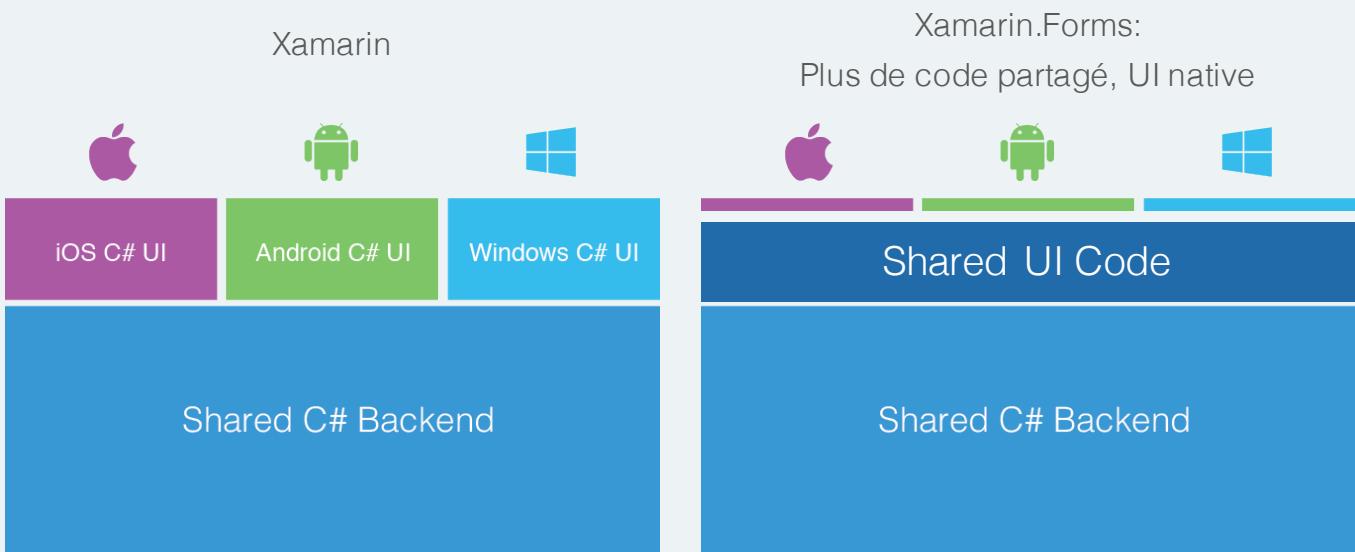
#OrleansTechTalks

Meet Xamarin.Forms

Build native UIs for iOS, Android and Windows Phone
from a single, shared C# codebase.

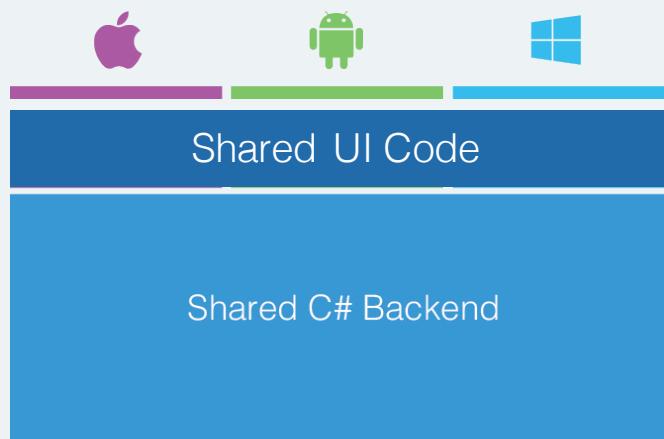


L'approche de Xamarin.Forms



De quoi dispose-t-on ?

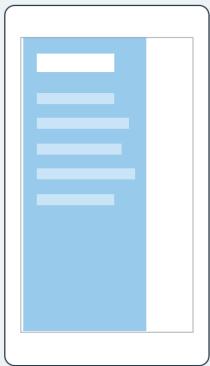
- +40 Pages, Layouts, et Controls
 - Utilisable en C# ou Xaml
- Two-way Data Binding
- Navigation
- Animation API
- Dependency Service
- Messaging Center



Pages



Content



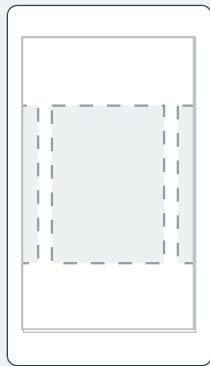
MasterDetail



Navigation

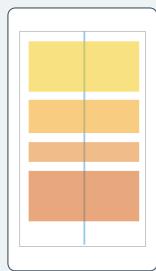


Tabbed

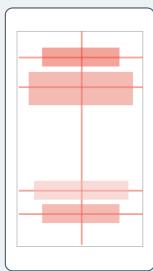


Carousel

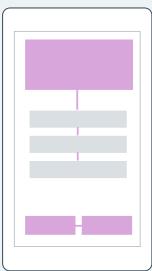
Layouts



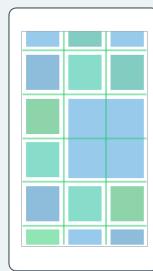
Stack



Absolute



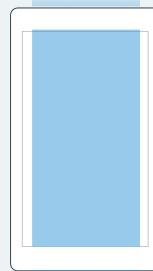
Relative



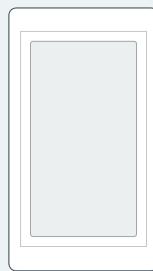
Grid



Content View



ScrollView



Frame

Controls

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	

Exemple Xamarin.Forms

```
using Xamarin.Forms;

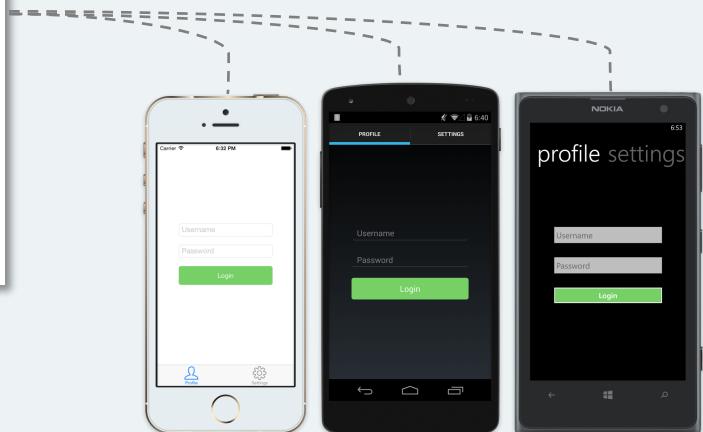
var profilePage = new ContentPage {
    Title = "Profile",
    Icon = "Profile.png",
    Content = new StackLayout {
        Spacing = 20, Padding = 50,
        VerticalOptions = LayoutOptions.Center,
        Children = {
            new Entry { Placeholder = "Username" },
            new Entry { Placeholder = "Password", IsPassword = true },
            new Button {
                Text = "Login",
                TextColor = Color.White,
                BackgroundColor = Color.FromHex("770065") }}}
};

var settingsPage = new ContentPage {
    Title = "Settings",
    Icon = "Settings.png",
    (...)}
};

var MainPage = new TabbedPage { Children = { profilePage, settingsPage } };
```

Utilisez une seule API pour générer une interface native pour chaque plateforme

À l'exécution, les pages et contrôles Xamarin.Forms sont transformés en composants natifs.



Démo

#OrleansTechTalks

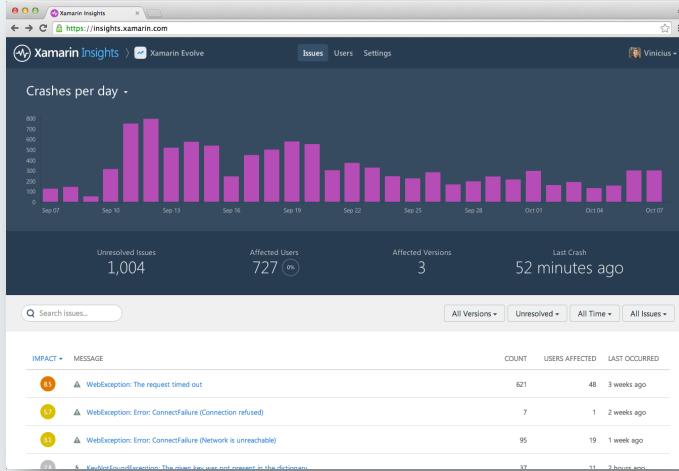


Xamarin Test Cloud

The screenshot shows the Xamarin Test Cloud interface for a test run titled "Flipboard" on "master". The test "User creates an account" has passed with 5 successful steps. Other tests listed include "User signs in with Facebook", "User signs in with Google", "User has incorrect password", "User has incorrect email", "User signs out", "Reading articles", "User reads the cover story", "User reads the News section", "User reads the Technology section", "User reads Twitter articles", "User adds a section", "User comments on an article", "Collecting articles", "User collects an article", and "User collects and shares an article". The interface displays 15 device icons, each showing the Flipboard app running on a different Android device model and version.

Test Step	Status	Device	Version
Given I am on the start screen	Pass	LG Nexus 5	Android 4.4.2
When I go to the login screen	Pass	Samsung Galaxy S II	Android 4.1.2
And I enter valid credentials	Pass	Samsung Galaxy S III	Android 4.1.2
Then I should be logged in	Pass	Samsung Galaxy Duos	Android 4.0.4
User signs in with Facebook	Pass	Samsung Galaxy Core	Android 4.1.2
User signs in with Google	Pass	Samsung Galaxy Grand Duos	Android 4.2.2
User has incorrect password	Pass	Samsung Galaxy S Duos 2	Android 4.2.2
User has incorrect email	Pass	LG Nexus 4	Android 4.4.2
User signs out	Pass	HTC One	Android 4.4.2
Reading articles	Pass	Samsung Galaxy Note	Android 4.1.2
User reads the cover story	Pass	Sony Xperia Z	Android 4.3
User reads the News section	Pass	LG G2	Android 4.4.2
User reads the Technology section	Pass	Samsung Galaxy Grand Duos	Android 4.1.2
User reads Twitter articles	Pass	Huawei Ascend Y300	Android 4.1.1
User adds a section	Pass	Samsung Galaxy Centura	Android 4.0.4
User comments on an article	Pass		
Collecting articles	Pass		
User collects an article	Pass		
User collects and shares an article	Pass		

Testez vos applications sur des milliers de devices dans le cloud



Monitez votre application en temps réel. Récupérez les crashes et les exceptions détaillées de votre application suite à son utilisation par de vrais utilisateurs.

Create native iOS, Android, Mac and Windows apps in C#.

Join our community of 999,895 developers.

[Download now](#)



> Get Started Today: xamarin.com/download

> github.com/Julien-Mialon
> blog.julienmialon.com