



Développement d'application cross- platform avec Xamarin

Julien Mialon : mialon.julien@gmail.com

Valentin Jubert : valentin.jubert@outlook.fr

Au sommaire...

Séance 1

- Écosystème mobile
- Natif ou Cross-Platform?
- Xamarin et ses outils
- C# & .NET
- Xamarin.Forms

Séance 2

- Xamarin.Forms avancé
- DevOps (CI / CD / CT) + AppCenter
- OAuth2 password flow
- Push notifications
- Cache
- Architectures
- Aperçu de Flutter
- Publication sur les stores



Xamarin.Forms

Bindings

```
public class HomeViewModel : ViewModelBase
{
    private string _title;
    private ObservableCollection<Todo> _todos;

    public string Title
    {
        get => _title;
        set => SetProperty(ref _title, value);
    }

    public ObservableCollection<Todo> Todos
    {
        get => _todos;
        set => SetProperty(ref _todos, value);
    }
}
```

Bindings

- La vue définit un contexte pour les bindings dans son constructeur
- Ici HomeViewModel

```
public HomePage()  
{  
    InitializeComponent();  
    BindingContext = new HomeViewModel();  
}
```

Bindings

- Une fois le contexte définit, toutes les **PROPRIÉTÉS PUBLIC** du contexte deviennent disponibles pour le binding

```
<Label Text="{Binding Title}" />
```



```
<Label Text="{Binding _title}" />
```



Bindable Properties

- Bindable Property = Propriété supportant le binding
- Dans une vue/control custom, c'est le seul moyen d'avoir du binding pour vos nouvelles propriétés
- Exemple: je crée un control qui permet de gérer un champ texte et un label au-dessus. J'y ajoute deux BP
 - Title: le texte qui ira dans le label
 - Text: le texte saisi dans l'input

Bindable Properties

```
public static readonly BindableProperty TitleProperty =  
    BindableProperty.Create(  
        propertyName: nameof(Title),  
        returnType: typeof(string),  
        declaringType: typeof(InputControl),  
        defaultValue: null,  
        defaultBindingMode: BindingMode.OneWay,  
        propertyChanged: OnTitlePropertyChanged  
    );  
  
public string Title  
{  
    -  
    get => (string)GetValue(TitleProperty);  
    set => SetValue(TitleProperty, value);  
}
```




DÉMO

— live





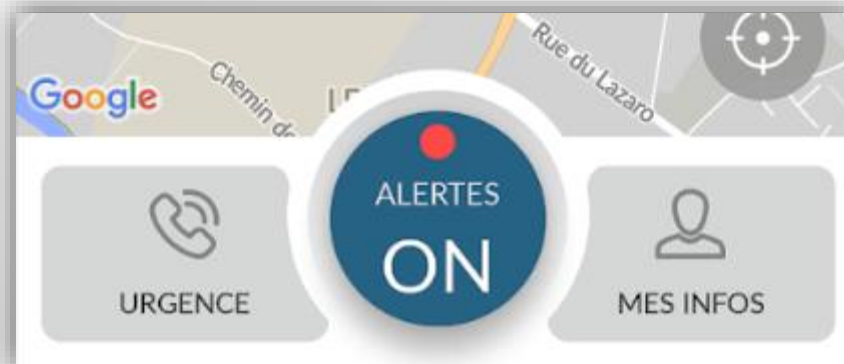
Aa Bb Cc Dd

$2 + 2 = \dots$

Questions ?

Custom Renderer

- Comment faire pour réaliser des vues très complexe ?



Custom Renderer

- Les custom renderers permettent de définir une abstraction « partagée » pour des contrôles avec des rendus natif
- C'est de cette manière que sont réalisés tous les control du framework

Custom Renderer

```
public class MyEntry : Entry
{
}
```

```
<ContentPage ...
  xmlns:local="clr-namespace:CustomRenderer;assembly=CustomRenderer"
  ...>
  ...
  <local:MyEntry Text="In Shared Code" />
  ...
</ContentPage>
```

Custom Renderer

```
[assembly: ExportRenderer(typeof(MyEntry), typeof(MyEntryRenderer))]  
namespace CustomRenderer.Android  
{  
    class MyEntryRenderer : EntryRenderer  
    {  
        public MyEntryRenderer(Context context) : base(context)  
        {  
        }  
  
        protected override void OnElementChanged(ElementChangedEventArgs<Entry> e)  
        {  
            base.OnElementChanged(e);  
  
            if (Control != null)  
            {  
                Control.SetBackgroundColor(global::Android.Graphics.Color.LightGreen);  
            }  
        }  
    }  
}
```



Aa Bb Cc Dd

$2 + 2 = \dots$

Questions ?



DevOps

DevOps

- La théorie : unifier le développement et l'exploitation
 - Le développeur développe la fonctionnalité, la teste, la pousse en prod et en assure le suivi
 - Tester au plus tôt et de manière automatisée
 - Tester dans un environnement proche de la production
 - Release régulière
 - Boucle de feedback rapide

DevOps

- Le point le plus problématique : release régulière
- Mise en place de :
 - CI = Continuous Integration
 - CT = Continuous Testing
 - CD = Continuous Deployment

DevOps

- Plusieurs solutions existent :
 - Bitrise (Build + Test)
 - Azure DevOps (Build + Test)
 - **AppCenter** (Build + Test + Déploiement)
 - ...

Add new app



App name:

LiveCoding

Icon:



Description:

Enter a brief description (optional)

Owner:



OS:

- ☐ iOS
- ☒ Android
- ☐ Windows
- ☐ macOS Preview

Platform:

- ☐ Java / Kotlin
- ☐ React Native
- ☐ Cordova Preview
- ☒ Xamarin
- ☐ Unity

Using a different platform? [Let us know.](#)

Add new app



LiveCoding

Android



Overview



Build



Test



Distribute



Diagnostics



Analytics



Push



Settings

Select a service



Azure DevOps



GitHub



Bitbucket



● master



This branch has not been configured to build yet.

It sure looks intriguing, though!

LAST COMMIT



Add postman collection + update cours/td su...

Julien Mialon

2 weeks ago

[Configure build](#)

Build app

Project:

LiveCoding2.Android.csproj



Configuration:

Debug



SDK version:

Xamarin.Android 9.0



Build scripts:

None

Learn more about [custom build scripts](#)

Build frequency:



Build this branch on every push



Manually choose when to run builds

Automatically increment version code

Choose a format to increment your builds.



Off

Sign builds

☒ On

Builds must be signed to run on devices.

Keystore



Keystore file:
livecoding.keystore



Environment variables

Keystore password:

.....

Key alias:

.....

Key password:

.....

Test on a real device Free

☒ On

Verify that your build works on a real device by running a launch test.

[!\[\]\(99f58673407353e96a019fbca558fd72_img.jpg\) Do not use personal data in your tests](#)

Distribute builds

☒ On

Release your app to testers or to a store.

Collaborators



1 tester



Running...

00:37

STARTED
just now

LAUNCH TESTED
N/A

SIGNED
N/A



Build 1


Manual build

DURATION
5 min 59 sec

LAUNCH TESTED
[Results](#)

SIGNED
Yes

Test runs

Date	Version	Duration	Status	Results
Feb 23, 2019, 6:07:39 PM	1.0 (1)	1 min	 PASSED	1 test passed <div></div>

Releases

Release history

Number ↓

Version

Destinations

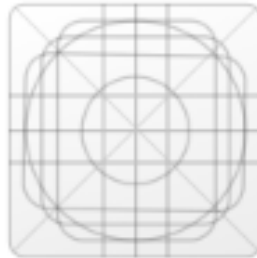


1

1.0 (1)

Collaborators

A new version of **LiveCoding for Android** is
available.



LiveCoding

1.0 (1)

for Android

Install

What's new

Add postman collection + update cours/td subjects

DevOps

- Au final, une fois le build configuré, il ne vous reste plus qu'à coder. À chaque push, votre code est :
 - **Compilé**
 - **Testé**
 - **Déployé**



Aa Bb Cc Dd

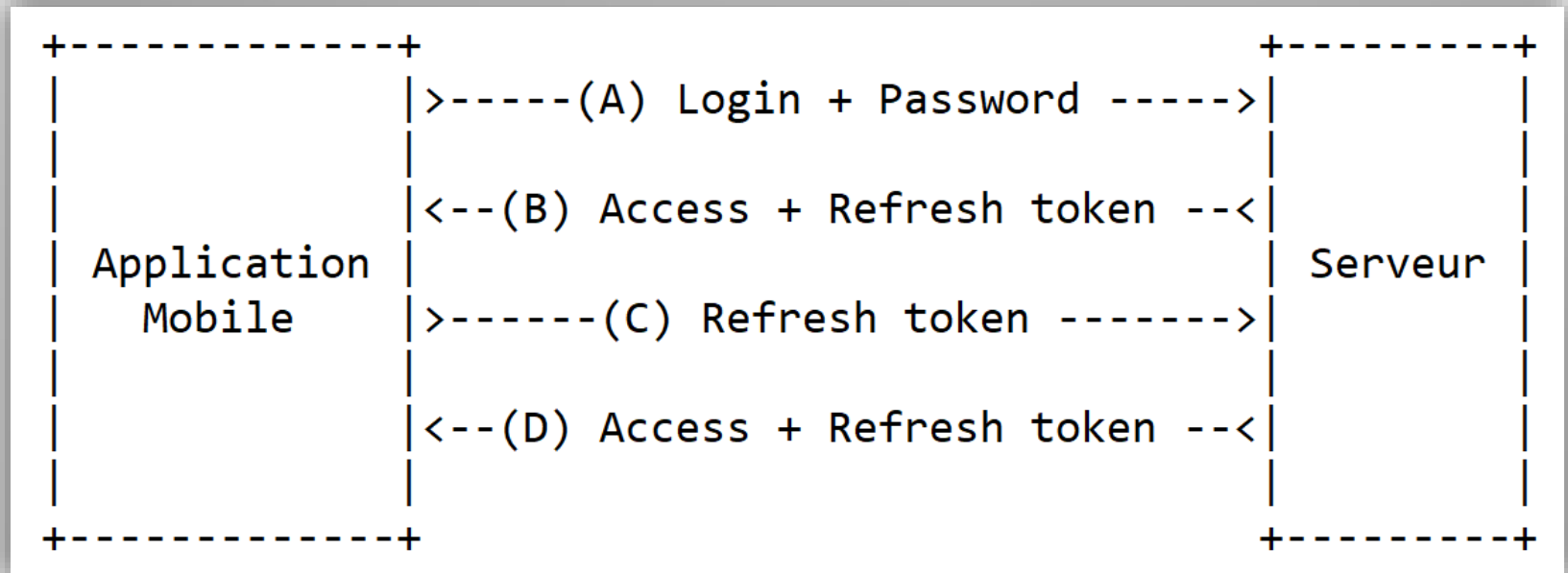
$2 + 2 = \dots$

Questions ?

OAuth2

- L'authentification dans une application mobile nécessite un certain niveau de sécurité
- Votre téléphone peut être volé, copié, ...
- Le but est que dans ce cas, vos login/mot de passe ne soit pas compromis

OAuth2



OAuth2

- Le login + le mot de passe n'est nécessaire qu'à la première étape => pas nécessaire de le stocker
- L'access token a une durée de vie limité (en général < 1 jour)
- Le refresh token permet de récupérer un nouvel access token et peut être invalidé afin de ne plus permettre la connexion

OAuth2

- Inconvénients :
 - Il faut envoyer l'access token dans toutes les requêtes HTTP
 - Il faut refresh l'access token quand il est expiré ou presque expiré

POST

`{{api_root}}/oauth2/login`

Params

Authorization

Headers (1)

Body ●

Pre-request Script

Tests ●

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

JSON (application/json) ▼

```
1 {  
2   "email": "{{oauth2_login}}",  
3   "password": "{{oauth2_password}}",  
4   "grant_type": "Password",  
5   "client_id": "{{oauth2_client_id}}",  
6   "client_secret": "{{oauth2_client_secret}}"  
7 }
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

JSON ▼



```
1 {  
2   "data": {  
3     "access_token": "5179d0a4deb648fa936ba4781c20f59a",  
4     "refresh_token": "9753a98d411e41a7b968c3bd867f685a",  
5     "expires_in": 86399,  
6     "token_type": "Bearer"  
7   },  
8   "is_success": true,  
9   "error_code": null,  
10  "error_message": null  
11 }
```

▶ front /me

GET



{{front_root}}/me

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests ●

KEY

VALUE



Authorization

Bearer 5179d0a4deb648fa936ba4781c20f59a

Key

Value

POST

`{{api_root}}/oauth2/refresh`

Params

Authorization

Headers (1)

Body ●

Pre-request Script

Tests ●

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

JSON (application/json) ▼

```
1 {  
2   "refresh_token": "{{refresh_token}}",  
3   "client_id": "{{oauth2_client_id}}",  
4   "client_secret": "{{oauth2_client_secret}}"  
5 }
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

JSON ▼



```
1 {  
2   "data": {  
3     "access_token": "182e35f6dbbe47d689e6a101d5408171",  
4     "refresh_token": "abf7f924bbfe4e71825ac02c722fb7c4",  
5     "expires_in": 86399,  
6     "token_type": "Bearer"  
7   },  
8   "is_success": true,  
9   "error_code": null,  
10  "error_message": null  
11 }
```



Aa Bb Cc Dd

$2 + 2 = \dots$

Questions ?



Push notifications

Push notifications



Twilio Blog

18:32

How to add push notifications to your Andr..

360 Hello 360 • now ^

April Bowman

Checkout the sunset in Berlin



data updated

7:24 PM

allow sync immediatly

APPROVE

REJECT

Push notifications

CRÉATION

Création
manuelle

Génération
automatique

ENVOI

Serveur
d'envoi

TRANSMISSION

serveurs de push

iOS
APNS

Android
GCM

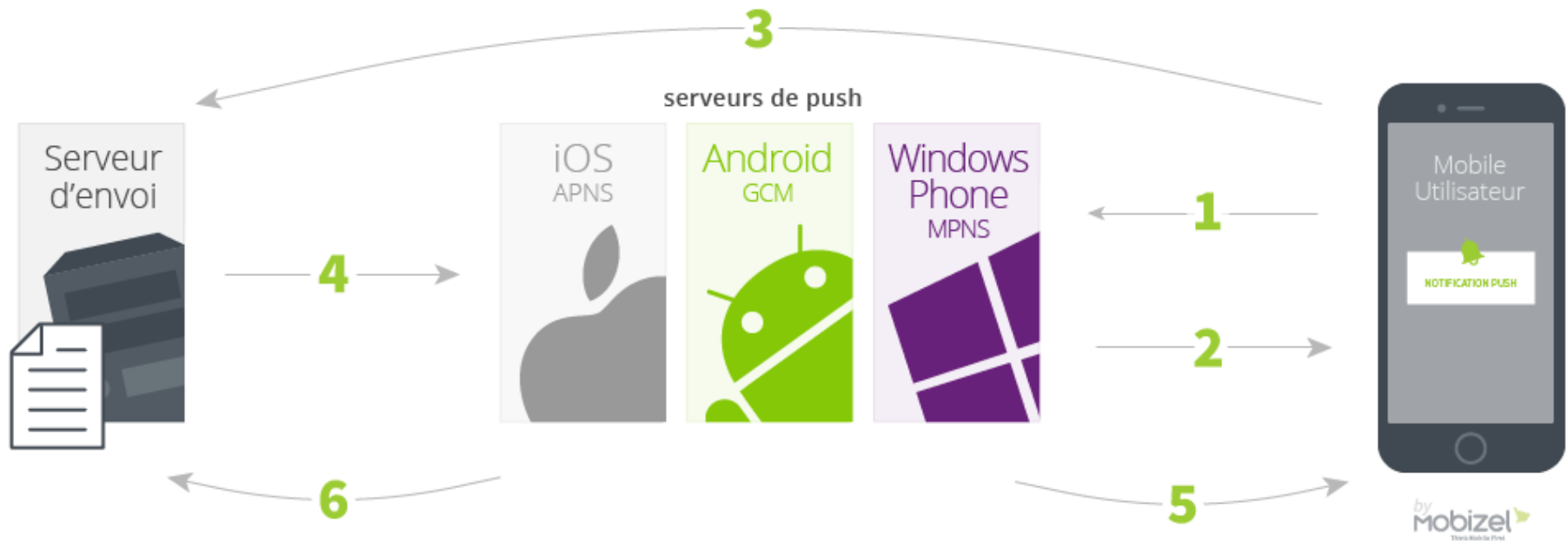
Windows
Phone
MPNS

RÉCEPTION

Mobile
Utilisateur

NOTIFICATION PUSH

Push notifications





Cache

Cache

- Pourquoi faire du cache ?

- Mode offline
- Interruption du réseau
- ...



- <https://github.com/jamesmontemagno/monkey-cache>

Cache

```
async Task<IEnumerable<Monkey>> GetMonkeysAsync()
{
    var url = "http://montemagno.com/monkeys.json";

    //Dev handle online/offline scenario
    if(!CrossConnectivity.Current.IsConnected)
    {
        return Barrel.Current.Get<IEnumerable<Monkey>>(key: url);
    }

    //Dev handles checking if cache is expired
    if(!Barrel.Current.IsExpired(key: url))
    {
        return Barrel.Current.Get<IEnumerable<Monkey>>(key: url);
    }

    var client = new HttpClient();
    var json = await client.GetStringAsync(url);
    var monkeys = JsonConvert.DeserializeObject<IEnumerable<Monkey>>(json);

    //Saves the cache and pass it a timespan for expiration
    Barrel.Current.Add(key: url, data: monkeys, expireIn: TimeSpan.FromDays(1));
}
```



Architectures

Architectures

- MVVM : Model-View-ViewModel
- MVC : Model-View-Controller
- Rx : Reactive Extensions
- Components

Rx

- Basé sur le fait de réagir à des événements
- Un flux d'événement se produit et on s'abonne dessus



```
map(x => 10 * x)
```



Rx

```
static void Main()
{
    var oneNumberPerSecond = Observable.Interval(TimeSpan.FromSeconds(1));

    var lowNums = from n in oneNumberPerSecond
                  where n < 5
                  select n;

    Console.WriteLine("Numbers < 5:");

    lowNums.Subscribe(lowNum =>
    {
        Console.WriteLine(lowNum);
    });

    Console.ReadKey();
}
```

Rx

- Pour en savoir plus : <http://reactivex.io/>
- Rx.Net : <https://github.com/dotnet/reactive>
- RxUI : <https://github.com/reactiveui/ReactiveUI>

Components

- Architecture provenant du web (React par exemple)
- Utilisé surtout sur React.Native, Flutter
- Possible sur Android & iOS

Components

- Nécessaire de découper son application en composant
- Exemple un composant bouton, un composant map, un composant détail d'un lieu...
- Chaque composant peut utiliser des composant enfant
- Chaque composant doit être auto-géré

Components

- Avec Xamarin :
 - Un ComponentViewModel => partagé iOS/Android
 - Un ComponentView
- ComponentViewModel => génère un état pour la vue (du json simple par exemple)
- ComponentView reçoit l'état et met à jour l'affichage en fonction

Components / Flutter

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutter'),
        ),
        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```




Publication sur les stores

Publication Android

- Google Play (ou autre store privé)
- Votre application doit être signé avec un keystore
- Vous devez ouvrir un compte google play (25\$ à vie)
- Uploadé votre application, remplir sa fiche et la publier
- Temps de publication : < 1h

Publication iOS

- AppStore
- Vous devez ouvrir un compte développeur Apple (99\$/an)
- Vous devez signer votre application avec un certificat Apple
- Uploadé votre application et remplir sa fiche
- Validation manuelle par les équipes d'Apple
- Temps de publication : 2/3 jours



Aa Bb Cc Dd

$2 + 2 = \dots$

Questions ?