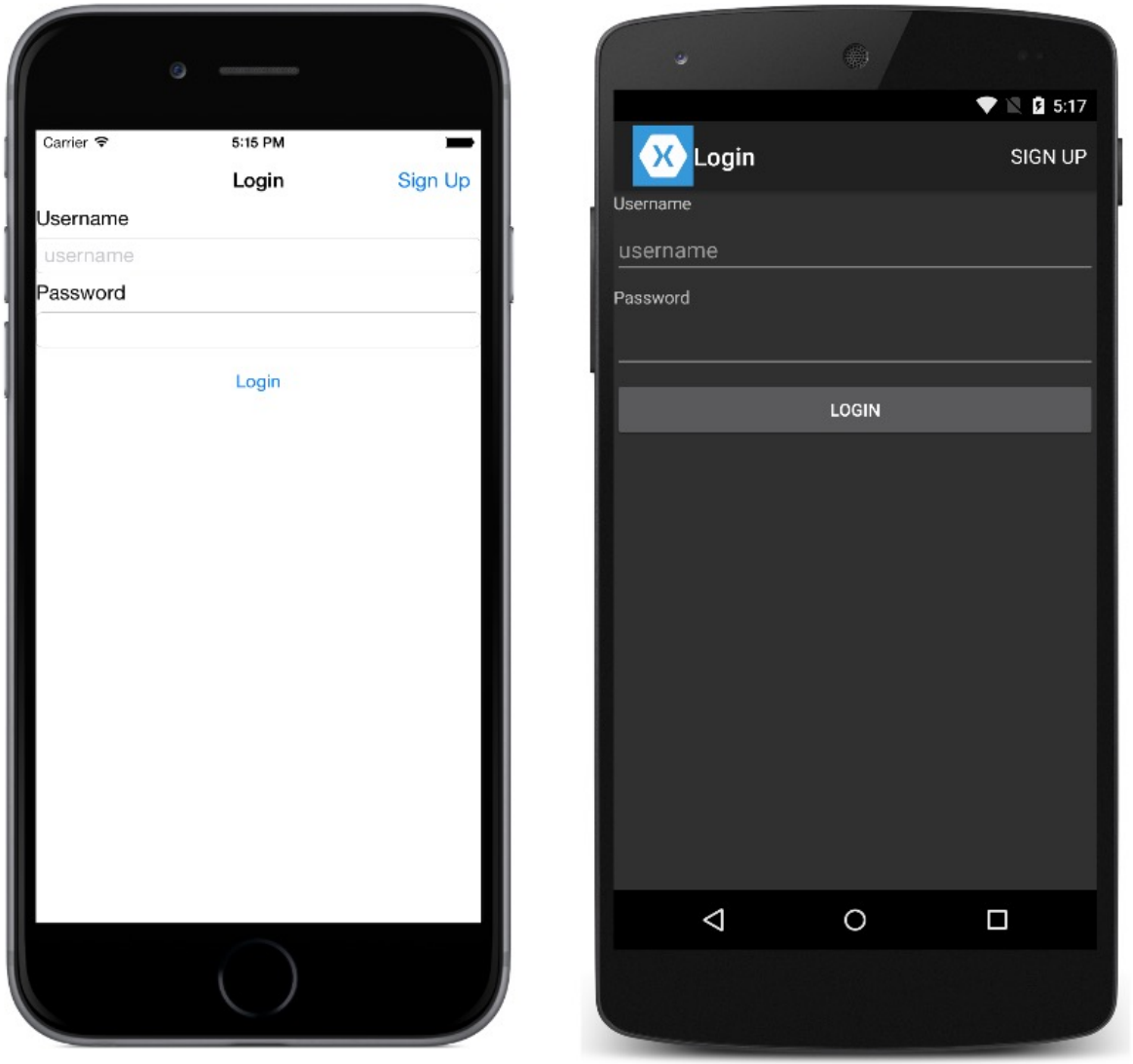
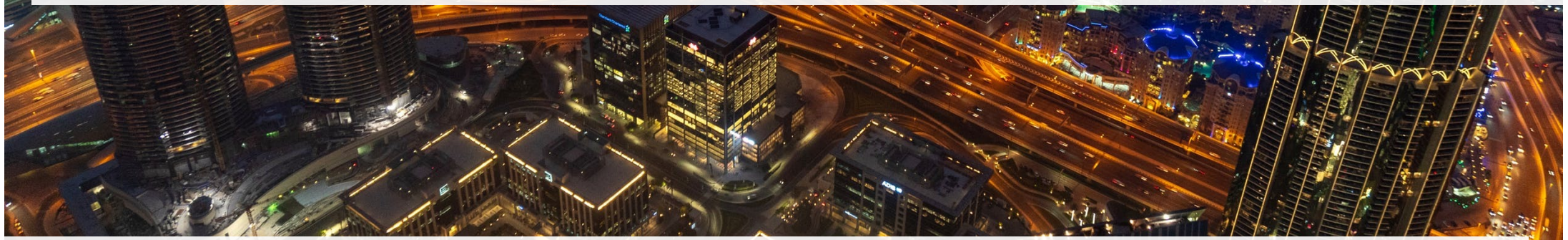

Adding authentication to your mobile app
Mark Allibone
Rey Technology
@mallibone



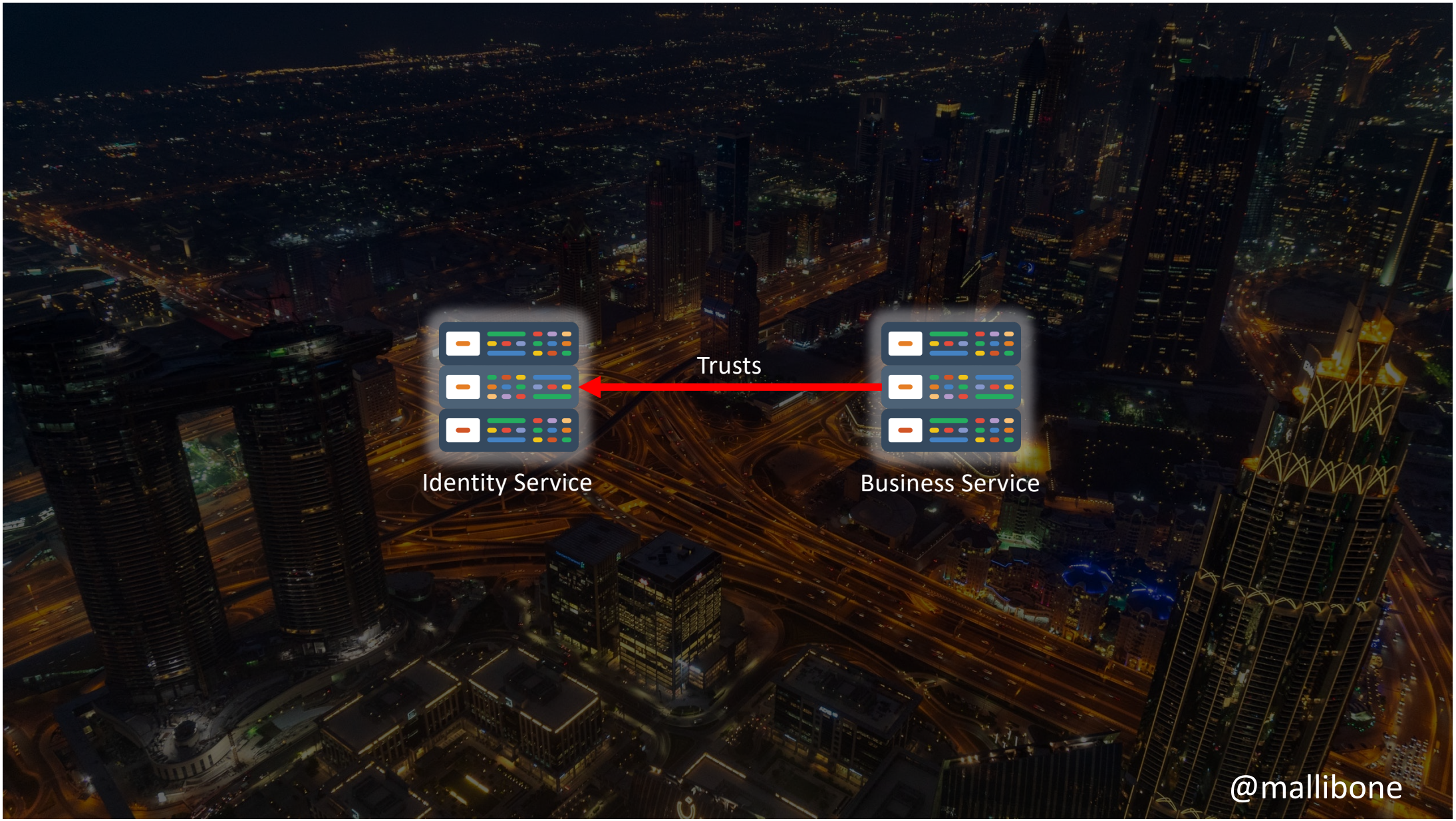
https://docs.microsoft.com/en-gb/samples/xamarin/xamarin-forms-samples/navigation-loginflow/

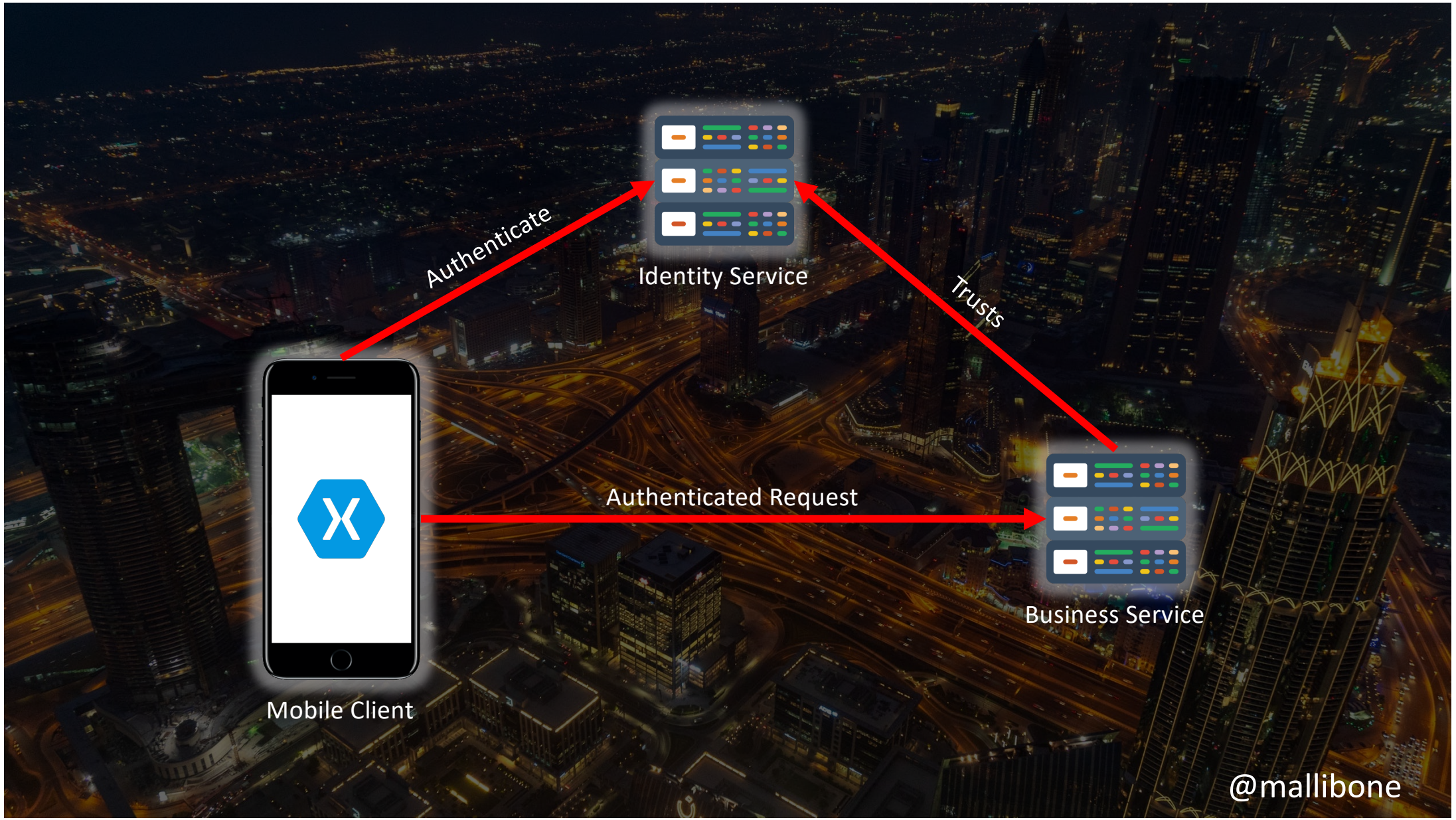
Why do we even need authentication?



@mallibone


Business Service
@mallibone


Trusts
Identity Service
Business Service
@mallibone


Authenticate
Identity Service
Trusts
Authenticated Request
Business Service
Mobile Client
@mallibone

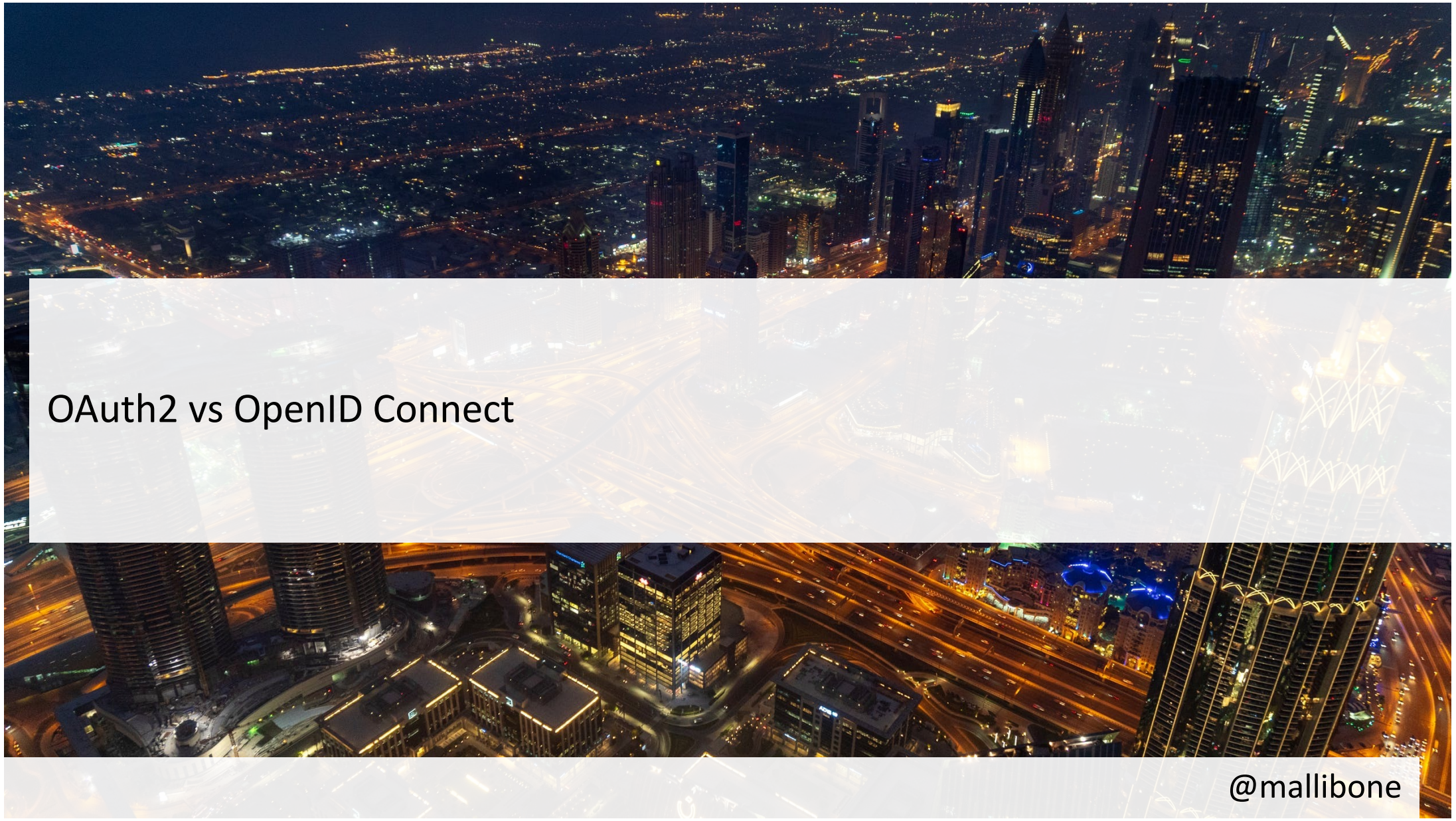
Having the right mindset


Choosing an Authentication Standard
@mallibone


OAUTH
OAUTH
@mallibone

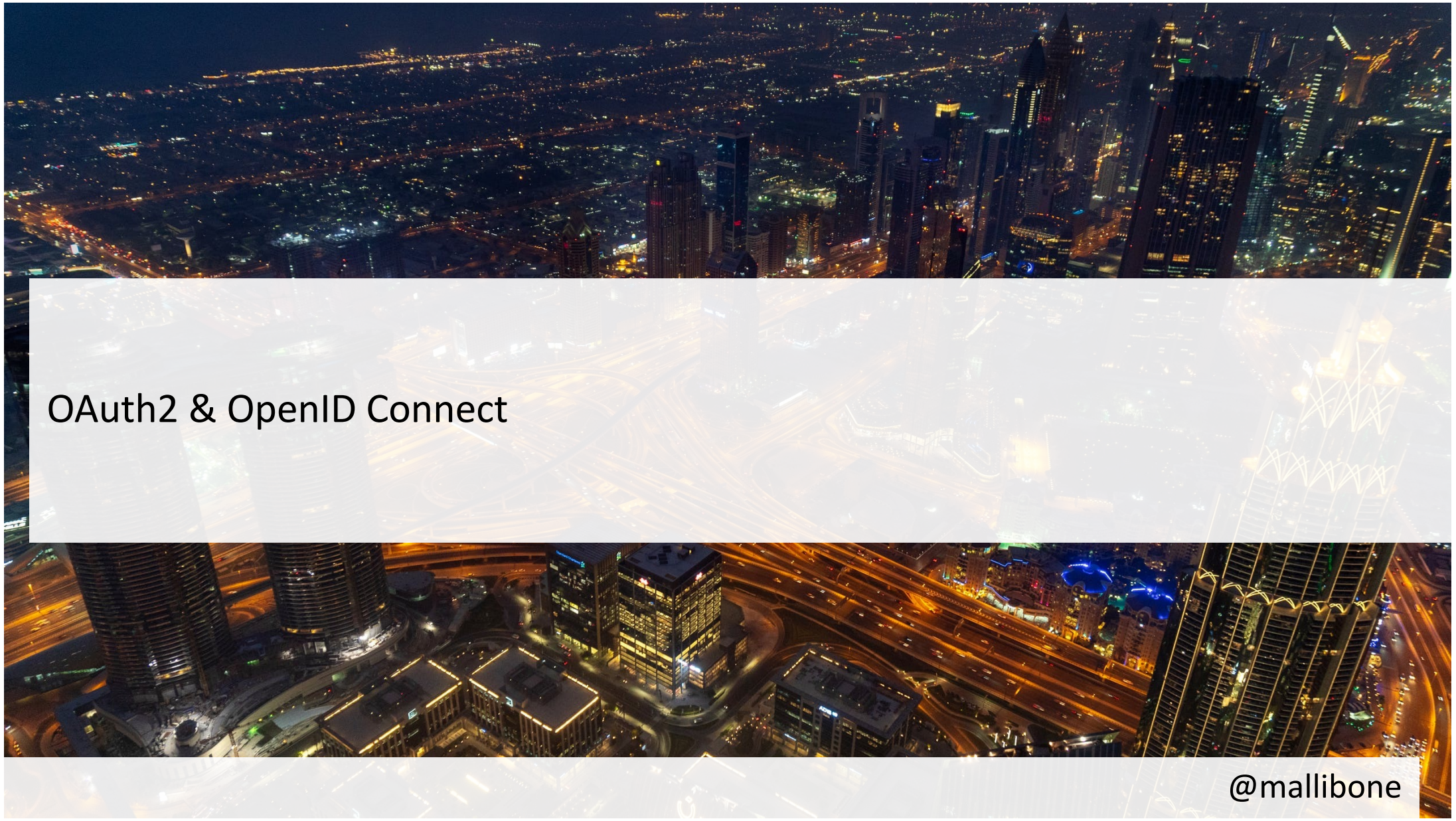

WHO AM I????
@mallibone


OpenID
Connect
@mallibone



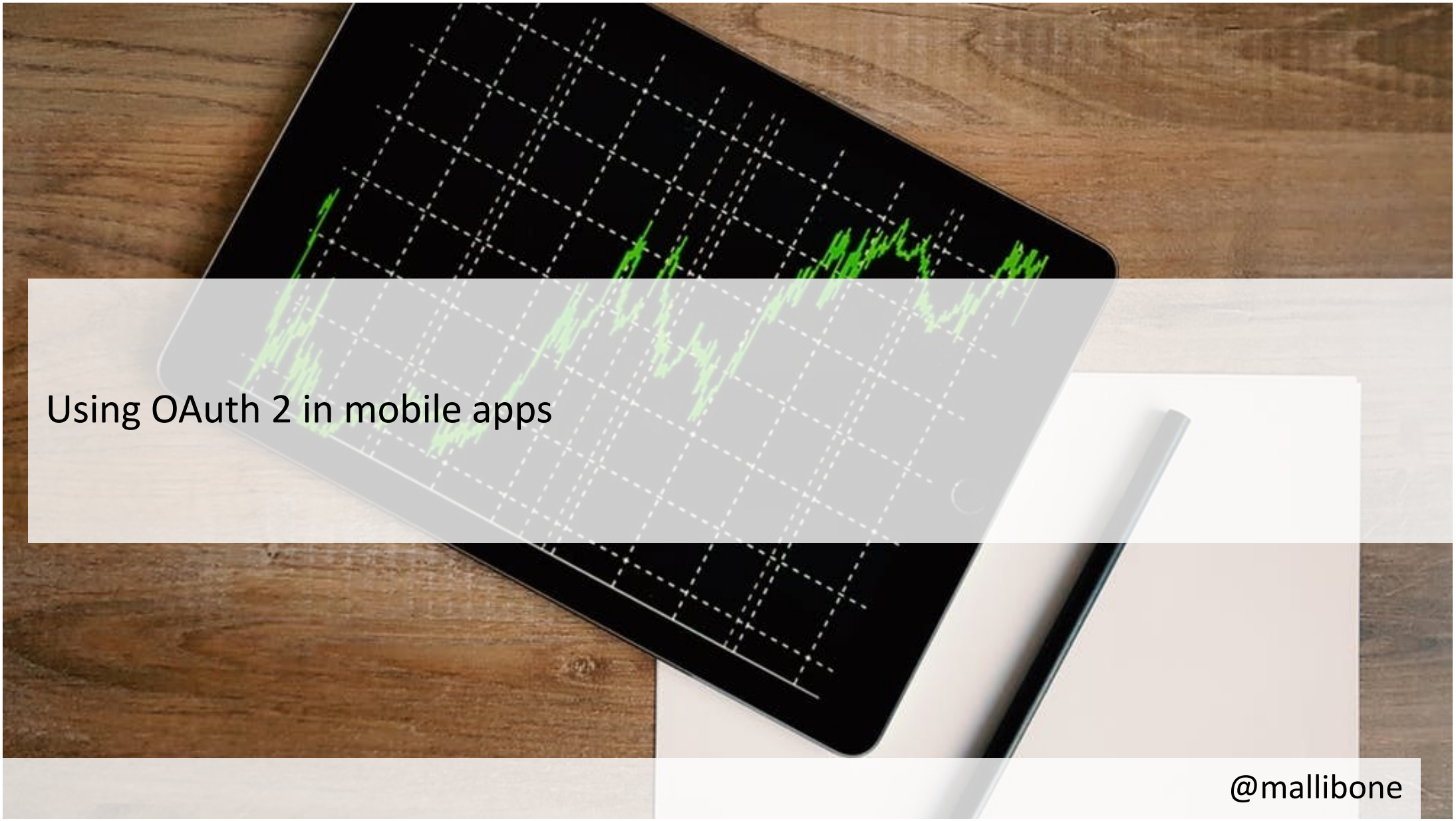
OAuth2 vs OpenID Connect

@mallibone



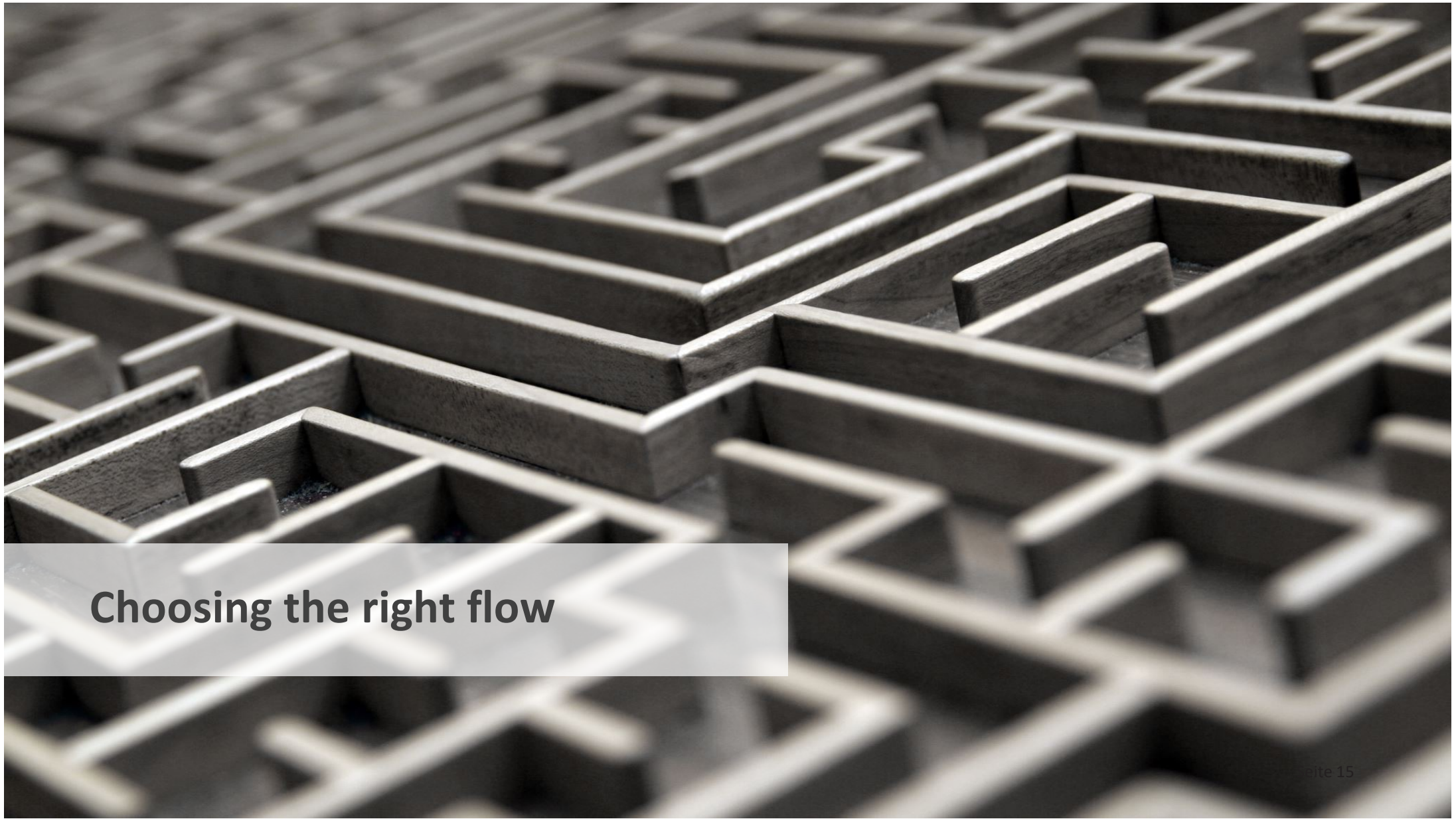
OAuth2 & OpenID Connect

@mallibone

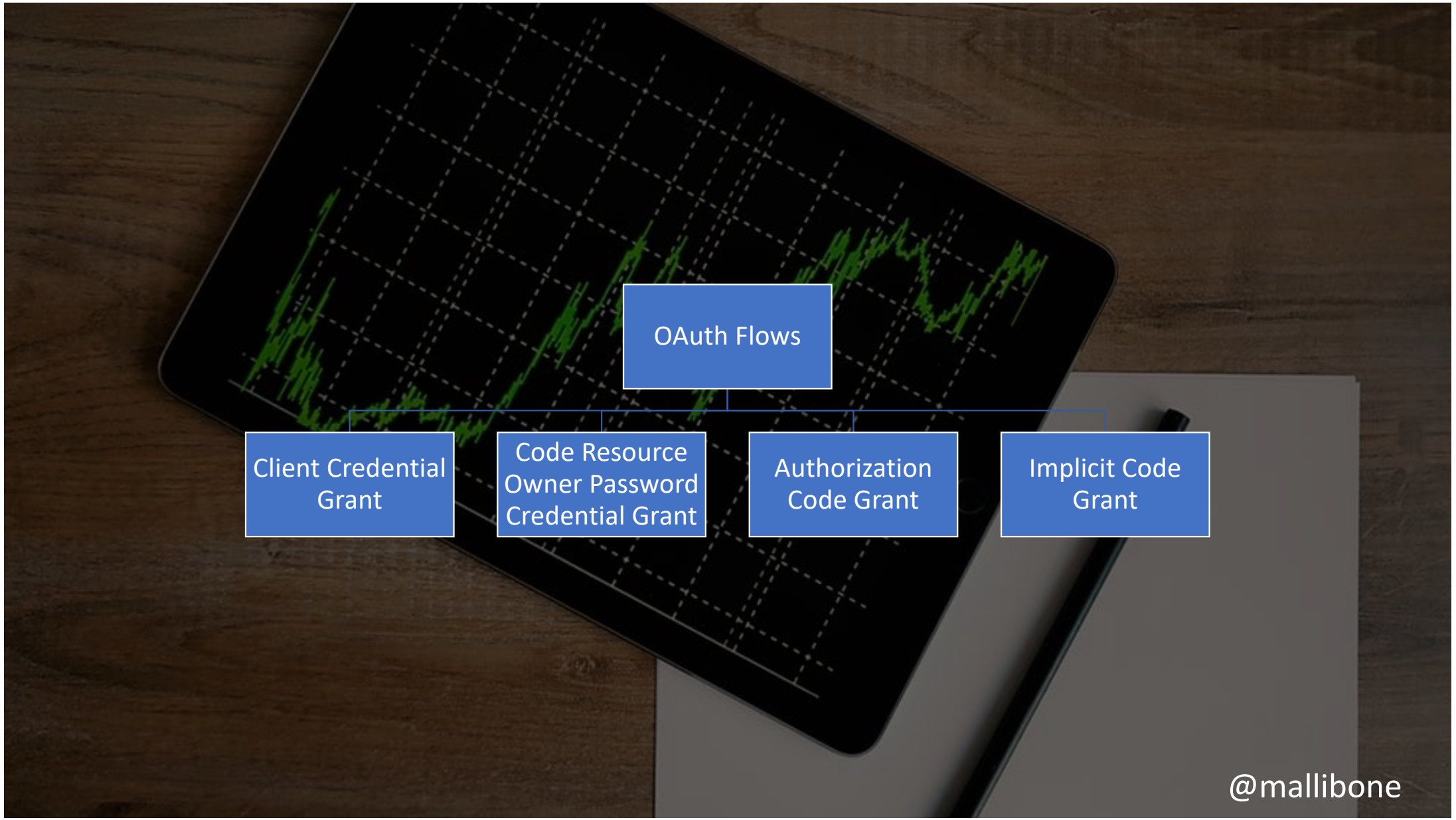


Using OAuth 2 in mobile apps

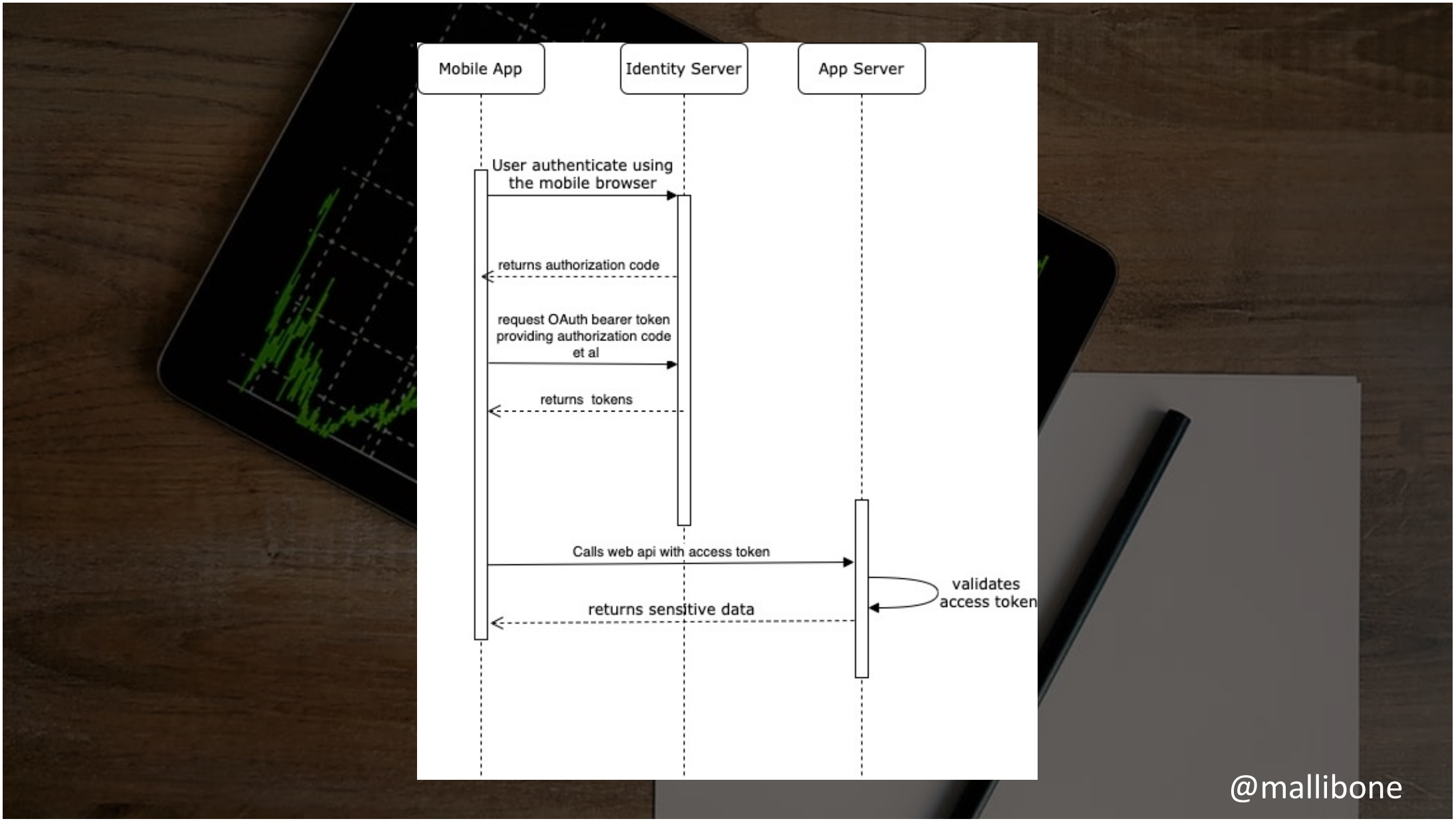
@mallibone



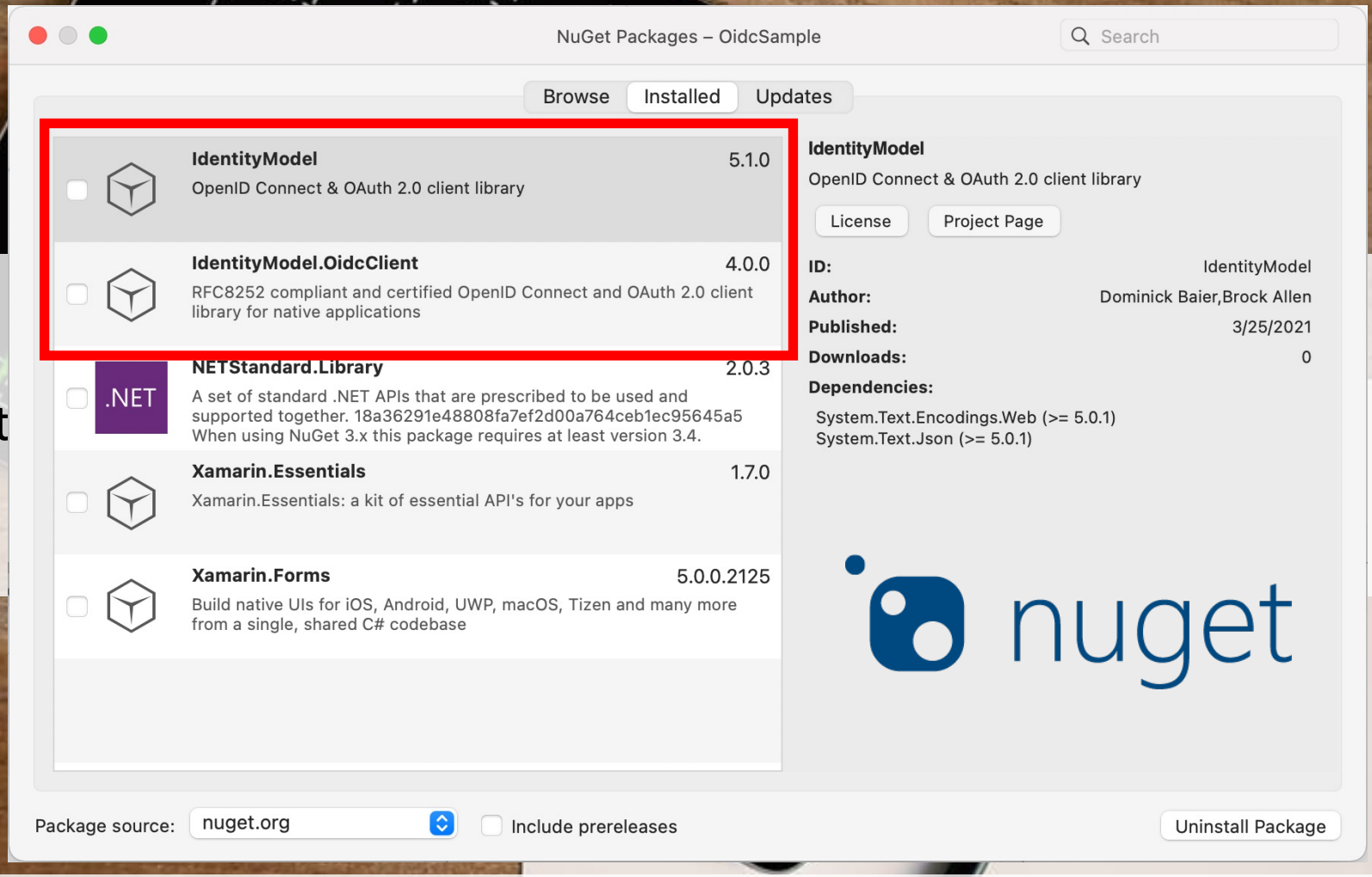
Choosing the right flow


OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone


OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone

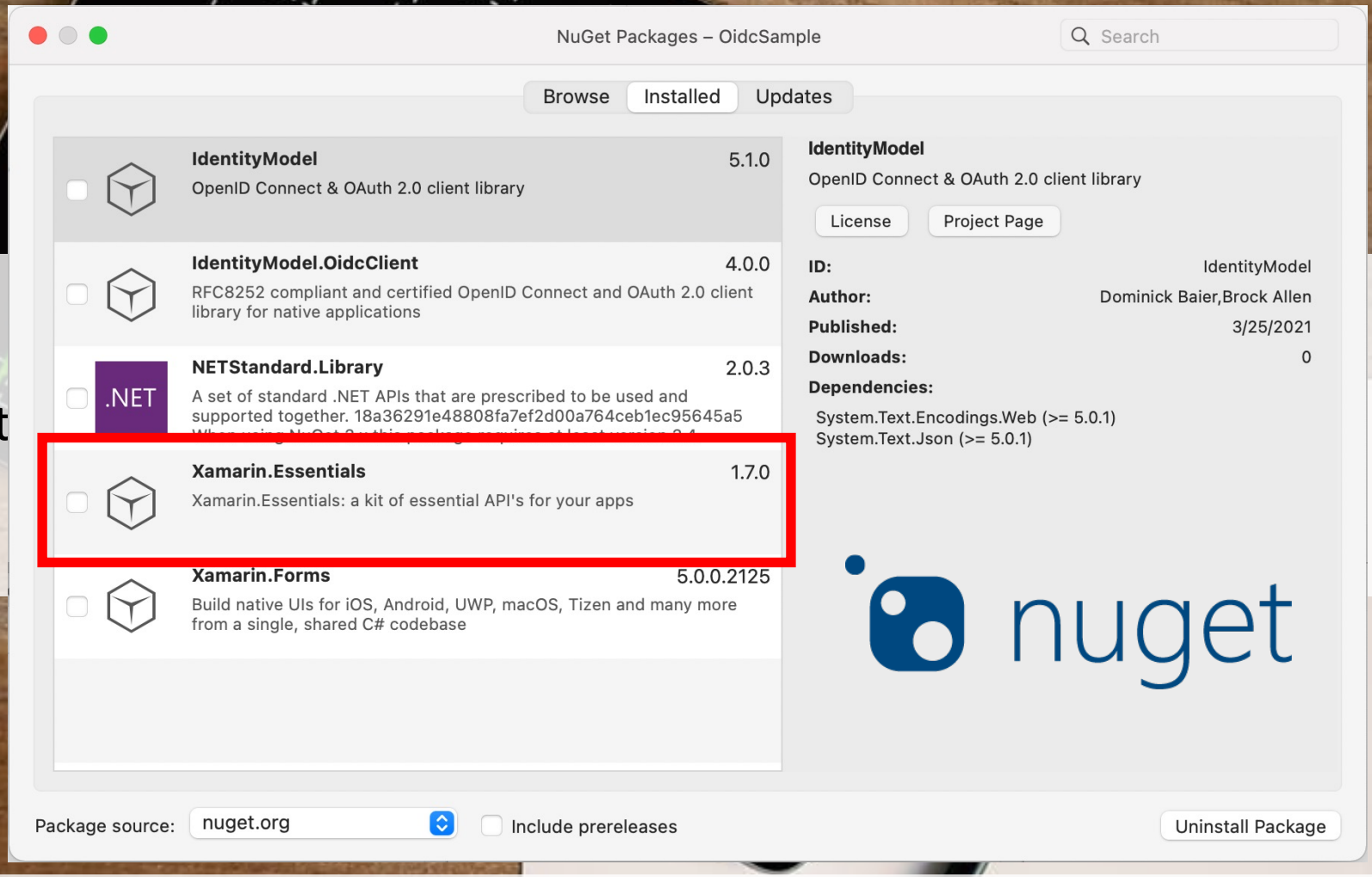

Mobile App
Identity Server
App Server
User authenticate using the mobile browser
returns authorization code
request OAuth bearer token providing authorization code et al
returns tokens
Calls web api with access token
validates access token
returns sensitive data
@mallibone

Getting start



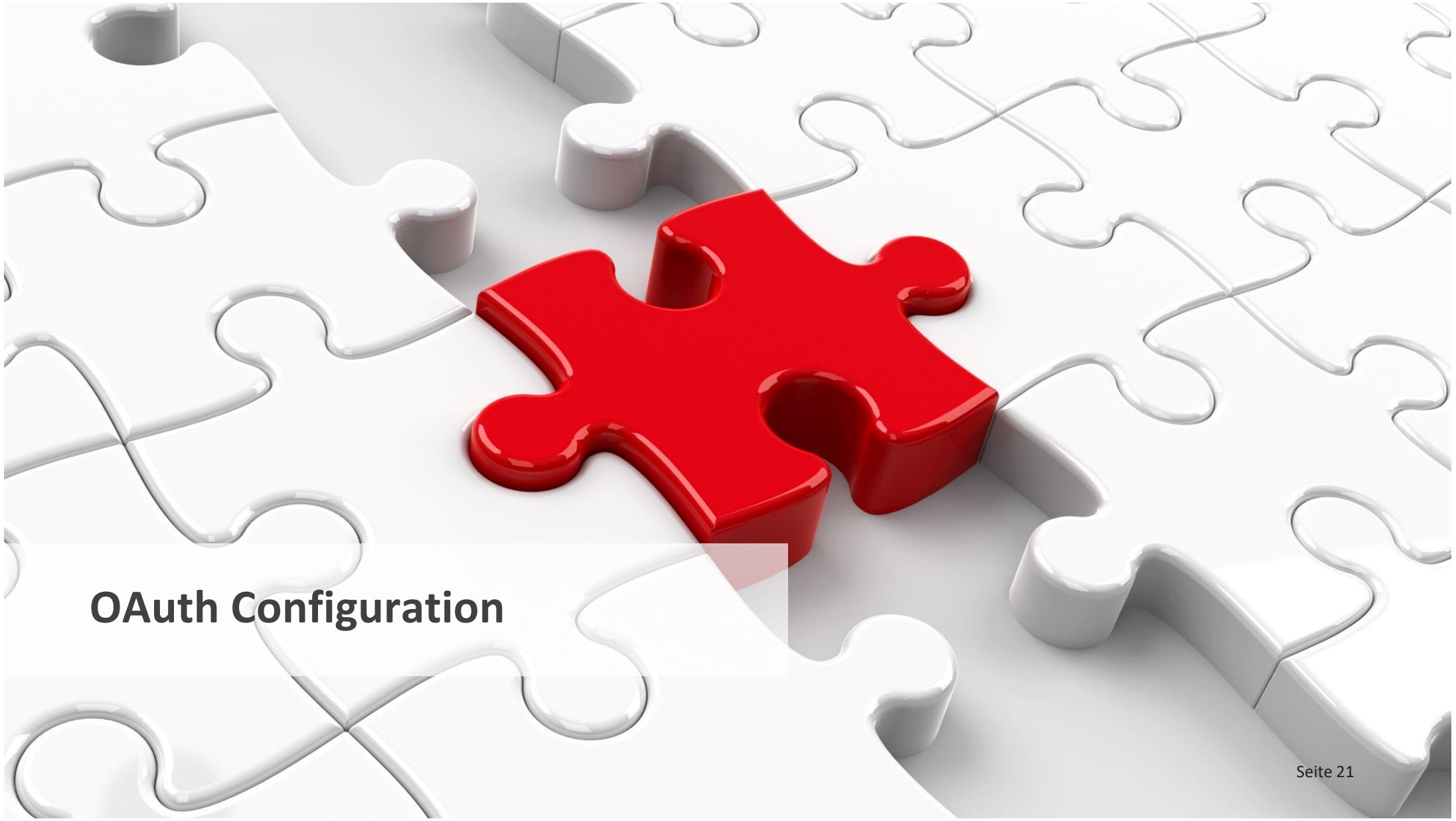
@mallibone

Getting start

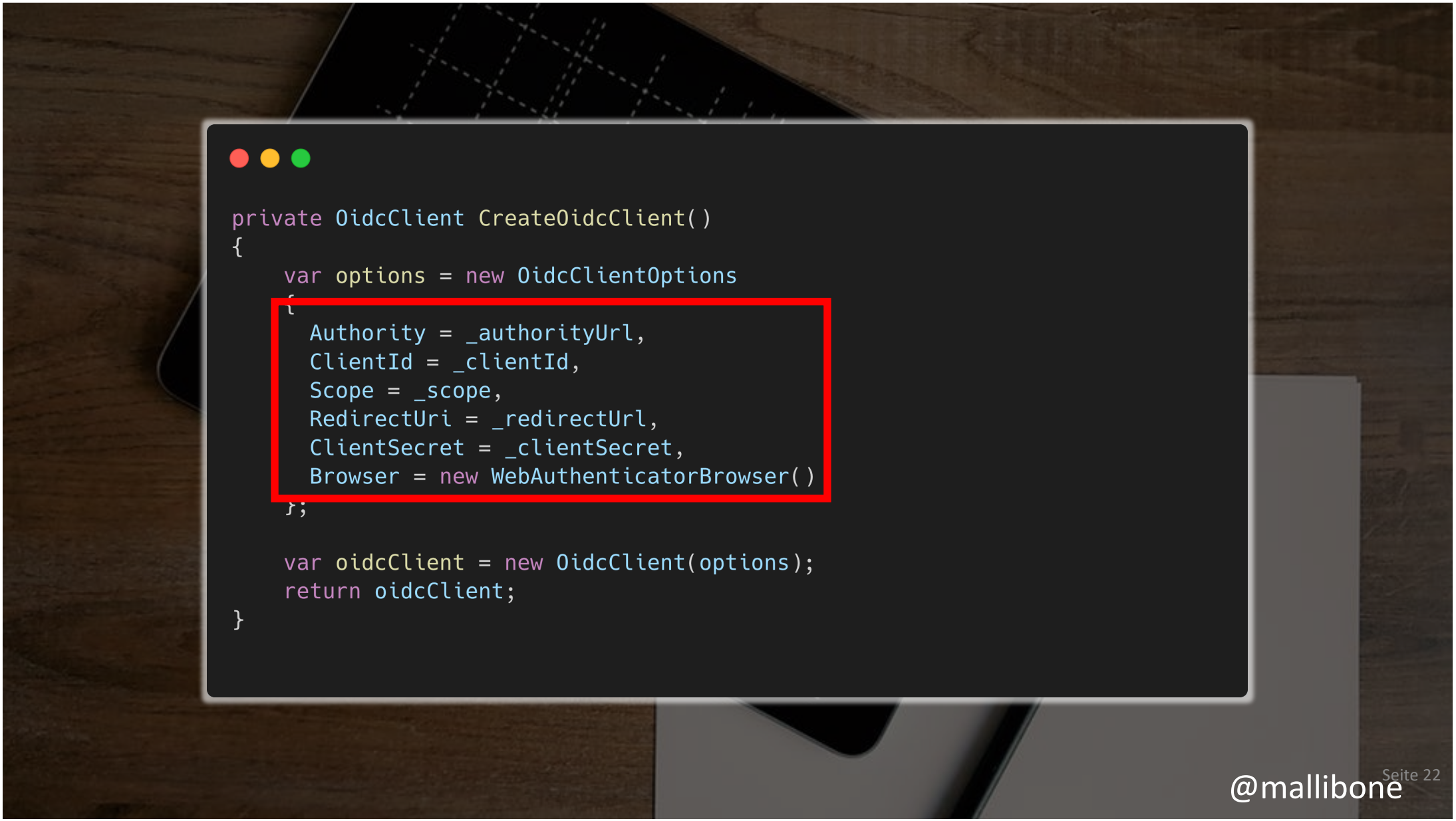


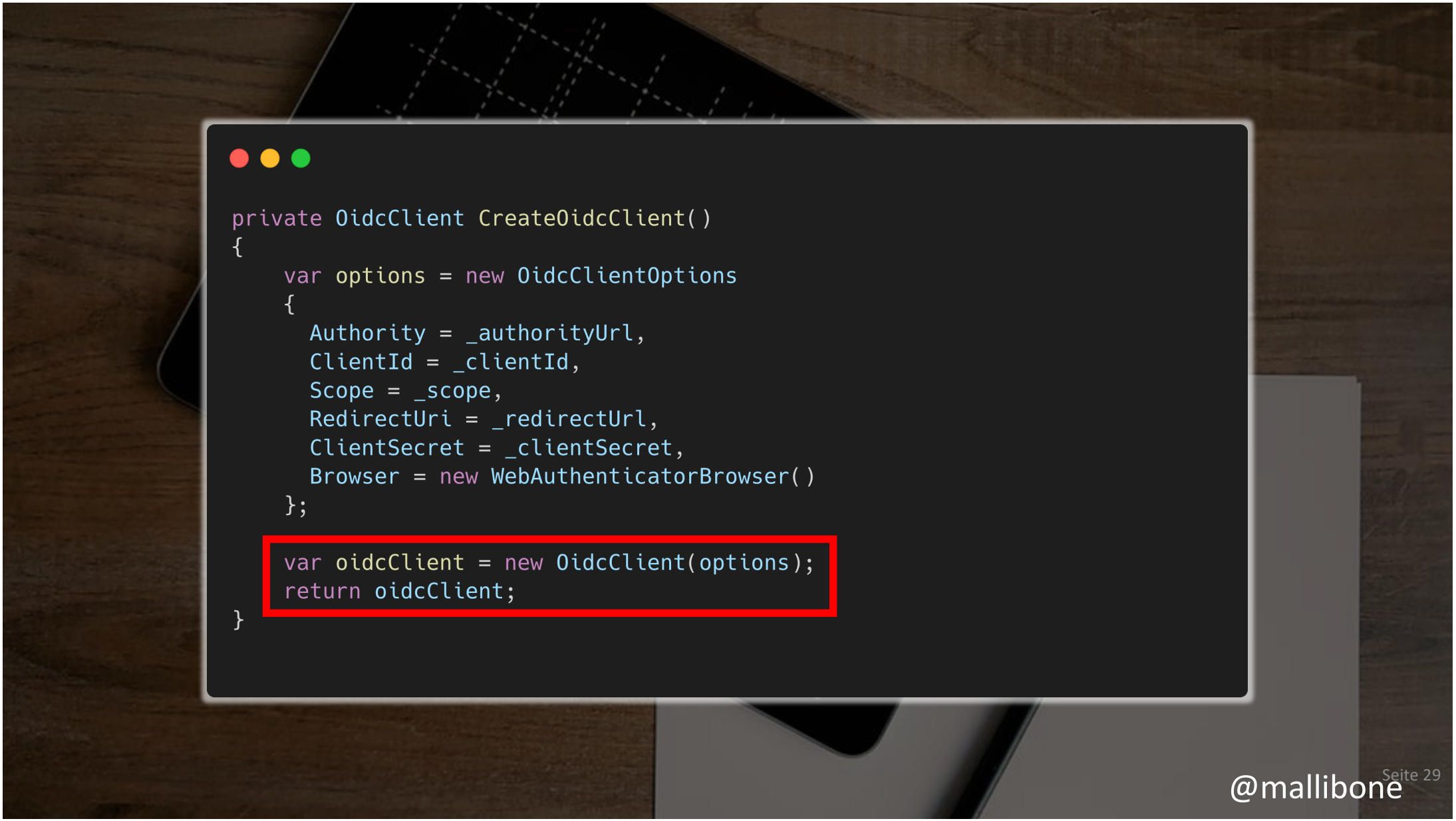
nuget

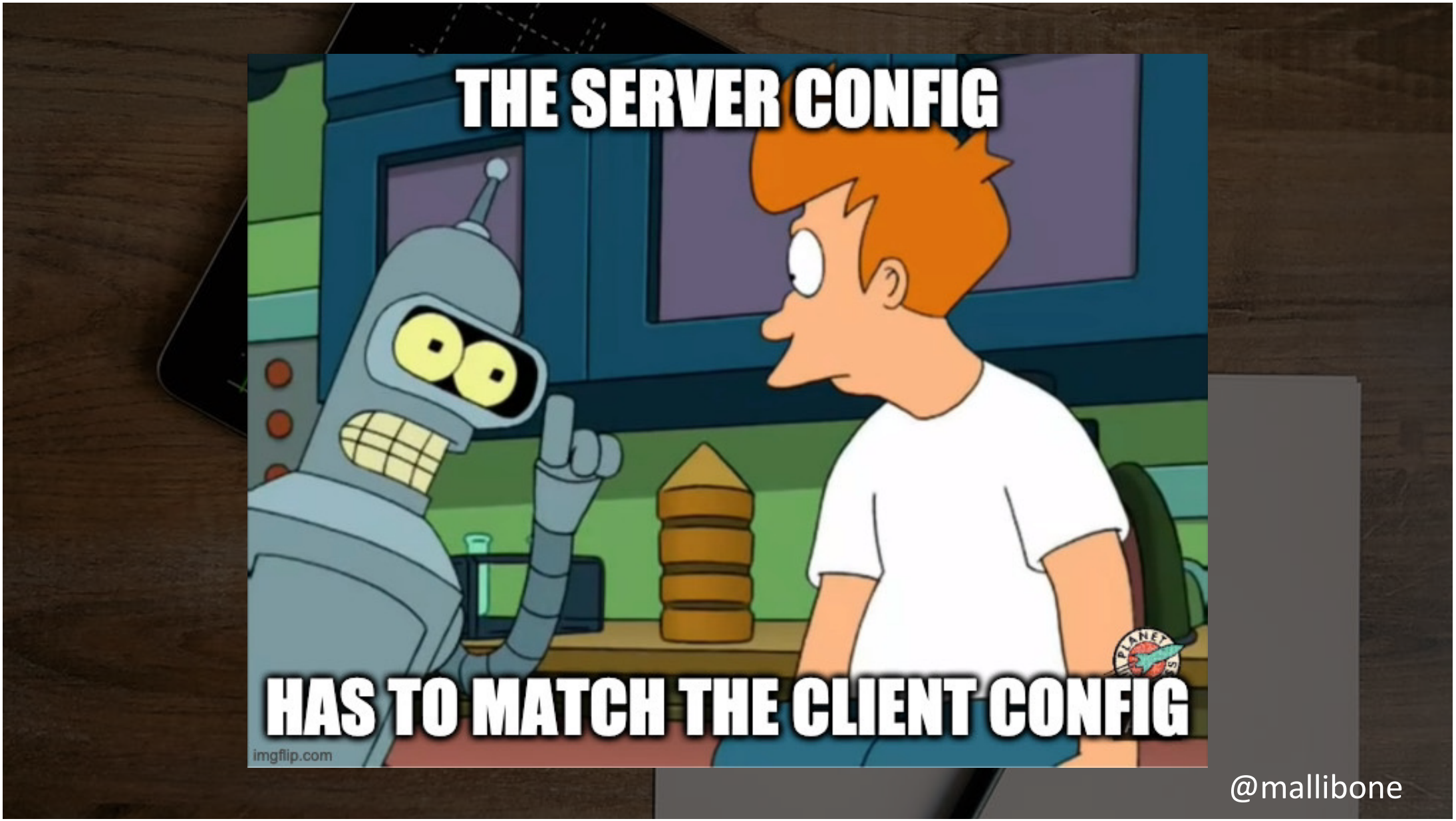
@mallibone



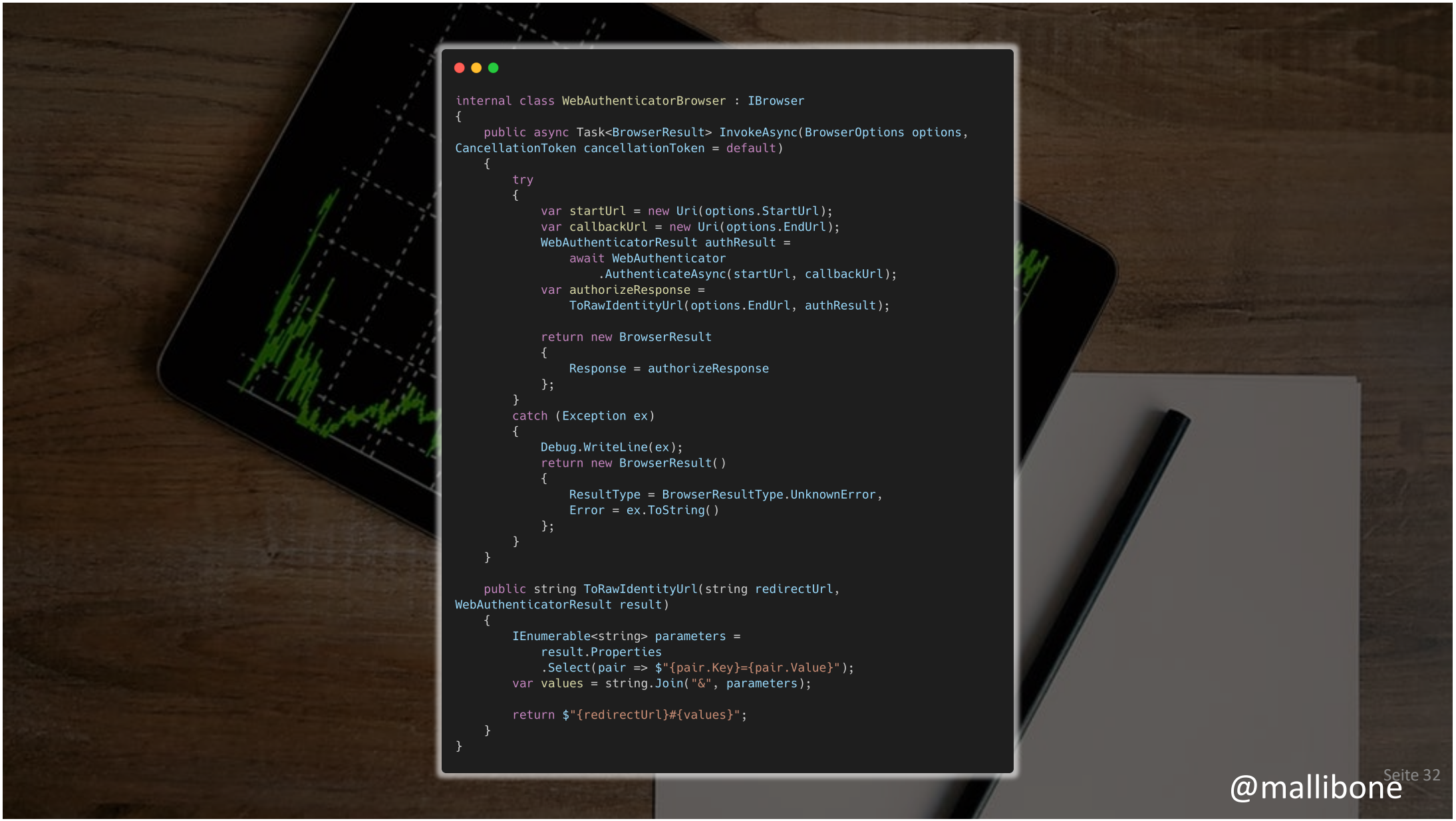
OAuth Configuration


private OidcClient CreateOidcClient()
{
var options = new OidcClientOptions
{
Authority = _authorityUrl,
ClientId = _clientId,
Scope = _scope,
RedirectUri = _redirectUrl,
ClientSecret = _clientSecret,
Browser = new WebAuthenticatorBrowser()
};
var oidcClient = new OidcClient(options);
return oidcClient;
}
@mallibone

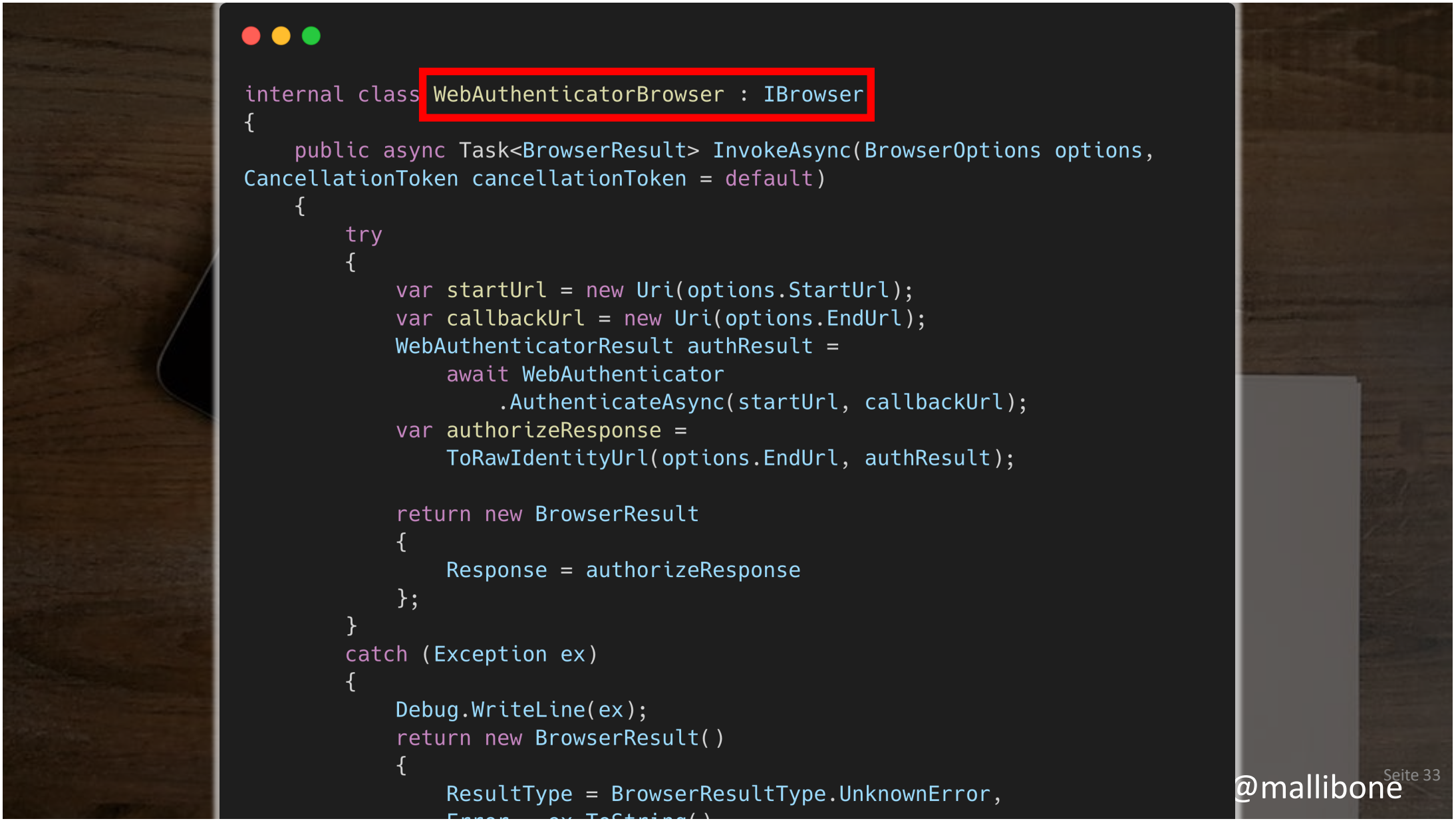

private OidcClient CreateOidcClient()
{
var options = new OidcClientOptions
{
Authority = _authorityUrl,
ClientId = _clientId,
Scope = _scope,
RedirectUri = _redirectUrl,
ClientSecret = _clientSecret,
Browser = new WebAuthenticatorBrowser()
};
var oidcClient = new OidcClient(options);
return oidcClient;
}
@mallibone

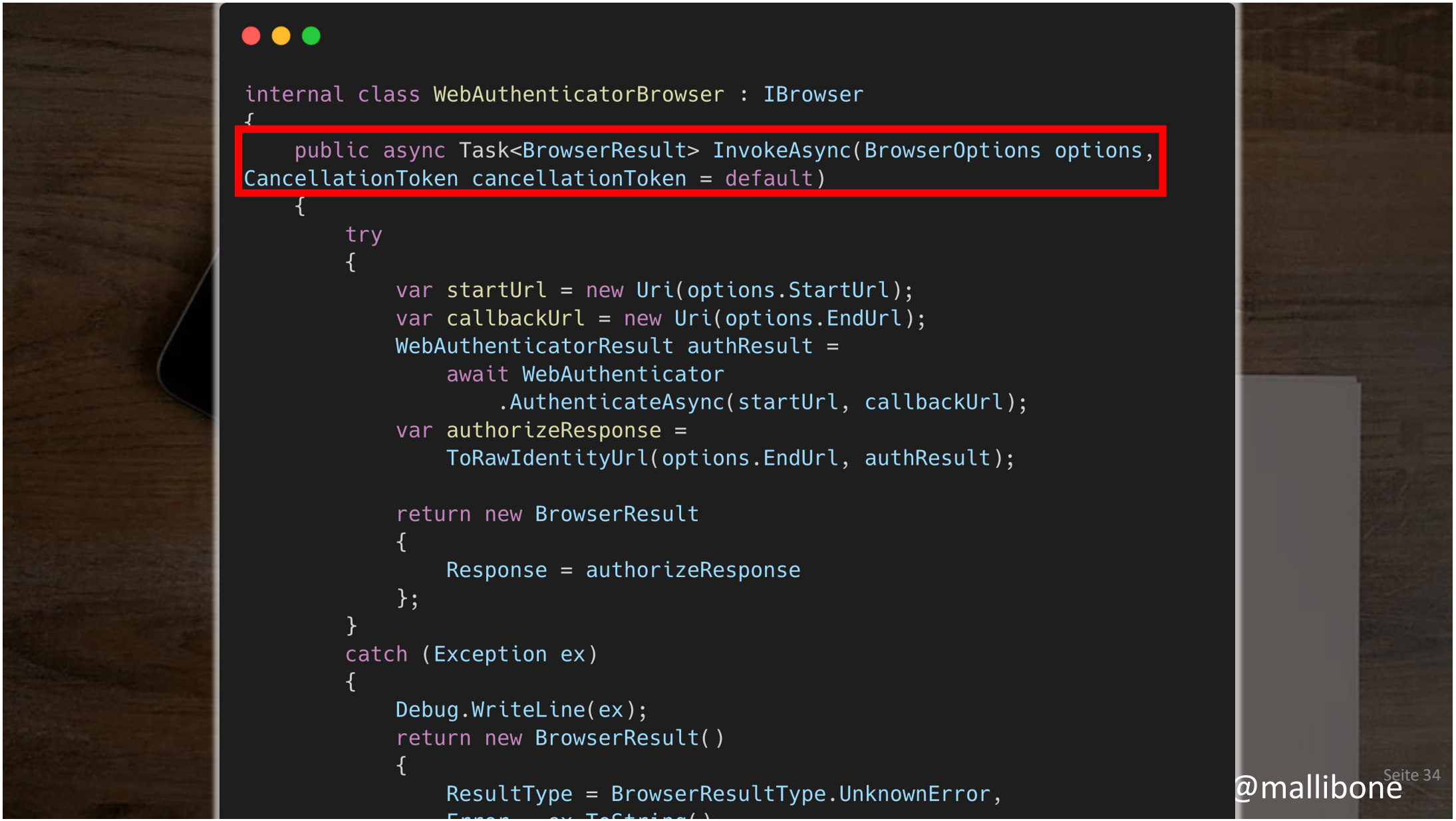

THE SERVER CONFIG
HAS TO MATCH THE CLIENT CONFIG
imgflip.com
@mallibone

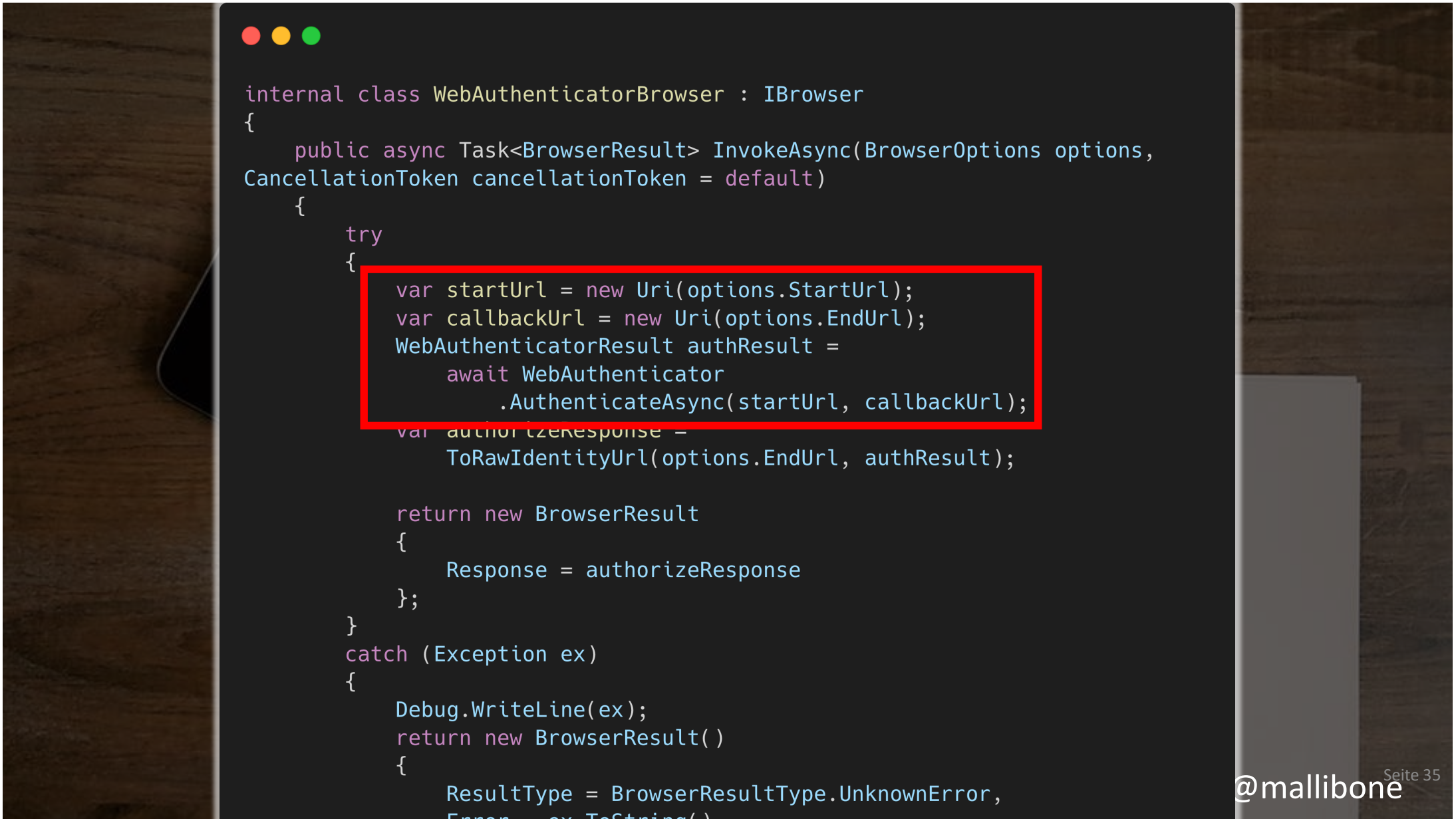
Implementing the Browser

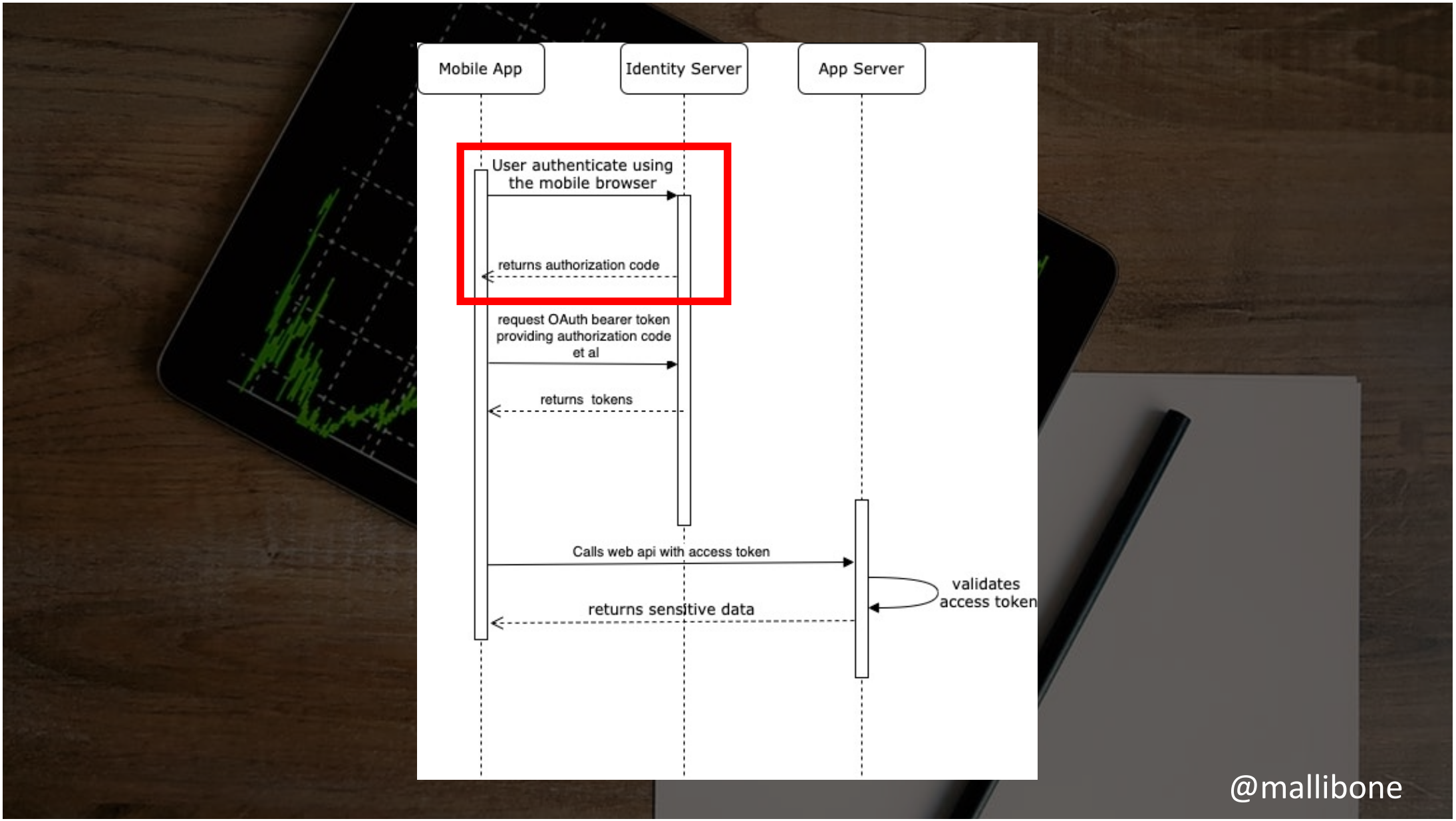


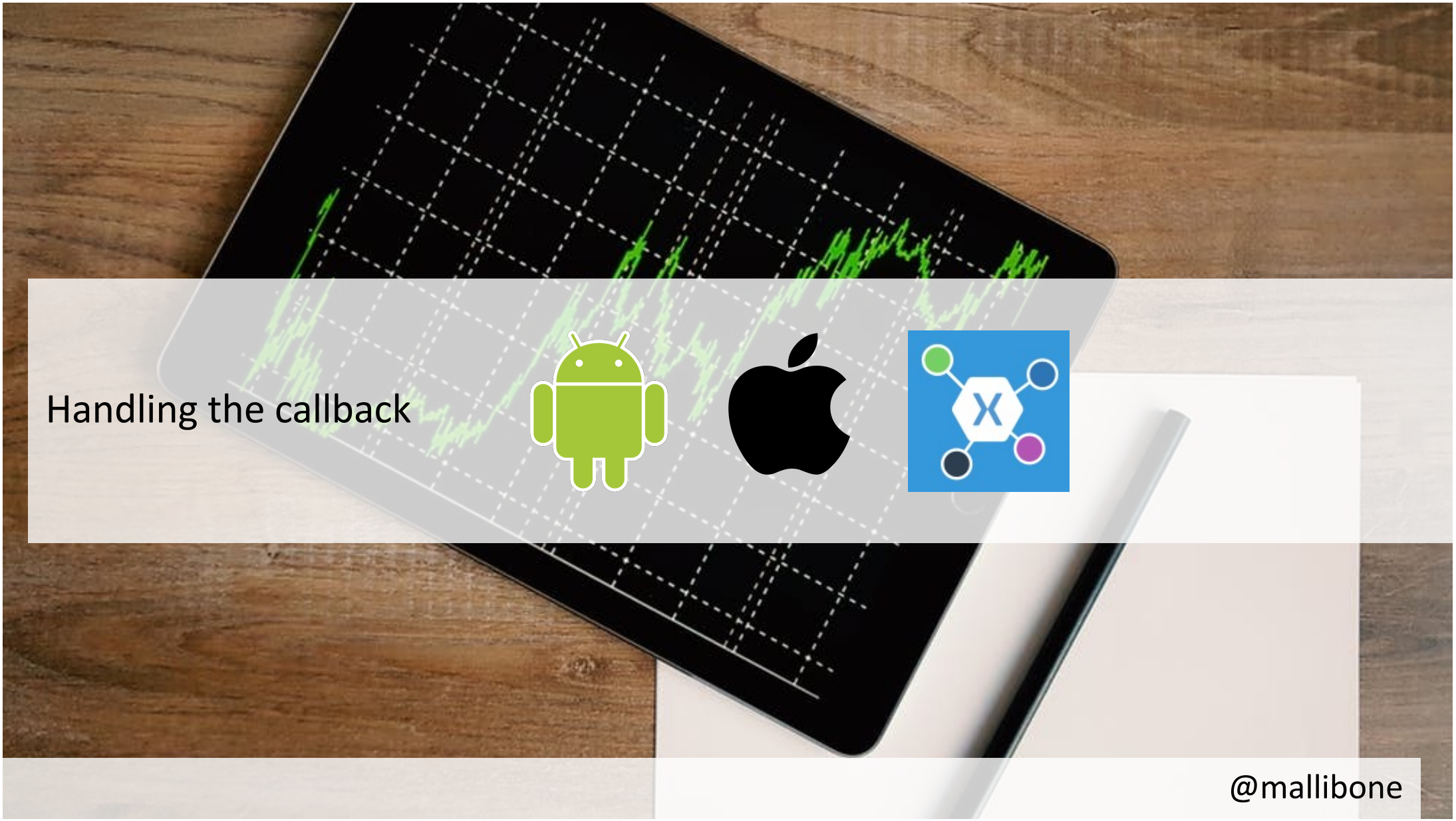
```
internal class WebAuthenticatorBrowser : IBrowser
{
    public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
    {
        try
        {
            var startUrl = new Uri(options.StartUrl);
            var callbackUrl = new Uri(options.EndUrl);
            WebAuthenticatorResult authResult =
                await WebAuthenticator
                    .AuthenticateAsync(startUrl, callbackUrl);
            var authorizeResponse =
                ToRawIdentityUrl(options.EndUrl, authResult);

            return new BrowserResult
            {
                Response = authorizeResponse
            };
        }
        catch (Exception ex)
        {
            Debug.WriteLine(ex);
            return new BrowserResult()
            {
                ResultType = BrowserResultType.UnknownError,
                Error = ex.ToString()
            };
        }
    }

    public string ToRawIdentityUrl(string redirectUrl,
WebAuthenticatorResult result)
    {
        IEnumerable<string> parameters =
            result.Properties
            .Select(pair => $"{pair.Key}={pair.Value}");
        var values = string.Join("&", parameters);

        return $"{redirectUrl}#{values}";
    }
}
```

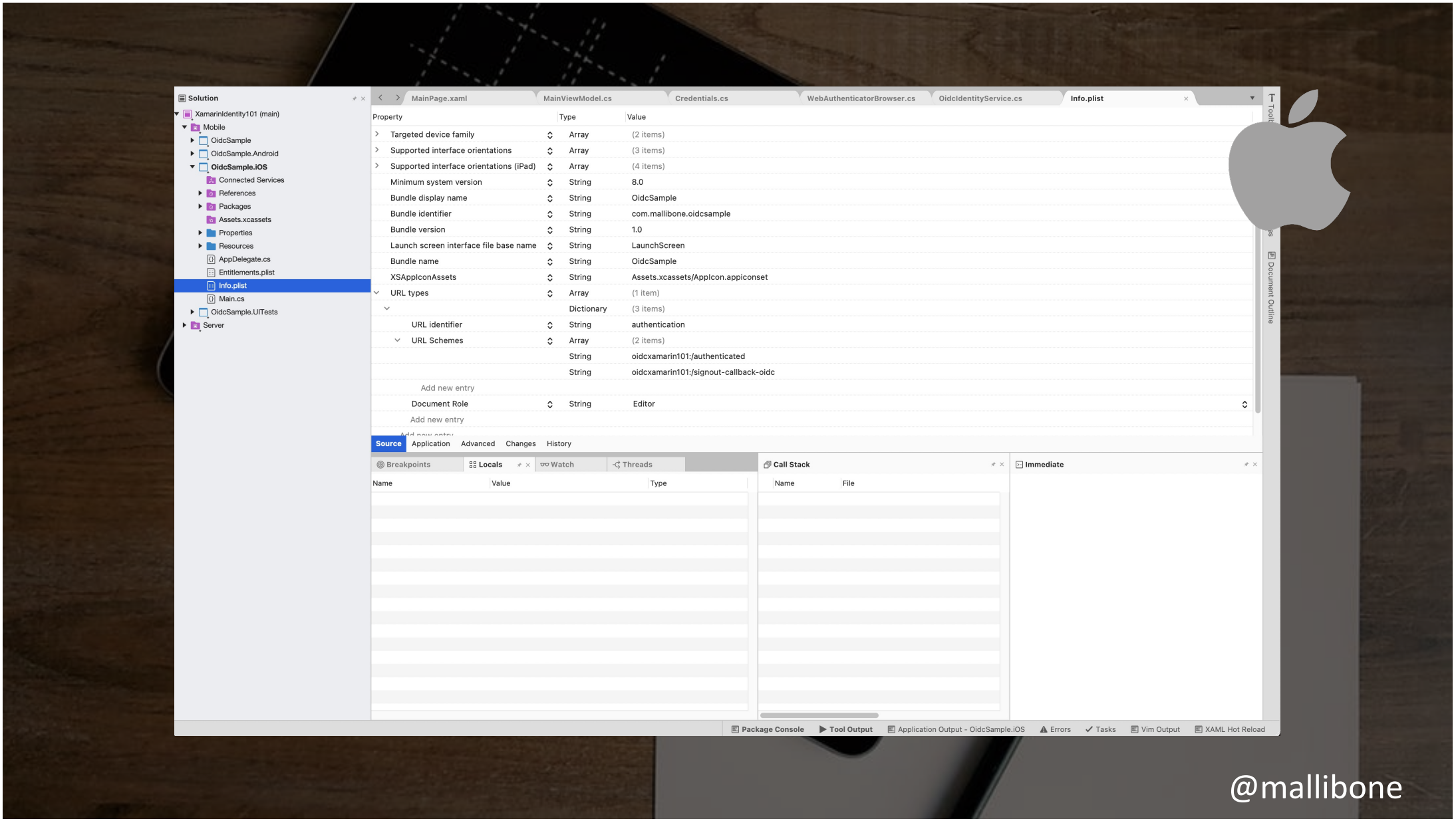

internal class WebAuthenticatorBrowser : IBrowser
{
public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
{
try
{
var startUrl = new Uri(options.StartUrl);
var callbackUrl = new Uri(options.EndUrl);
WebAuthenticatorResult authResult =
await WebAuthenticator
.AuthenticateAsync(startUrl, callbackUrl);
var authorizeResponse =
ToRawIdentityUrl(options.EndUrl, authResult);
return new BrowserResult
{
Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,


internal class WebAuthenticatorBrowser : IBrowser
{
public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
{
try
{
var startUrl = new Uri(options.StartUrl);
var callbackUrl = new Uri(options.EndUrl);
WebAuthenticatorResult authResult =
await WebAuthenticator
.AuthenticateAsync(startUrl, callbackUrl);
var authorizeResponse =
ToRawIdentityUrl(options.EndUrl, authResult);
return new BrowserResult
{
Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,

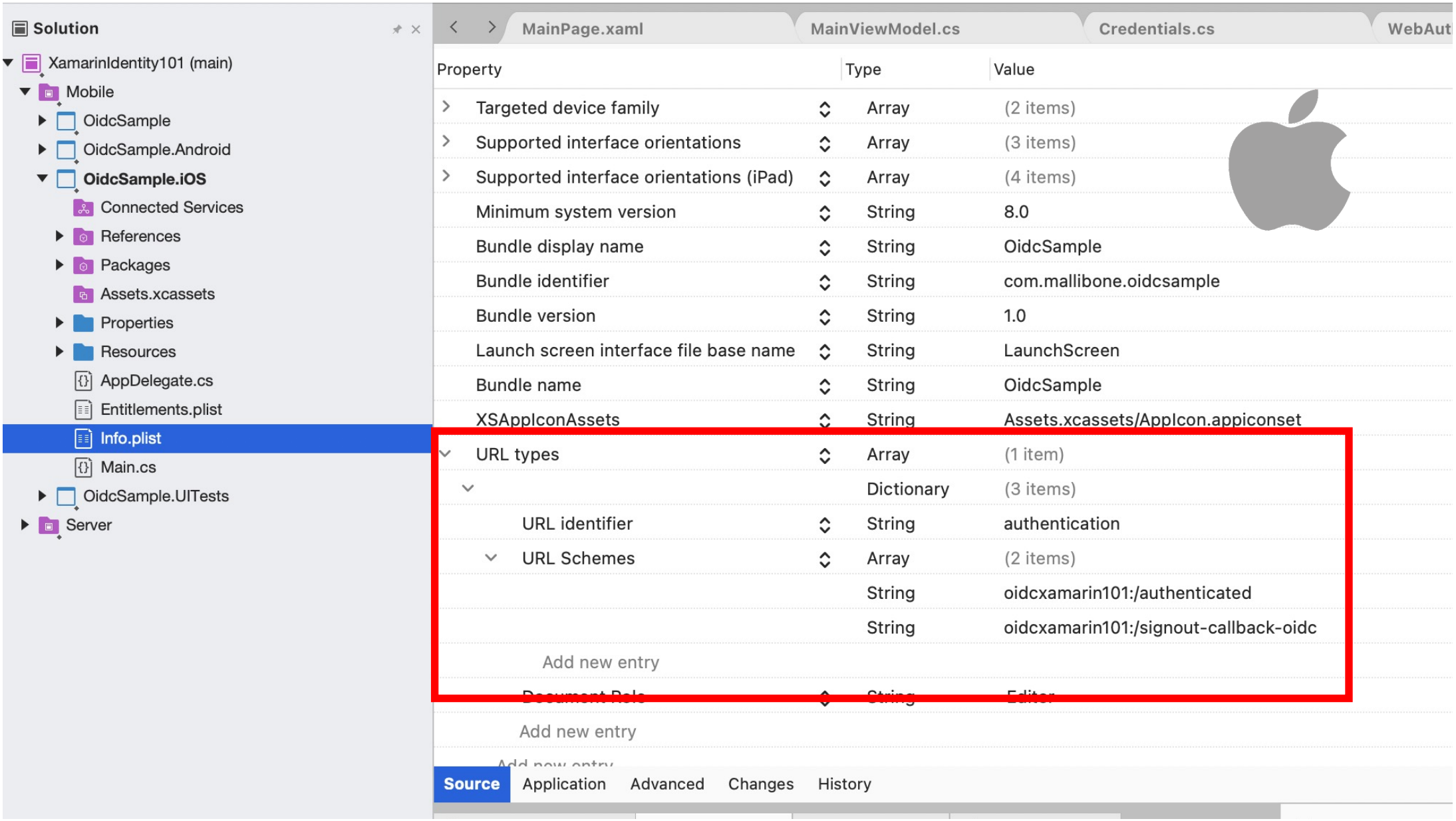

internal class WebAuthenticatorBrowser : IBrowser
{
public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
{
try
{
var startUrl = new Uri(options.StartUrl);
var callbackUrl = new Uri(options.EndUrl);
WebAuthenticatorResult authResult =
await WebAuthenticator
.AuthenticateAsync(startUrl, callbackUrl);
ToRawIdentityUrl(options.EndUrl, authResult);
return new BrowserResult
{
Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,
@mallibone

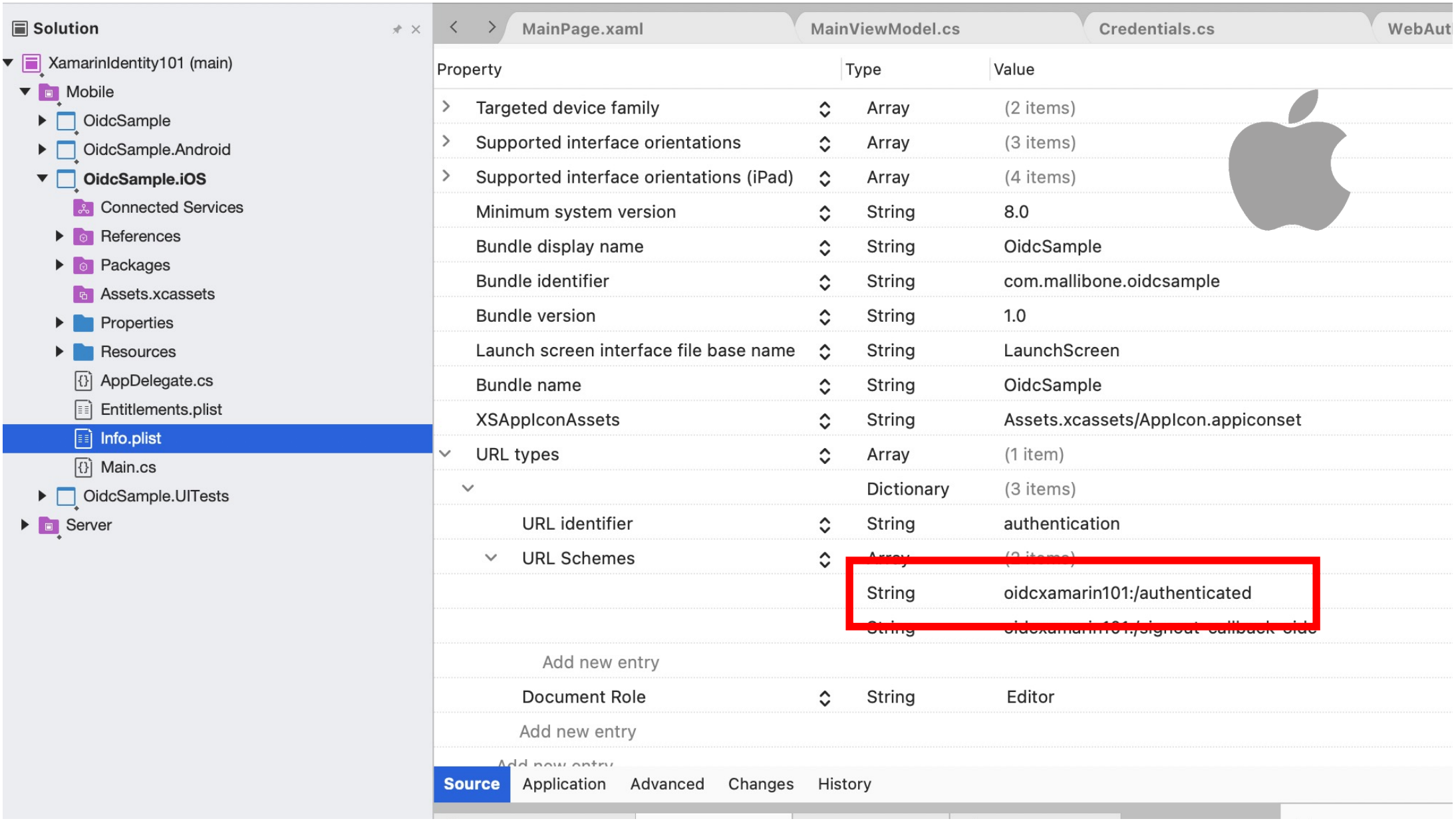

Mobile App
Identity Server
App Server
User authenticate using the mobile browser
returns authorization code
request OAuth bearer token providing authorization code et al
returns tokens
Calls web api with access token
validates access token
returns sensitive data
@mallibone

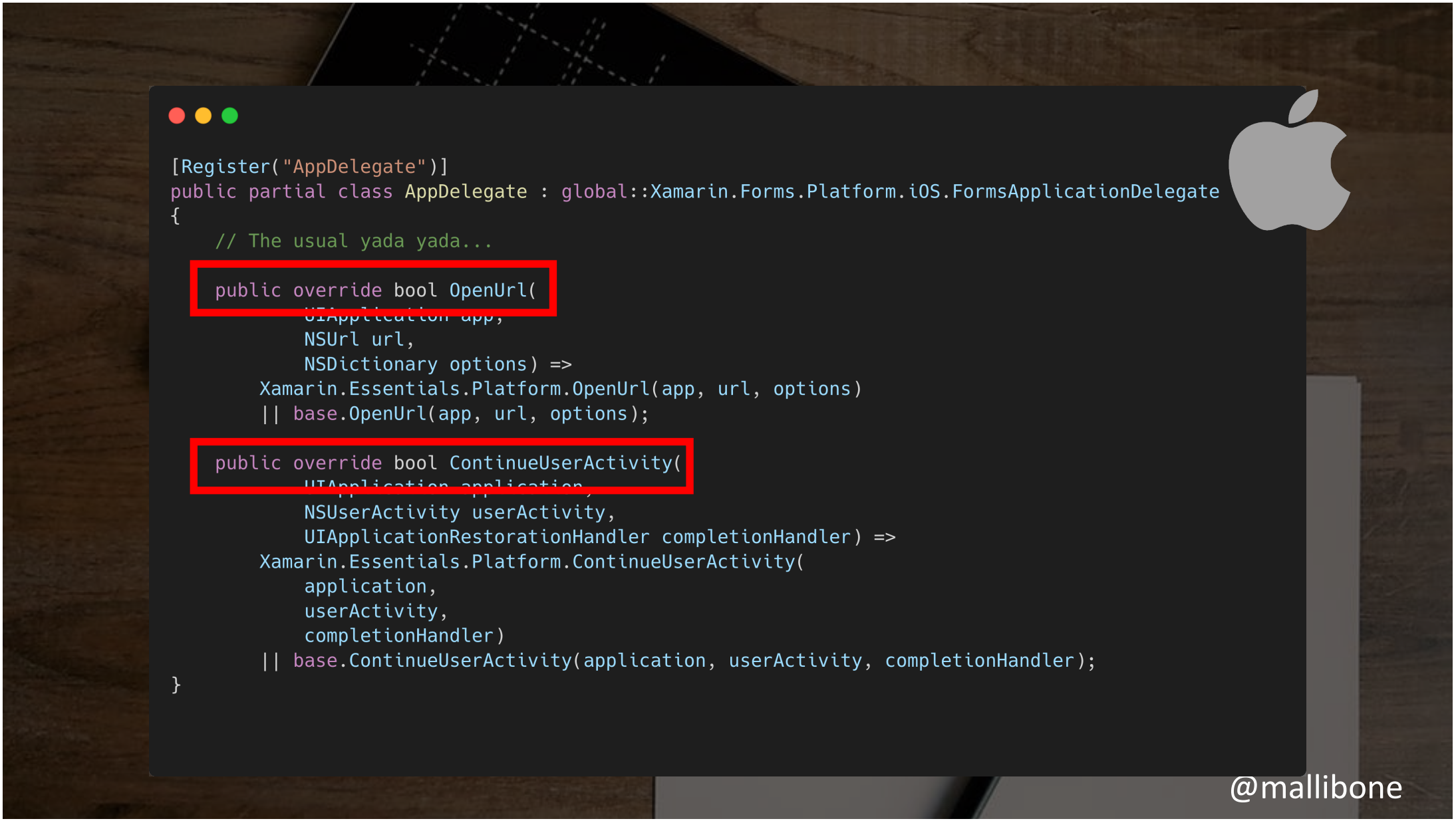

Handling the callback
@mallibone

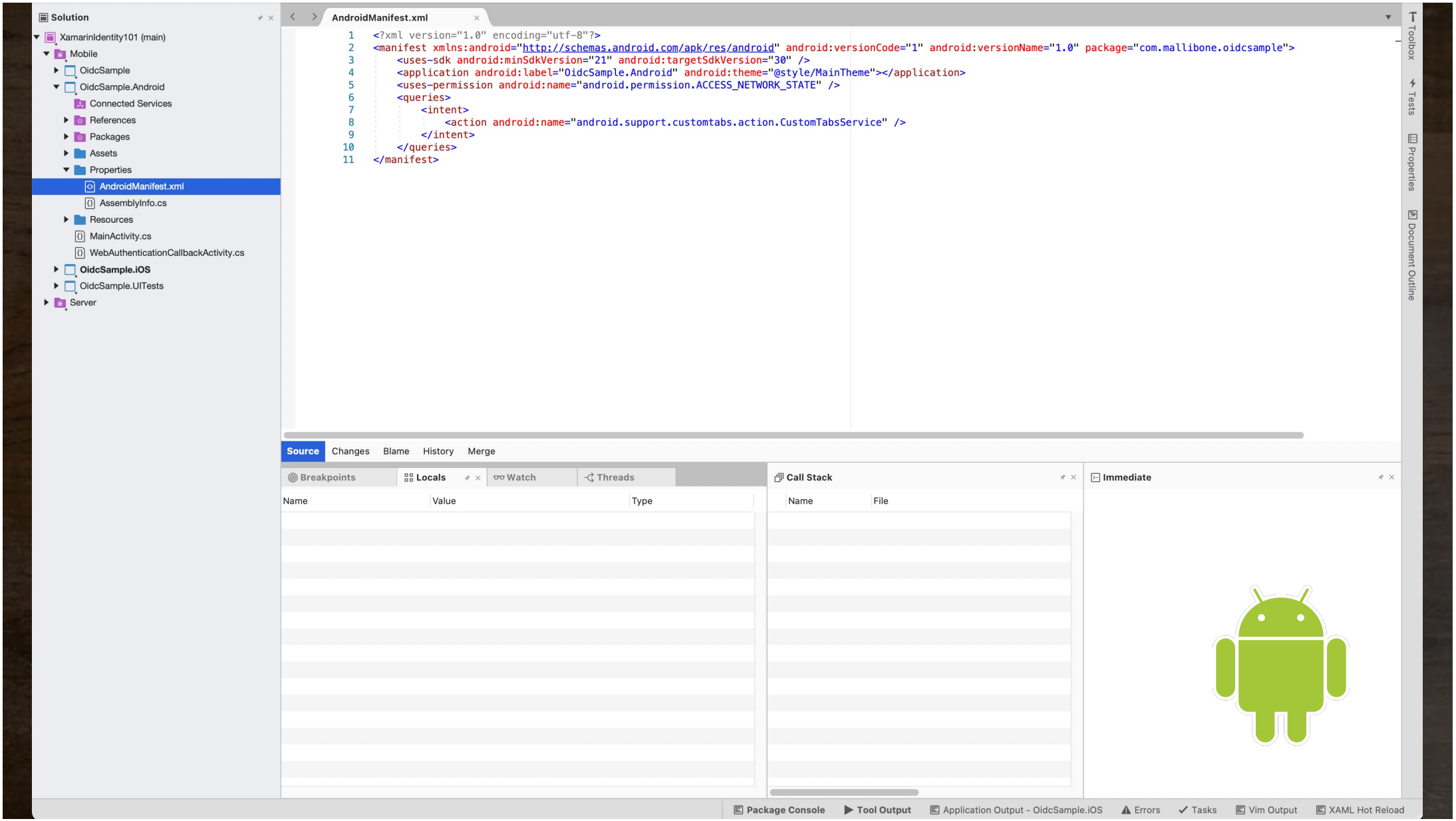

Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
OidcSample.iOS
Connected Services
References
Packages
Assets.xcassets
Properties
Resources
AppDelegate.cs
Entitlements.plist
Info.plist
Main.cs
OidcSample.UITests
Server
MainPage.xaml
MainViewModel.cs
Credentials.cs
WebAuthenticatorBrowser.cs
OidcIdentityService.cs
Info.plist
Property
Type
Value
Targeted device family
Array
(2 items)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
Minimum system version
String
8.0
Bundle display name
String
OidcSample
Bundle identifier
String
com.mallibone.oidcsample
Bundle version
String
1.0
Launch screen interface file base name
String
LaunchScreen
Bundle name
String
OidcSample
XSAppIconAssets
String
Assets.xcassets/AppIcon.appiconset
URL types
Array
(1 item)
Dictionary
(3 items)
URL identifier
String
authentication
URL Schemes
Array
(2 items)
String
oidcxamarin101:/authenticated
String
oidcxamarin101:/signout-callback-oidc
Add new entry
Document Role
String
Editor
Add new entry
Source
Application
Advanced
Changes
History
Breakpoints
Locals
Watch
Threads
Name
Value
Type
Call Stack
Name
File
Immediate
Package Console
Tool Output
Application Output - OidcSample.iOS
Errors
Tasks
Vim Output
XAML Hot Reload
Document Outline

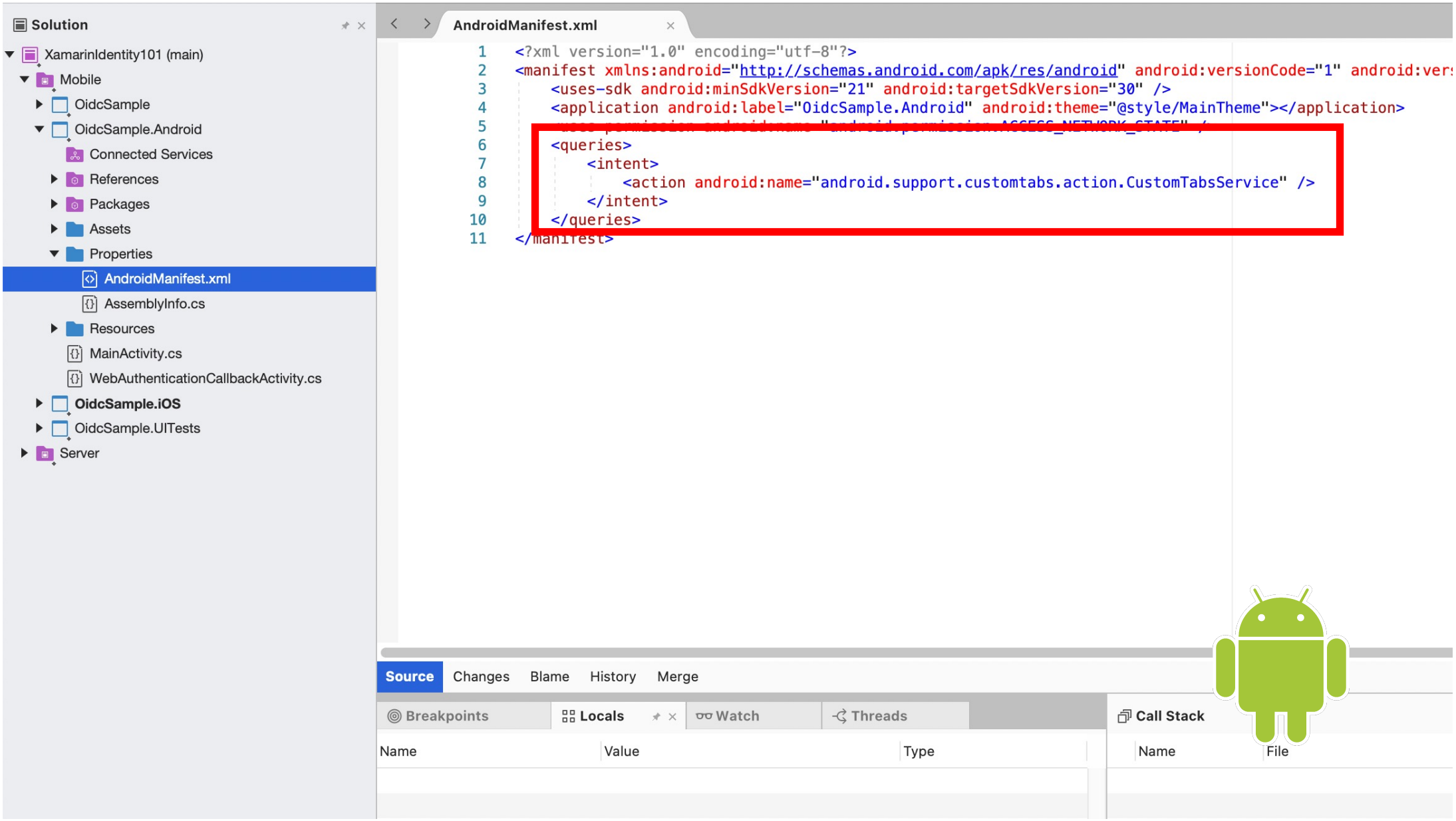
@mallibone

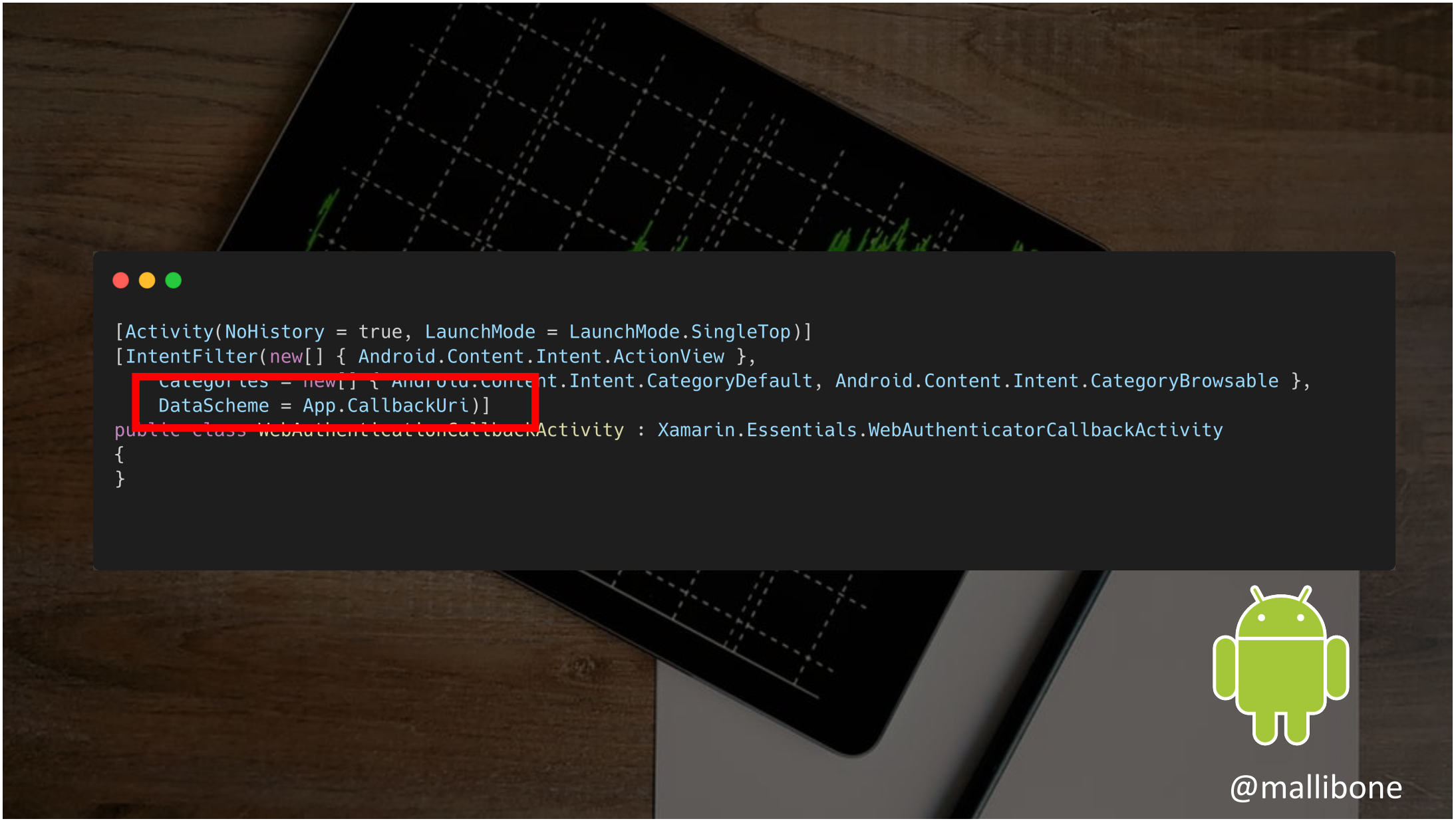

Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
OidcSample.iOS
Connected Services
References
Packages
Assets.xcassets
Properties
Resources
AppDelegate.cs
Entitlements.plist
Info.plist
Main.cs
OidcSample.UITests
Server
MainPage.xaml
MainViewModel.cs
Credentials.cs
WebAut
Property
Type
Value
Targeted device family
Array
(2 items)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
Minimum system version
String
8.0
Bundle display name
String
OidcSample
Bundle identifier
String
com.mallibone.oidcsample
Bundle version
String
1.0
Launch screen interface file base name
String
LaunchScreen
Bundle name
String
OidcSample
XSAppIconAssets
String
Assets.xcassets/AppIcon.appiconset
URL types
Array
(1 item)
Dictionary
(3 items)
URL identifier
String
authentication
URL Schemes
Array
(2 items)
String
oidcxamarin101:/authenticated
String
oidcxamarin101:/signout-callback-oidc
Add new entry
Add new entry
Source
Application
Advanced
Changes
History

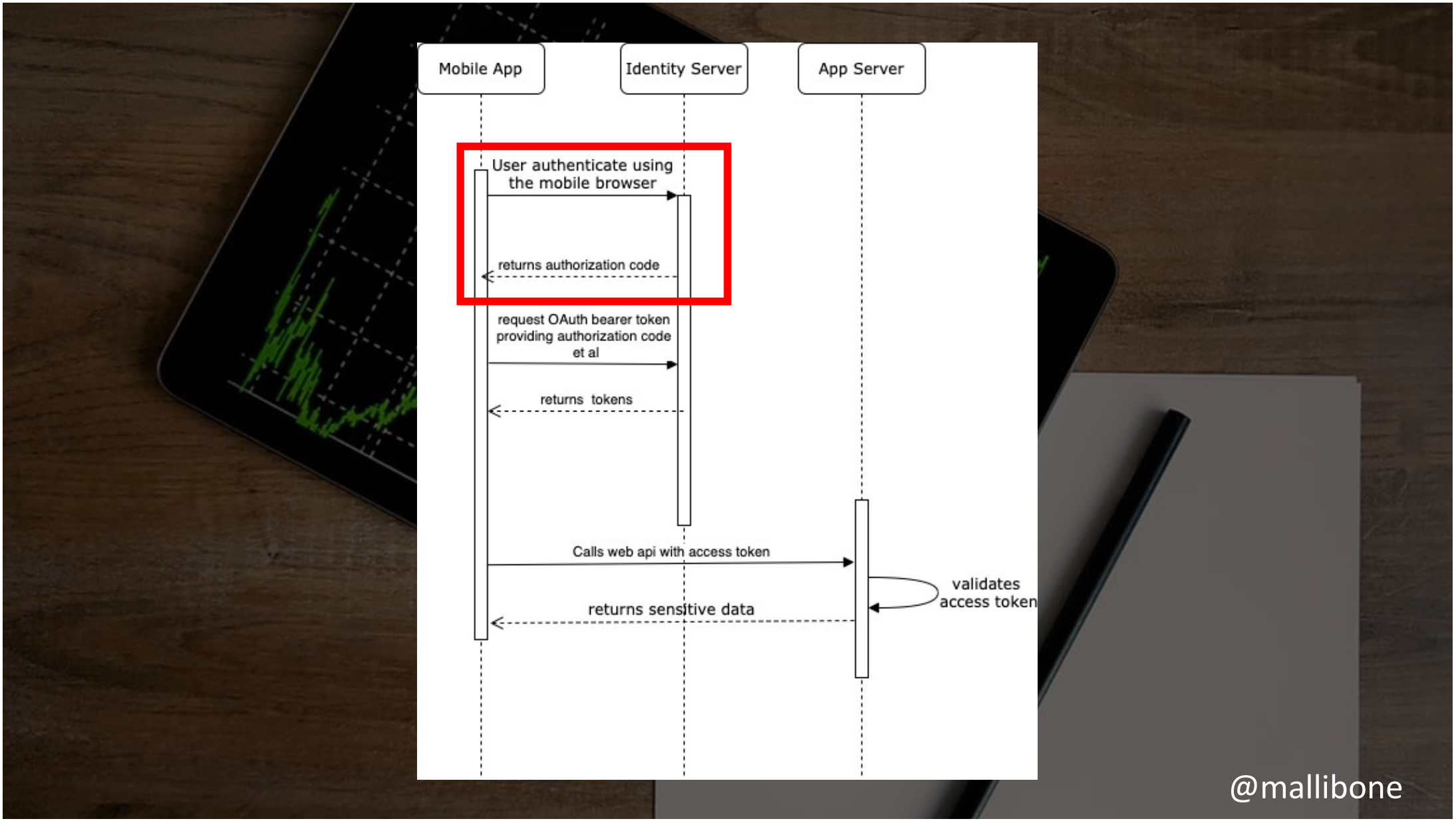

Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
OidcSample.iOS
Connected Services
References
Packages
Assets.xcassets
Properties
Resources
AppDelegate.cs
Entitlements.plist
Info.plist
Main.cs
OidcSample.UITests
Server
MainPage.xaml
MainViewModel.cs
Credentials.cs
Property
Type
Value
Targeted device family
Array
(2 items)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
Minimum system version
String
8.0
Bundle display name
String
OidcSample
Bundle identifier
String
com.mallibone.oidcsample
Bundle version
String
1.0
Launch screen interface file base name
String
LaunchScreen
Bundle name
String
OidcSample
XSAppIconAssets
String
Assets.xcassets/AppIcon.appiconset
URL types
Array
(1 item)
Dictionary
(3 items)
URL identifier
String
authentication
URL Schemes
String
oidcxamarin101:/authenticated
Add new entry
Document Role
String
Editor
Add new entry
Source
Application
Advanced
Changes
History

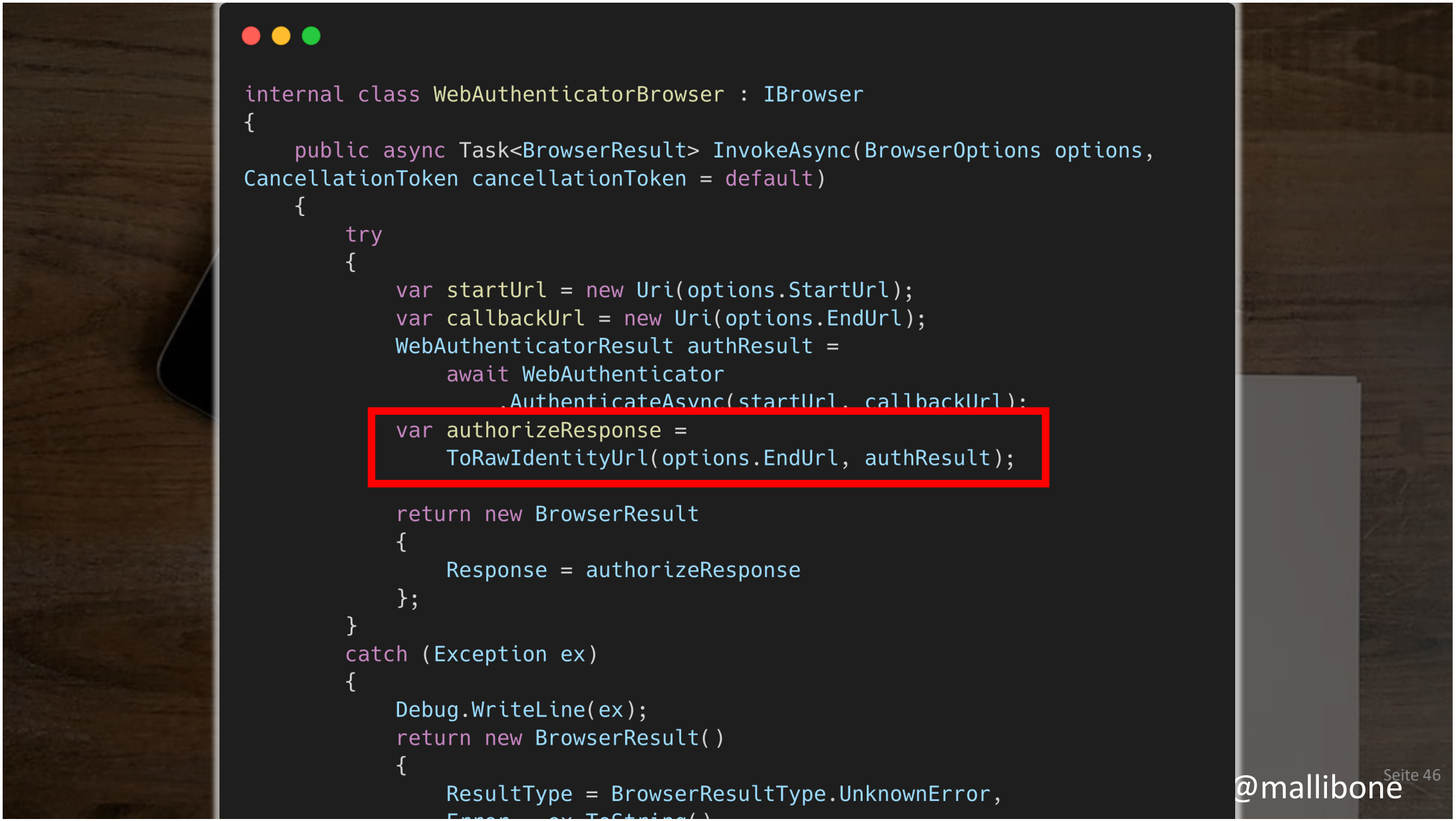

[Register("AppDelegate")]
public partial class AppDelegate : global::Xamarin.Forms.Platform.iOS.FormsApplicationDelegate
{
// The usual yada yada...
public override bool OpenUrl(
UIApplication app,
NSUrl url,
NSDictionary options) =>
Xamarin.Essentials.Platform.OpenUrl(app, url, options)
|| base.OpenUrl(app, url, options);
public override bool ContinueUserActivity(
UIApplication application,
NSUserActivity userActivity,
UIApplicationRestorationHandler completionHandler) =>
Xamarin.Essentials.Platform.ContinueUserActivity(
application,
userActivity,
completionHandler)
|| base.ContinueUserActivity(application, userActivity, completionHandler);
}
@mallibone


Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
Connected Services
References
Packages
Assets
Properties
AndroidManifest.xml
AssemblyInfo.cs
Resources
MainActivity.cs
WebAuthenticationCallbackActivity.cs
OidcSample.iOS
OidcSample.UITests
Server
AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" package="com.mallibone.oidcsample">
3 <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="30" />
4 <application android:label="OidcSample.Android" android:theme="@style/MainTheme"></application>
5 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
6 <queries>
7 <intent>
8 <action android:name="android.support.customtabs.action.CustomTabsService" />
9 </intent>
10 </queries>
11 </manifest>
Source
Changes
Blame
History
Merge
Breakpoints
Locals
Watch
Threads
Name
Value
Type
Call Stack
Name
File
Immediate
Toolbox
Tests
Properties
Document Outline
Package Console
Tool Output
Application Output - OidcSample.iOS
Errors
Tasks
Vim Output
XAML Hot Reload

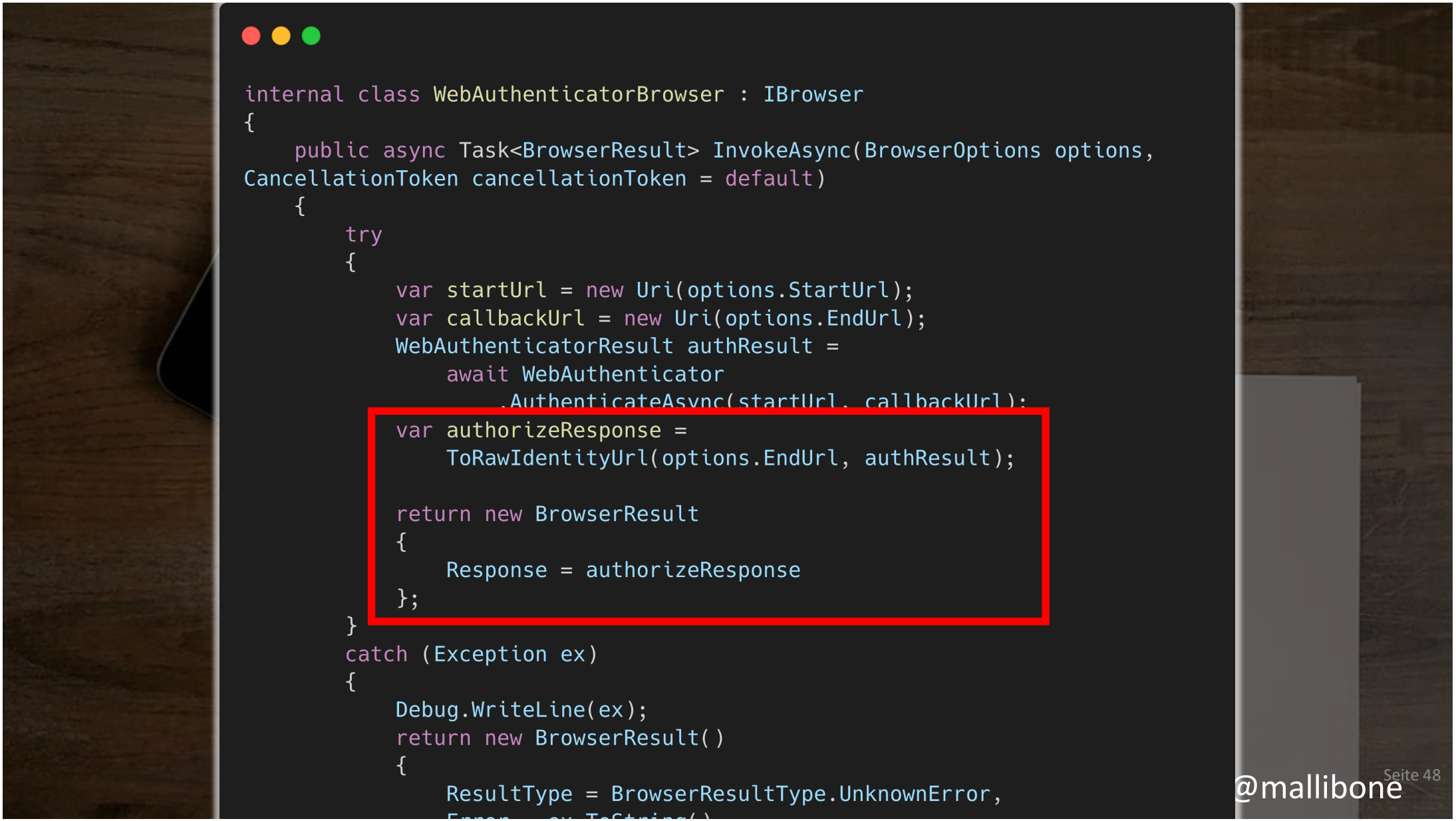

Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
Connected Services
References
Packages
Assets
Properties
AndroidManifest.xml
AssemblyInfo.cs
Resources
MainActivity.cs
WebAuthenticationCallbackActivity.cs
OidcSample.iOS
OidcSample.UITests
Server
AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:vers
3 <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="30" />
4 <application android:label="OidcSample.Android" android:theme="@style/MainTheme"></application>
6 <queries>
7 <intent>
8 <action android:name="android.support.customtabs.action.CustomTabsService" />
9 </intent>
10 </queries>
11 </manifest>
Source
Changes
Blame
History
Merge
Breakpoints
Locals
Watch
Threads
Call Stack
Name
Value
Type
Name
File

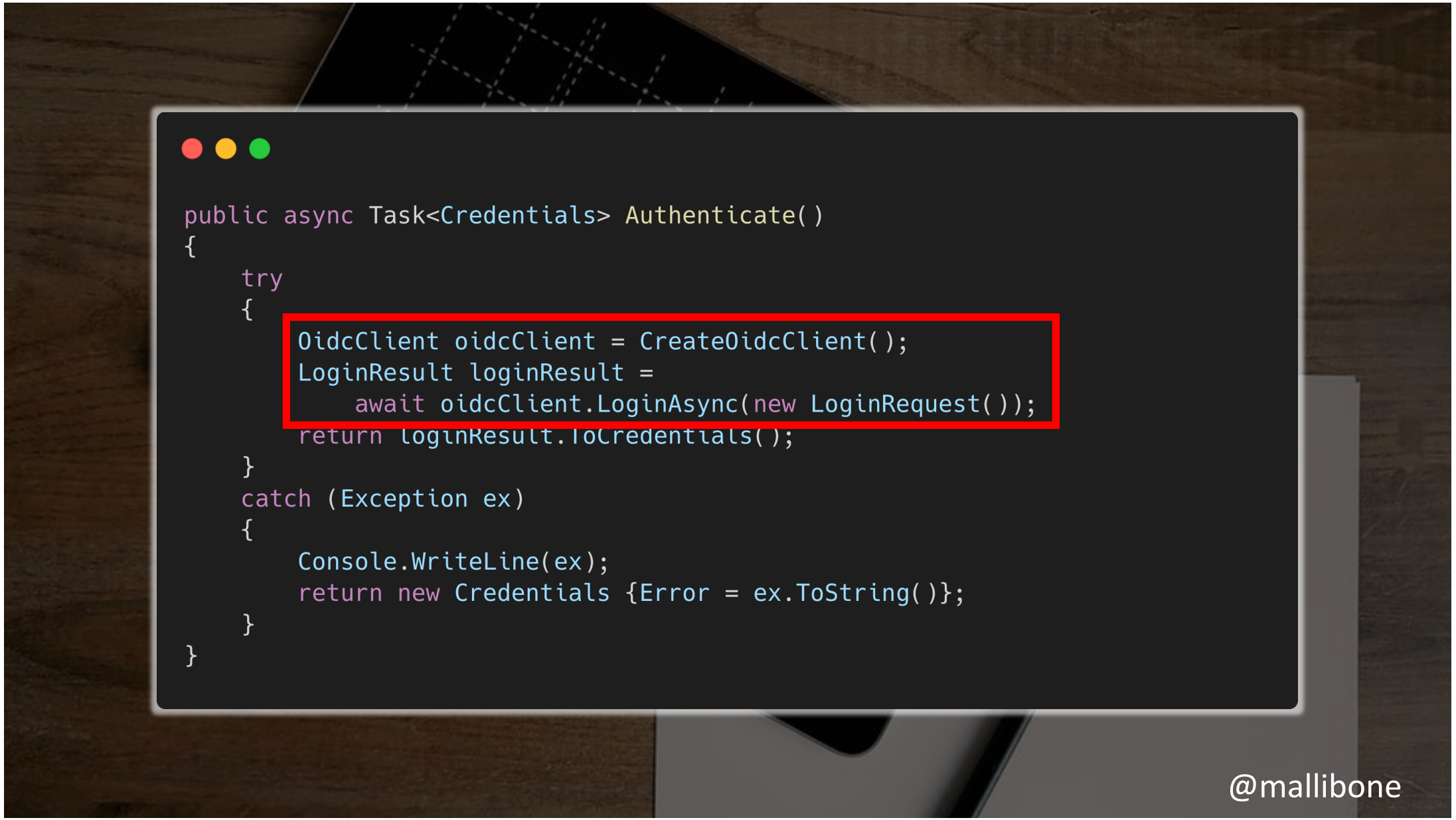

[Activity(NoHistory = true, LaunchMode = LaunchMode.SingleTop)]
[IntentFilter(new[] { Android.Content.Intent.ActionView },
Android.Content.Intent.CategoryDefault, Android.Content.Intent.CategoryBrowsable },
DataScheme = App.CallbackUri)]
Activity : Xamarin.Essentials.WebAuthenticatorCallbackActivity
{
}
@mallibone

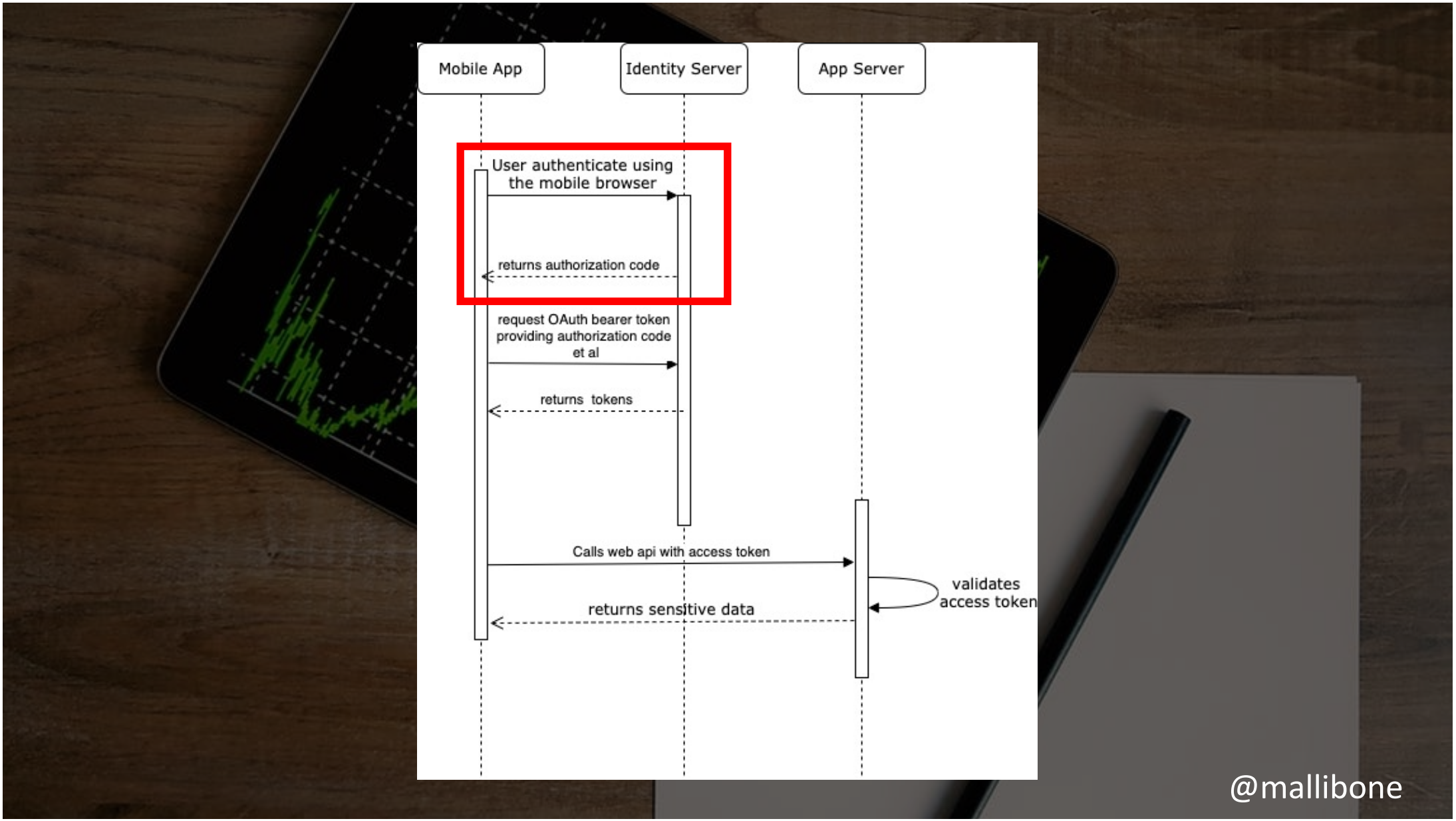

Mobile App
Identity Server
App Server
User authenticate using the mobile browser
returns authorization code
request OAuth bearer token providing authorization code et al
returns tokens
Calls web api with access token
validates access token
returns sensitive data
@mallibone

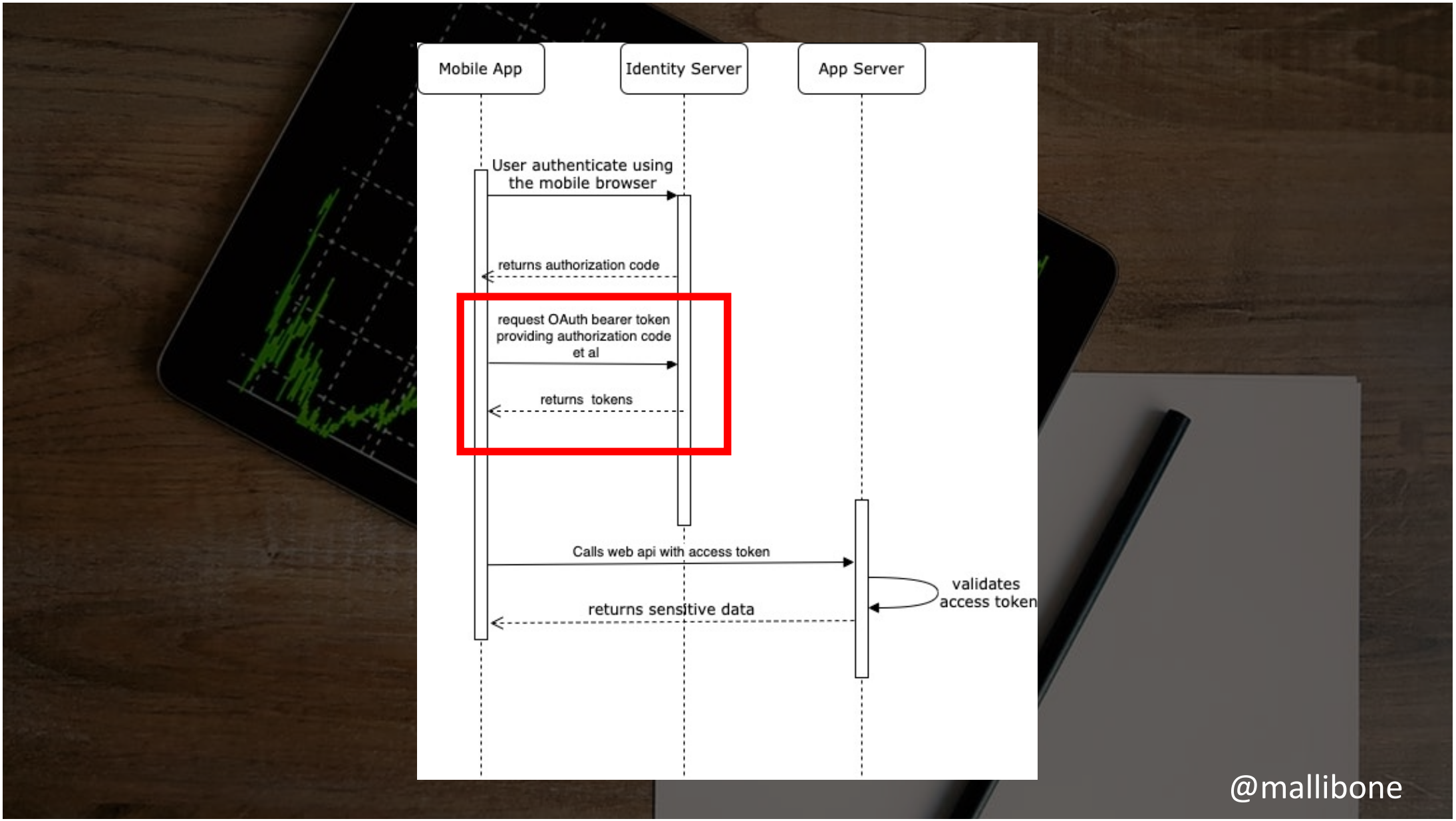

internal class WebAuthenticatorBrowser : IBrowser
{
public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
{
try
{
var startUrl = new Uri(options.StartUrl);
var callbackUrl = new Uri(options.EndUrl);
WebAuthenticatorResult authResult =
await WebAuthenticator
.AuthenticateAsync(startUrl, callbackUrl);
var authorizeResponse =
ToRawIdentityUrl(options.EndUrl, authResult);
return new BrowserResult
{
Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,

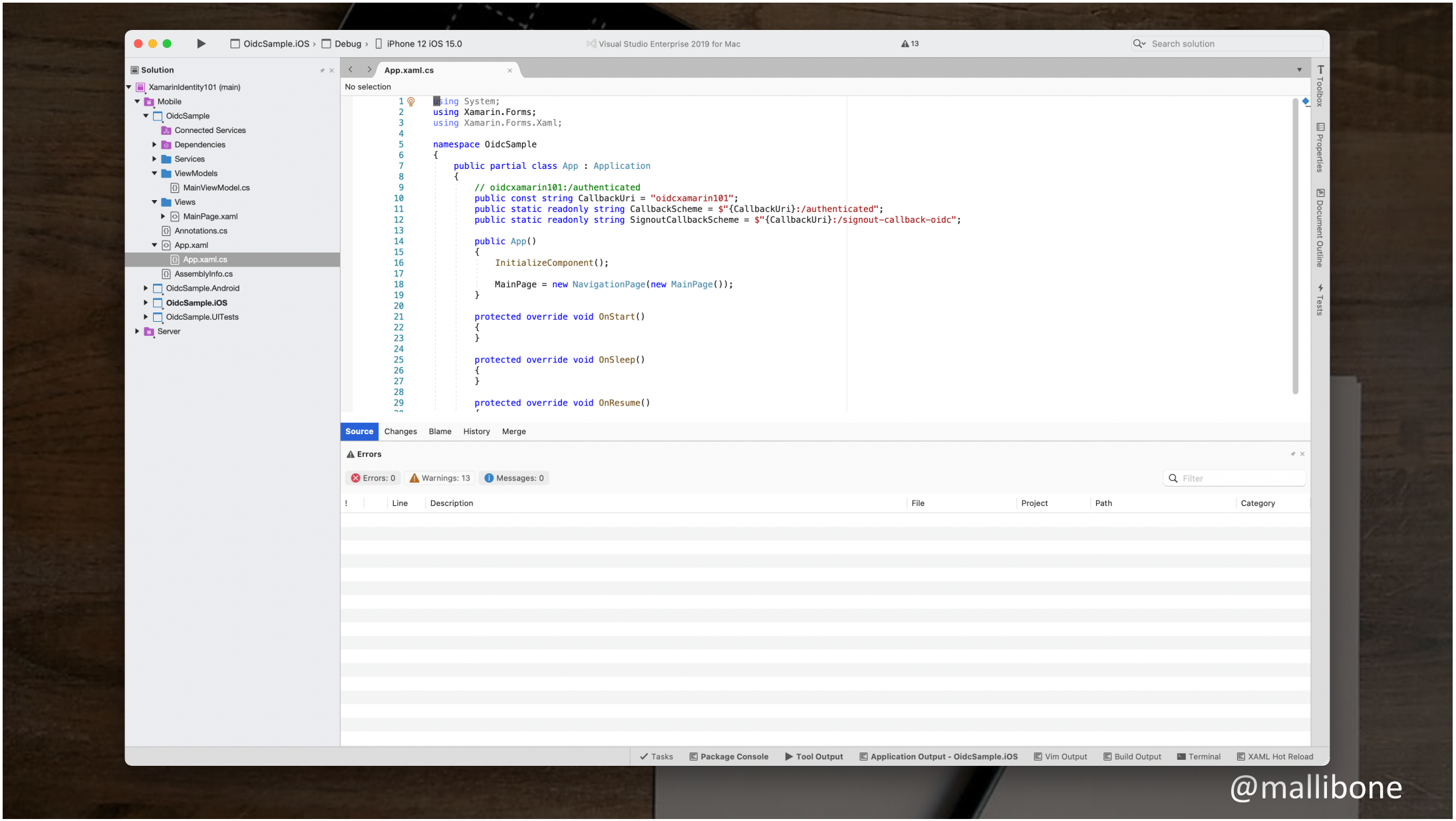

Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,
Error = ex.ToString()
};
}
}
public string ToRawIdentityUrl(string redirectUrl,
WebAuthenticatorResult result)
{
IEnumerable<string> parameters =
result.Properties
.Select(pair => $"{pair.Key}={pair.Value}");
var values = string.Join("&", parameters);
return $"{redirectUrl}#{values}";
}
}
@mallibone


internal class WebAuthenticatorBrowser : IBrowser
{
public async Task<BrowserResult> InvokeAsync(BrowserOptions options,
CancellationToken cancellationToken = default)
{
try
{
var startUrl = new Uri(options.StartUrl);
var callbackUrl = new Uri(options.EndUrl);
WebAuthenticatorResult authResult =
await WebAuthenticator
.AuthenticateAsync(startUrl, callbackUrl);
var authorizeResponse =
ToRawIdentityUrl(options.EndUrl, authResult);
return new BrowserResult
{
Response = authorizeResponse
};
}
catch (Exception ex)
{
Debug.WriteLine(ex);
return new BrowserResult()
{
ResultType = BrowserResultType.UnknownError,

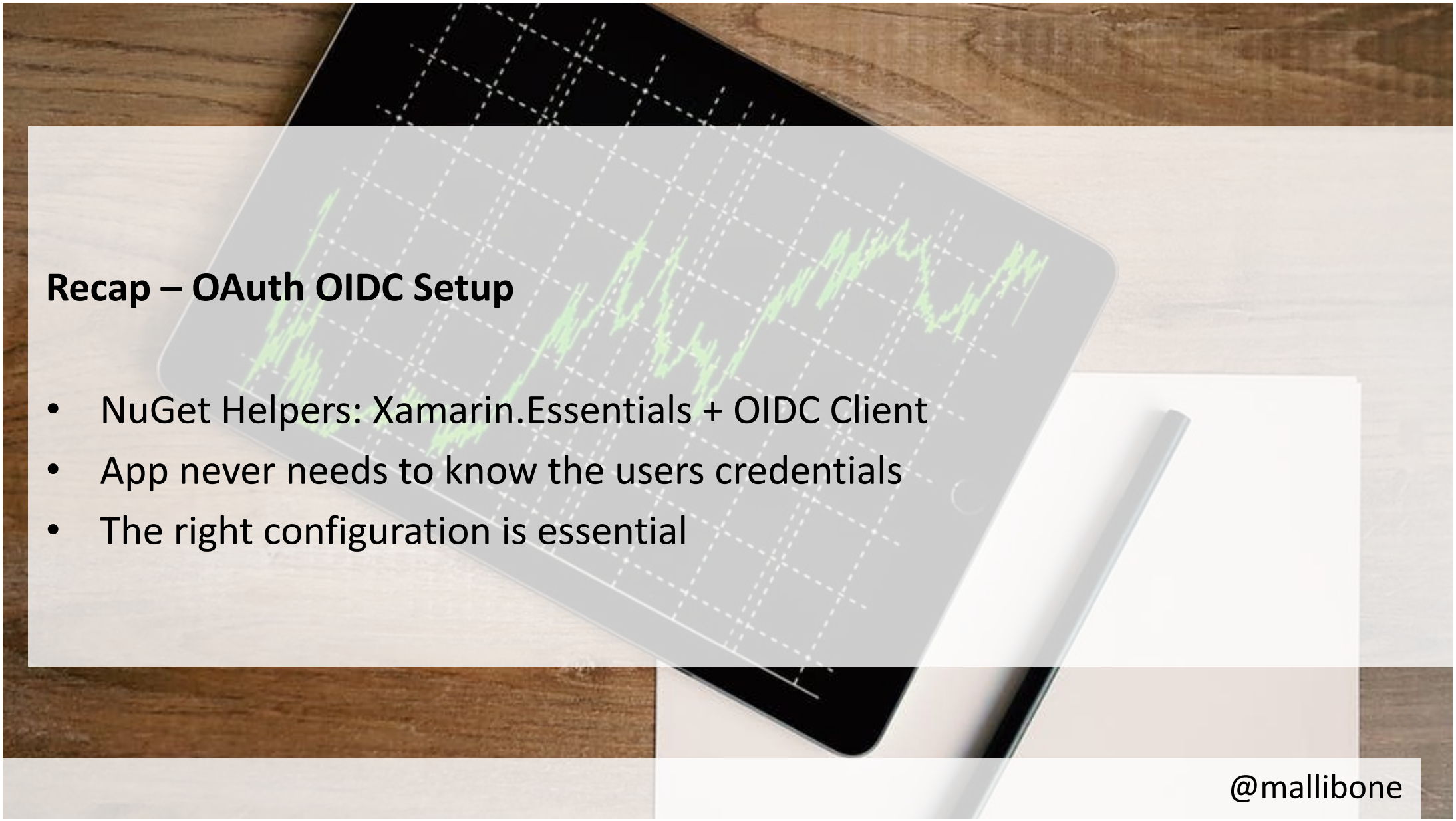

public async Task<Credentials> Authenticate()
{
try
{
OidcClient oidcClient = CreateOidcClient();
LoginResult loginResult =
await oidcClient.LoginAsync(new LoginRequest());
return loginResult.ToCredentials();
}
catch (Exception ex)
{
Console.WriteLine(ex);
return new Credentials {Error = ex.ToString()};
}
}
@mallibone


Mobile App
Identity Server
App Server
User authenticate using the mobile browser
returns authorization code
request OAuth bearer token providing authorization code et al
returns tokens
Calls web api with access token
validates access token
returns sensitive data
@mallibone


Mobile App
Identity Server
App Server
User authenticate using the mobile browser
returns authorization code
request OAuth bearer token providing authorization code et al
returns tokens
Calls web api with access token
validates access token
returns sensitive data
@mallibone


OidcSample.iOS › Debug › iPhone 12 iOS 15.0
Visual Studio Enterprise 2019 for Mac
13
Search solution
Solution
XamarinIdentity101 (main)
Mobile
OidcSample
Connected Services
Dependencies
Services
ViewModels
MainViewModel.cs
Views
MainPage.xaml
Annotations.cs
App.xaml
App.xaml.cs
AssemblyInfo.cs
OidcSample.Android
OidcSample.iOS
OidcSample.UITests
Server
App.xaml.cs
No selection
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;
namespace OidcSample
{
public partial class App : Application
{
// oidcxamarin101:/authenticated
public const string CallbackUri = "oidcxamarin101";
public static readonly string CallbackScheme = $"{CallbackUri}:/authenticated";
public static readonly string SignoutCallbackScheme = $"{CallbackUri}:/signout-callback-oidc";
public App()
{
InitializeComponent();
MainPage = new NavigationPage(new MainPage());
}
protected override void OnStart()
{
}
protected override void OnSleep()
{
}
protected override void OnResume()
Source
Changes
Blame
History
Merge
Errors
Errors: 0
Warnings: 13
Messages: 0
Filter
Line
Description
File
Project
Path
Category
Tasks
Package Console
Tool Output
Application Output - OidcSample.iOS
Vim Output
Build Output
Terminal
XAML Hot Reload
Toolbox
Properties
Document Outline
Tests

@mallibone



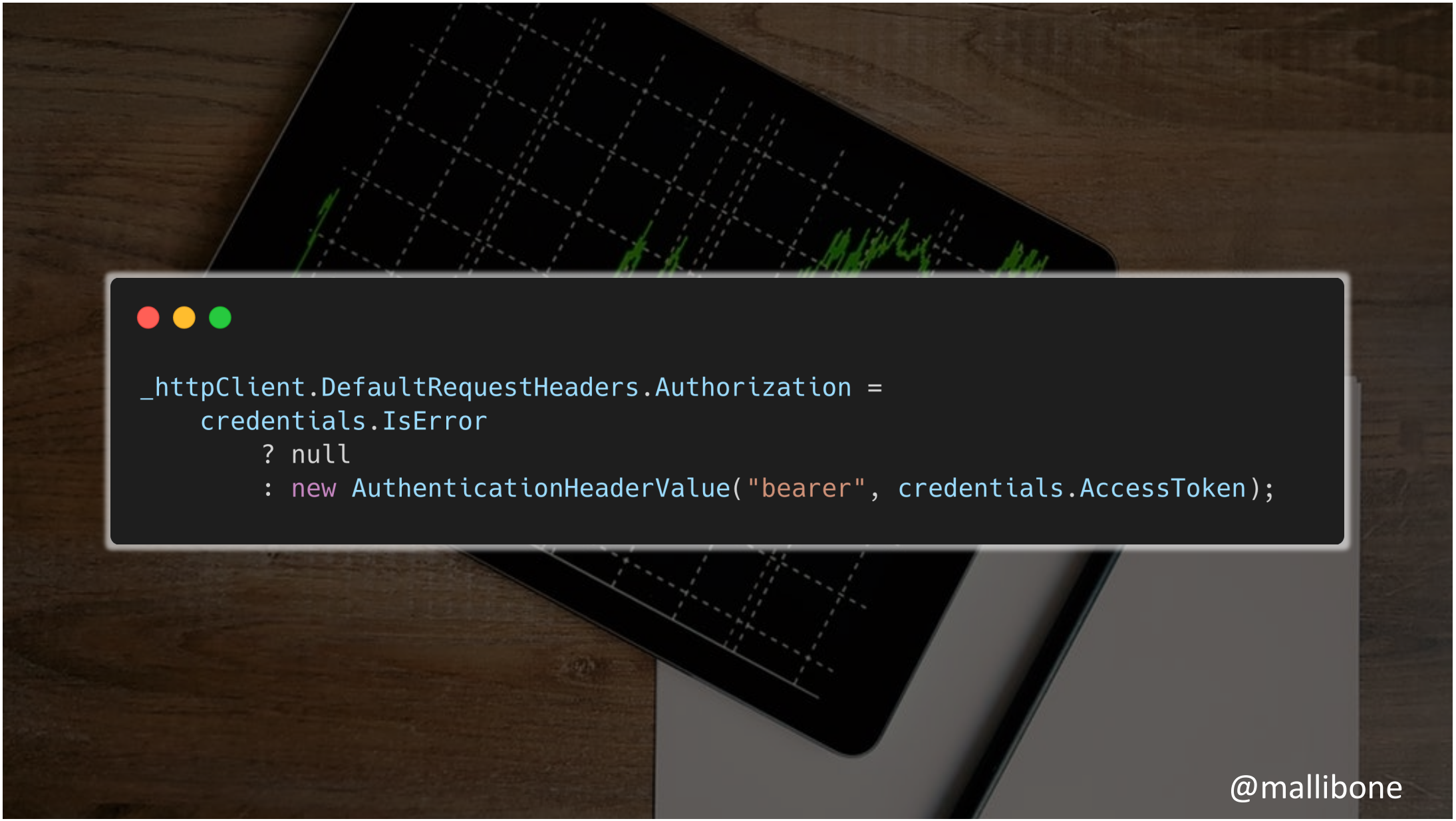
Recap – OAuth OIDC Setup

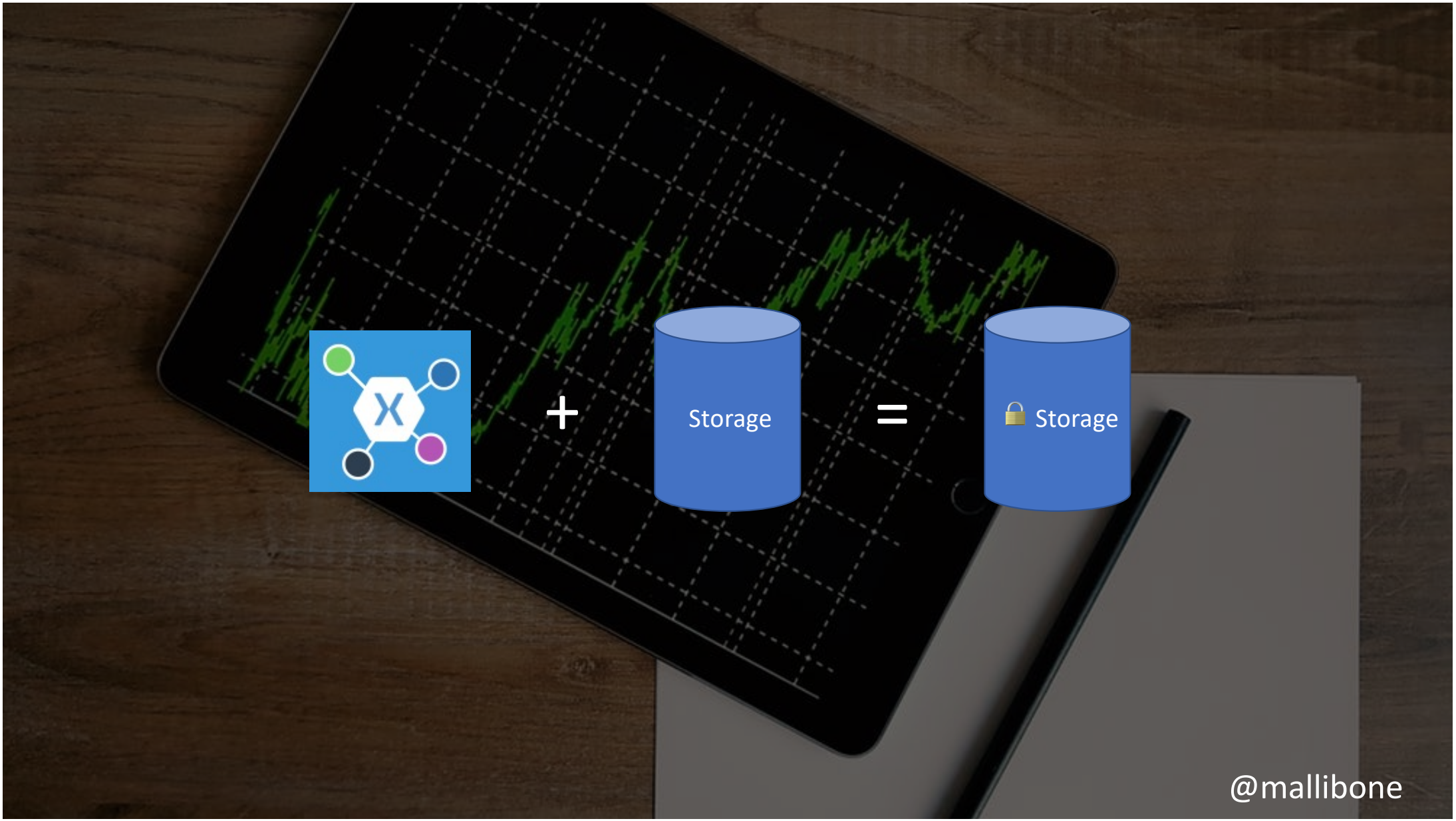
- NuGet Helpers: Xamarin.Essentials + OIDC Client
- App never needs to know the users credentials
- The right configuration is essential

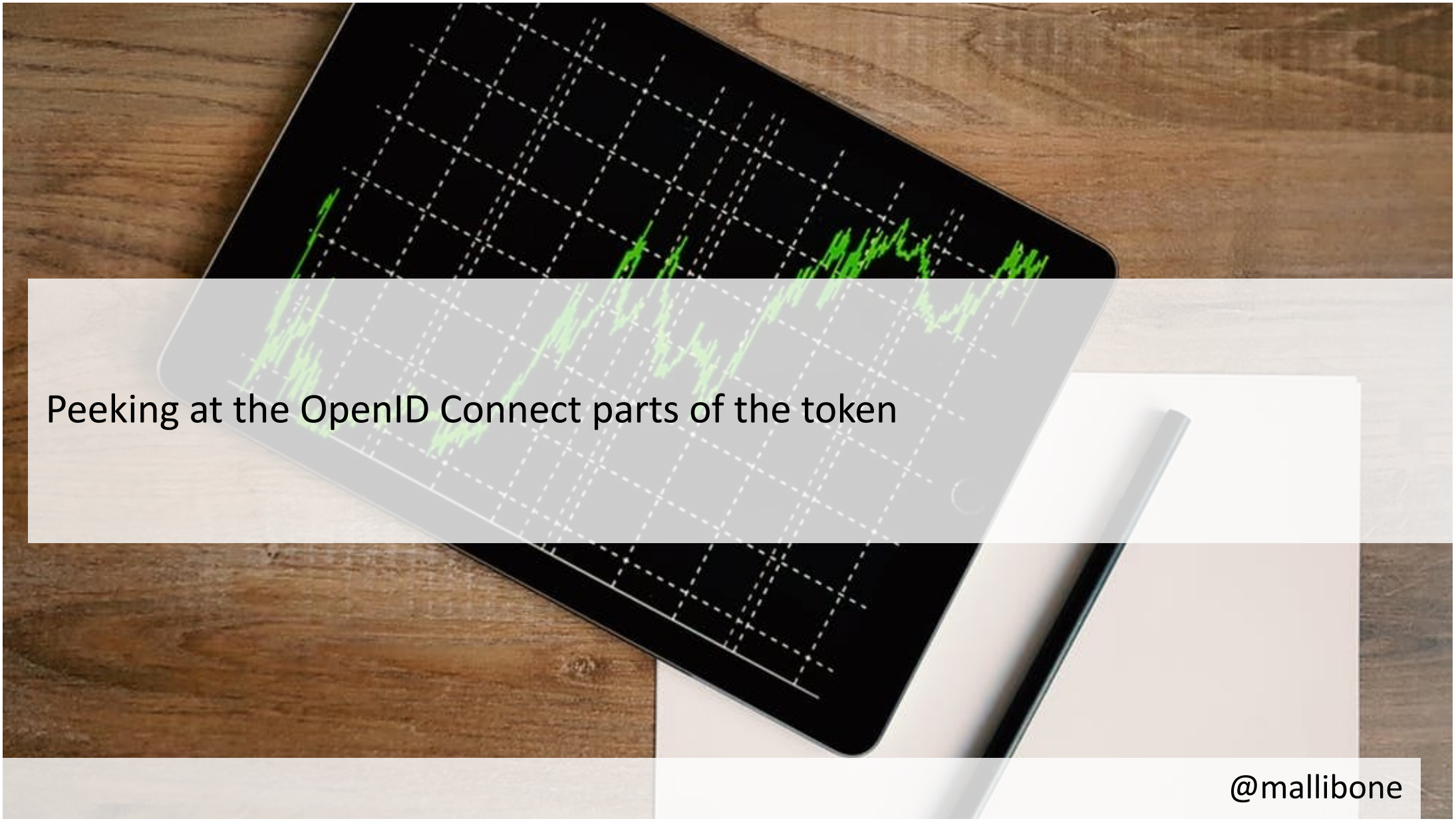

Me thinking the config was correct
The Identity Service
imgflip.com

@mallibone


SO WE GOT THE TOKENS USING A CODE
NOW WHAT?
imgflip.com
@mallibone

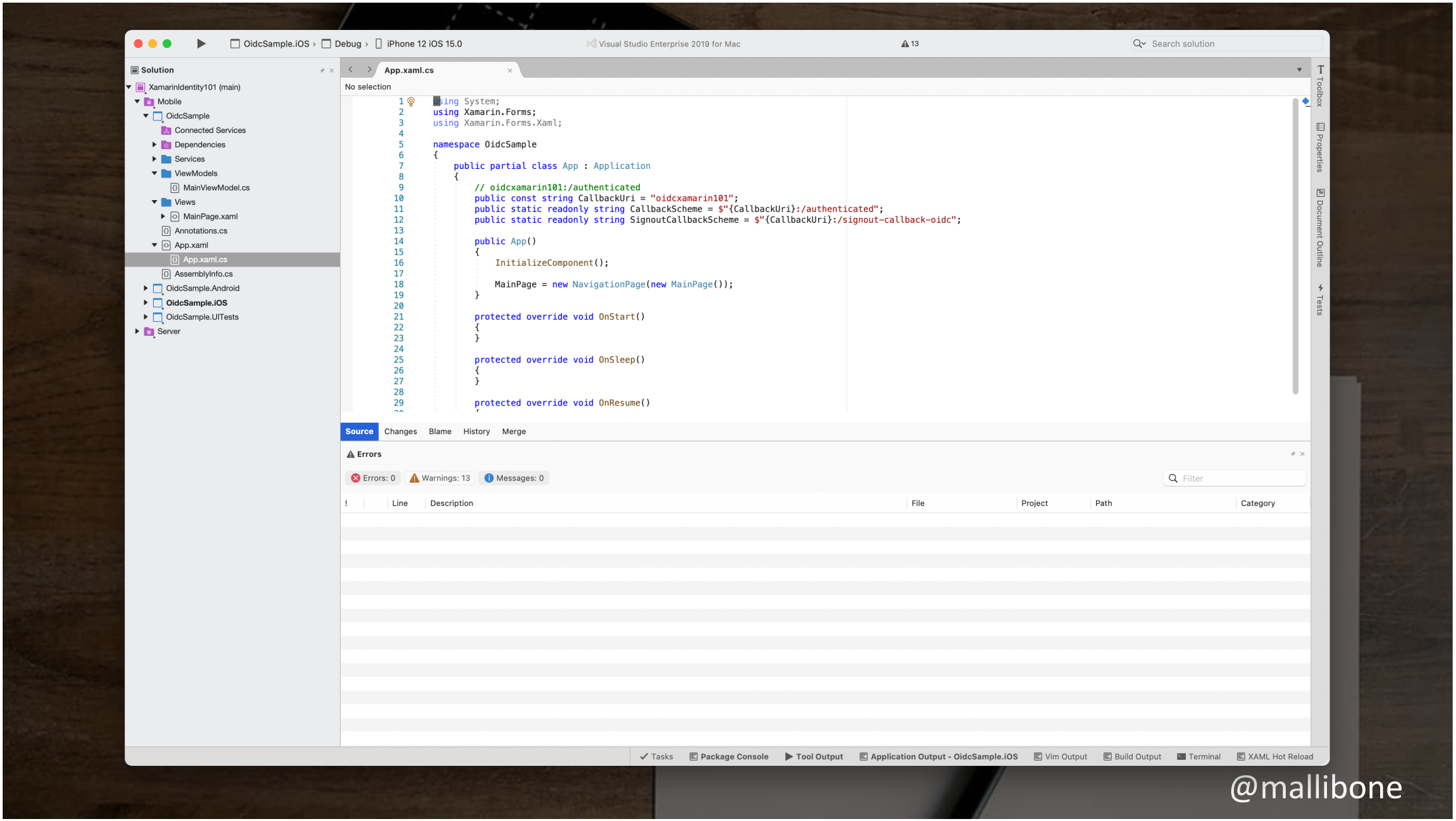

_httpClient.DefaultRequestHeaders.Authorization =
credentials.IsError
? null
: new AuthenticationHeaderValue("bearer", credentials.AccessToken);
@mallibone


Storage
Storage
@mallibone

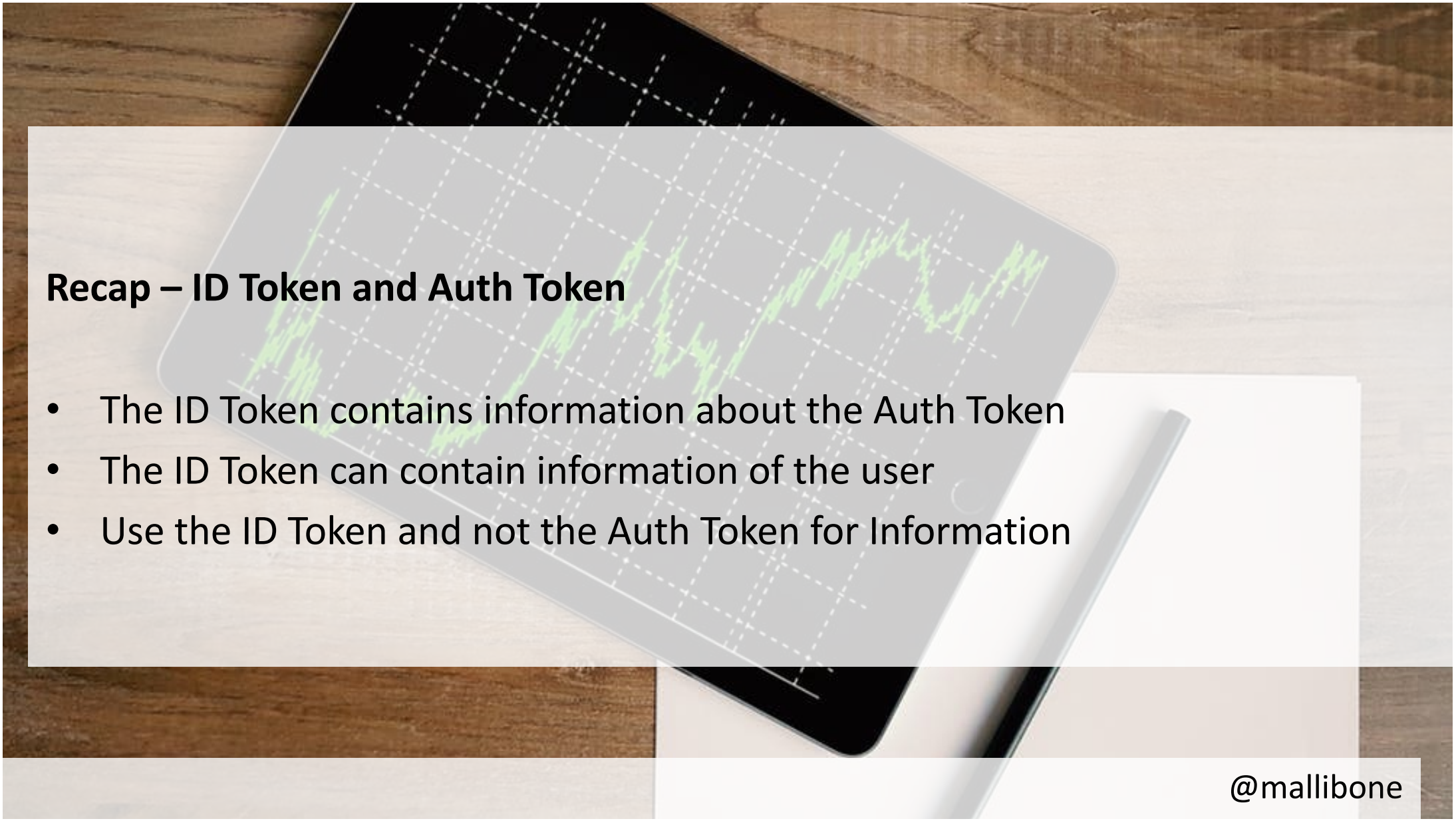


Peeking at the OpenID Connect parts of the token

@mallibone

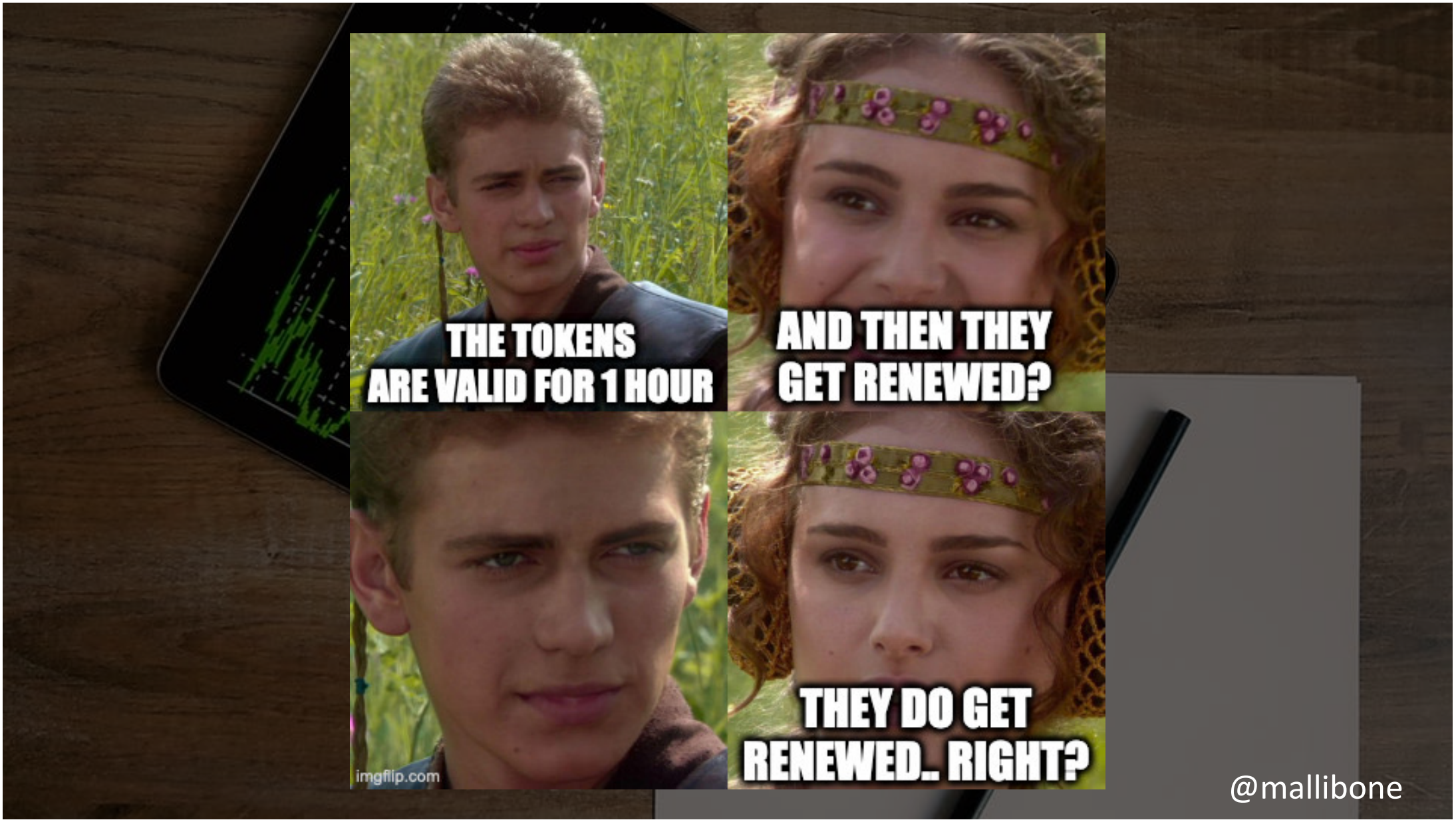

OidcSample.iOS › Debug › iPhone 12 iOS 15.0
Visual Studio Enterprise 2019 for Mac
13
Search solution
Solution
XamarinIdentity101 (main)
Mobile
OidcSample
Connected Services
Dependencies
Services
ViewModels
MainViewModel.cs
Views
MainPage.xaml
Annotations.cs
App.xaml
App.xaml.cs
AssemblyInfo.cs
OidcSample.Android
OidcSample.iOS
OidcSample.UITests
Server
App.xaml.cs
No selection
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;
namespace OidcSample
{
public partial class App : Application
{
// oidcxamarin101:/authenticated
public const string CallbackUri = "oidcxamarin101";
public static readonly string CallbackScheme = $"{CallbackUri}:/authenticated";
public static readonly string SignoutCallbackScheme = $"{CallbackUri}:/signout-callback-oidc";
public App()
{
InitializeComponent();
MainPage = new NavigationPage(new MainPage());
}
protected override void OnStart()
{
}
protected override void OnSleep()
{
}
protected override void OnResume()
Source
Changes
Blame
History
Merge
Errors
Errors: 0
Warnings: 13
Messages: 0
Filter
Line
Description
File
Project
Path
Category
Tasks
Package Console
Tool Output
Application Output - OidcSample.iOS
Vim Output
Build Output
Terminal
XAML Hot Reload
Toolbox
Properties
Document Outline
Tests

@mallibone



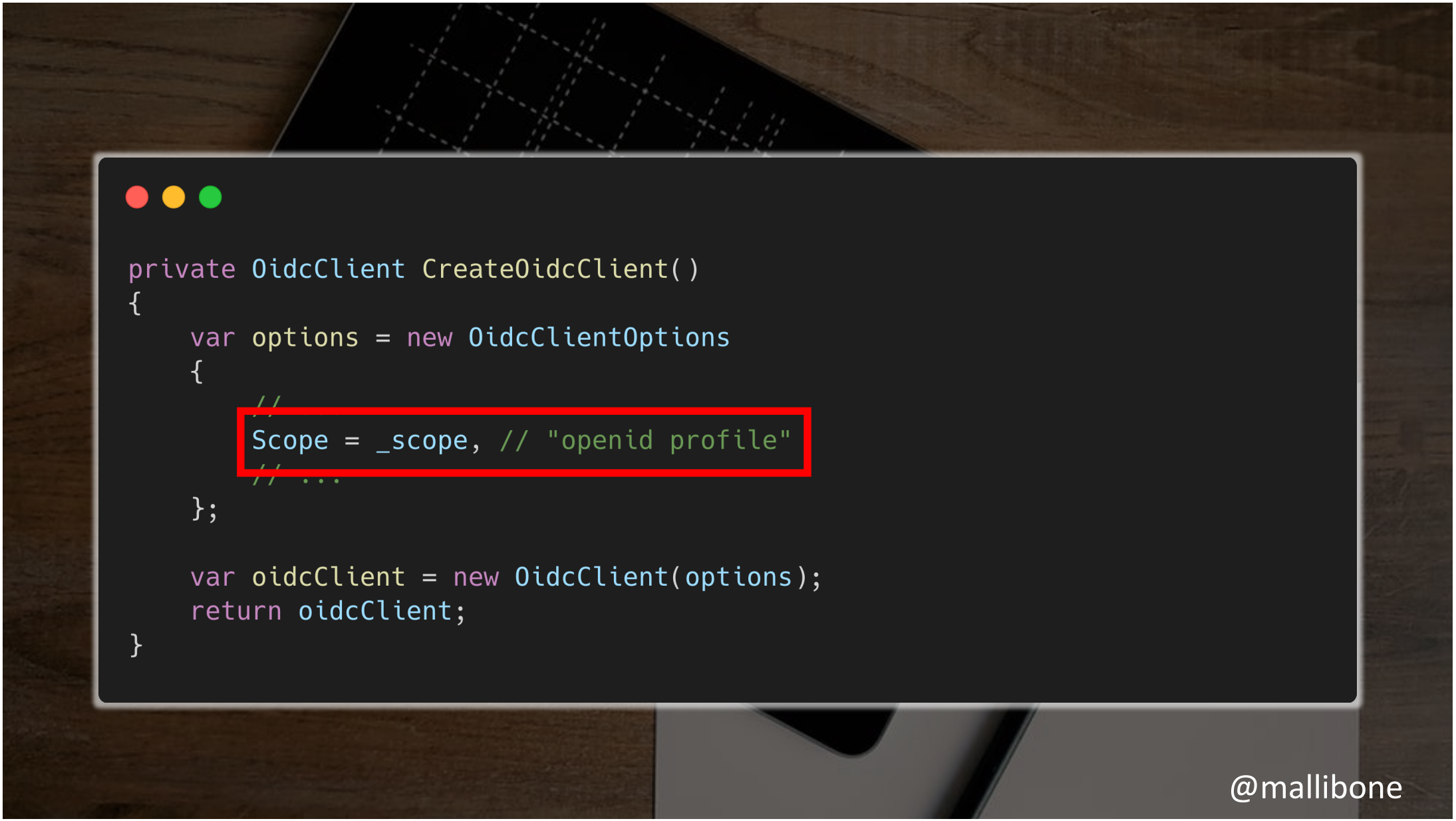
Recap – ID Token and Auth Token

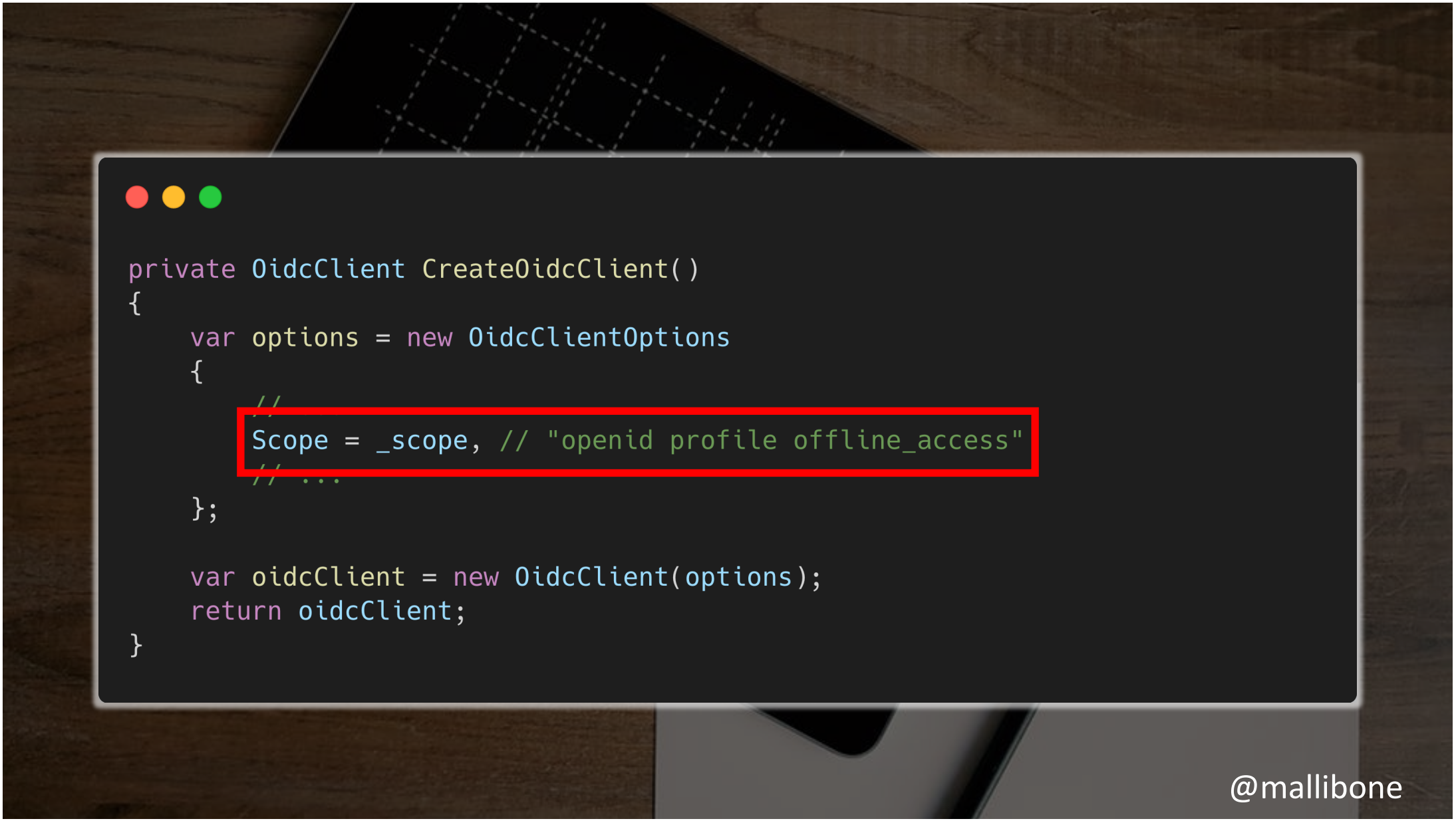
- The ID Token contains information about the Auth Token
- The ID Token can contain information of the user
- Use the ID Token and not the Auth Token for Information


THE TOKENS ARE VALID FOR 1 HOUR
AND THEN THEY GET RENEWED?
THEY DO GET RENEWED.. RIGHT?
imgflip.com
@mallibone

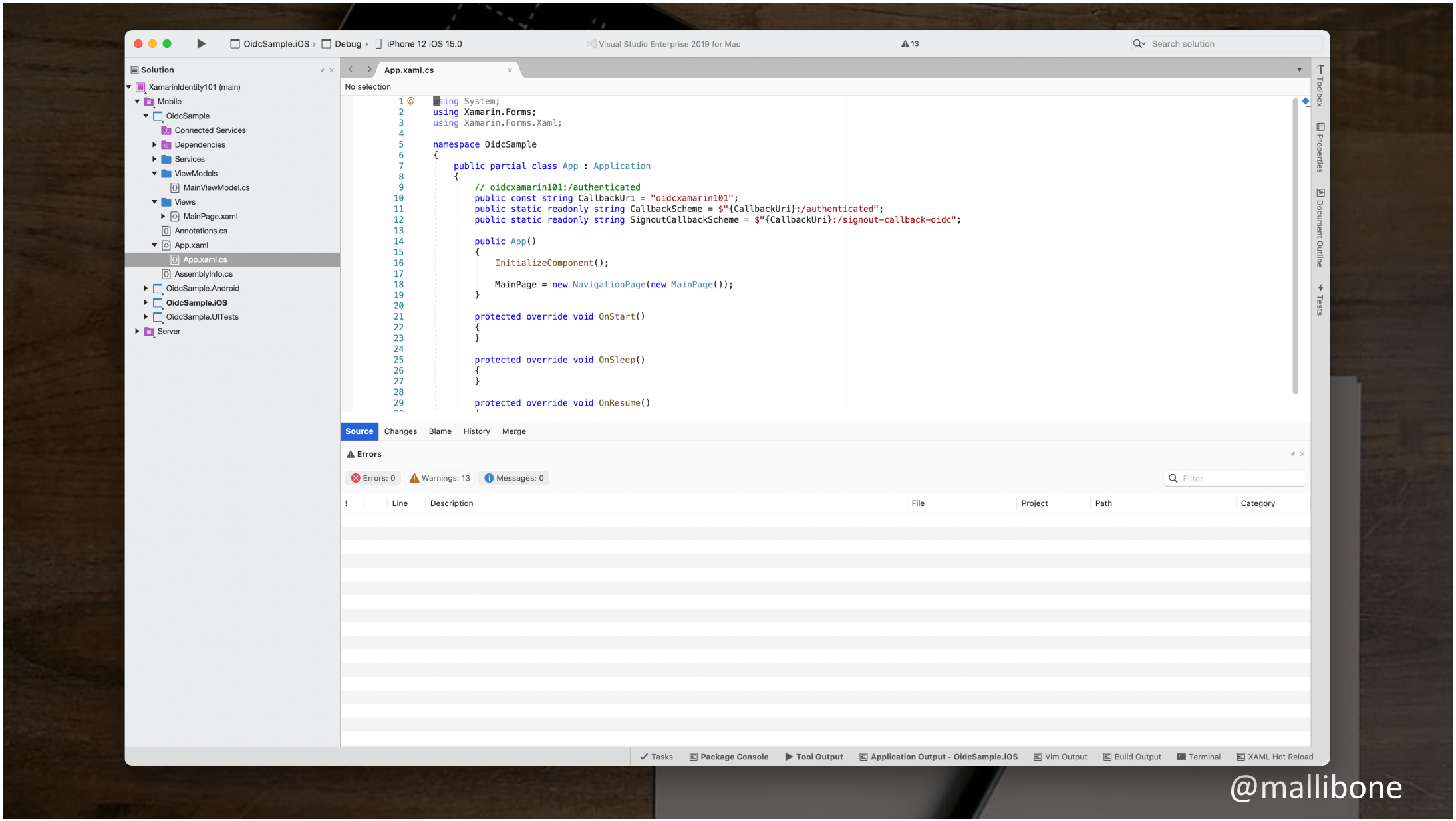
Refreshing your tokens

@mallibone

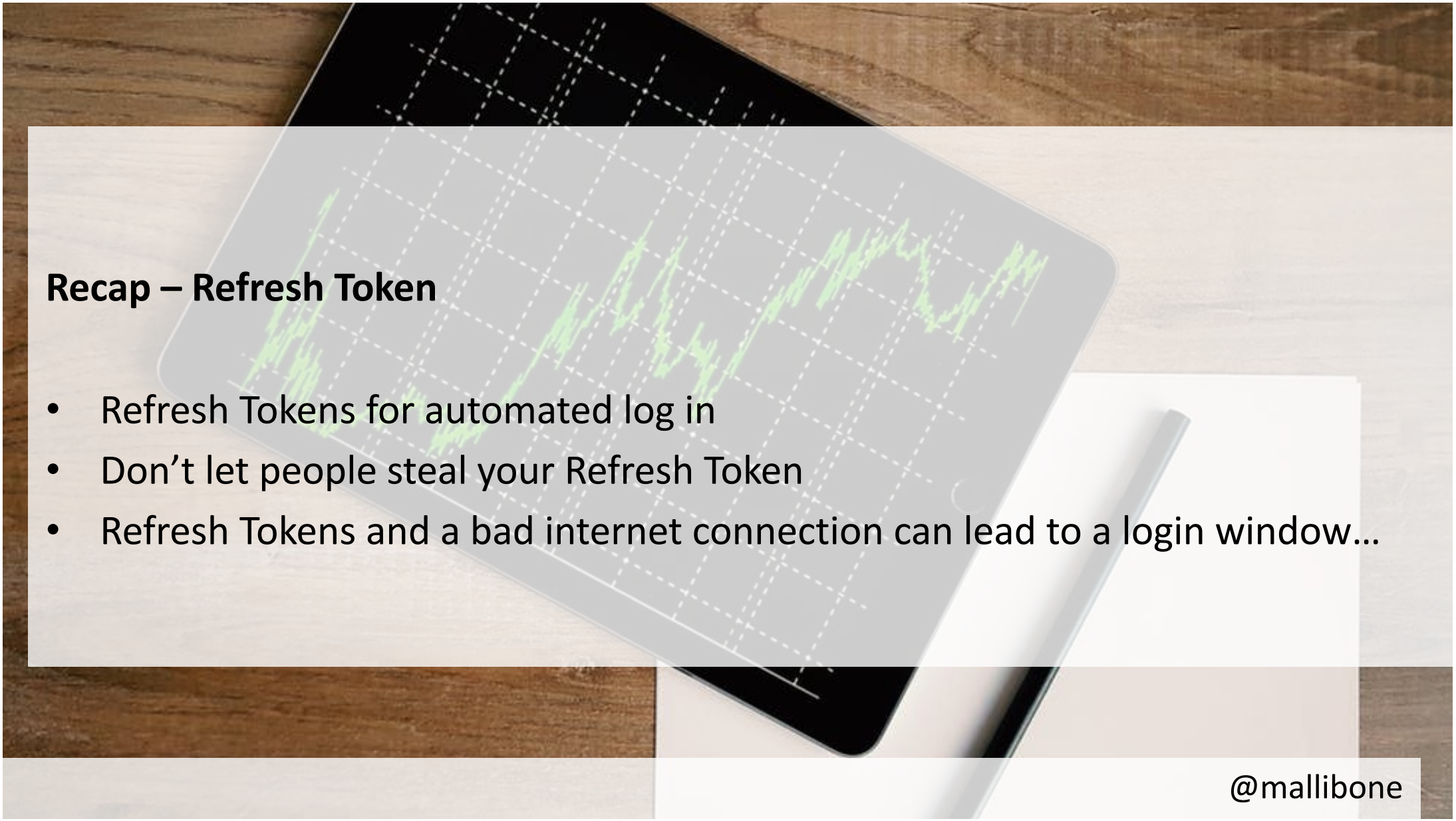

private OidcClient CreateOidcClient()
{
var options = new OidcClientOptions
{
//
Scope = _scope, // "openid profile"
// ...
};
var oidcClient = new OidcClient(options);
return oidcClient;
}
@mallibone


private OidcClient CreateOidcClient()
{
var options = new OidcClientOptions
{
//
Scope = _scope, // "openid profile offline_access"
// ...
};
var oidcClient = new OidcClient(options);
return oidcClient;
}
@mallibone


IT IS IMPORTANT
THAT THE SERVER IS
CONFIGURED TO ALLOW REFRESH TOKENS
imgflip.com
@mallibone


OidcSample.iOS › Debug › iPhone 12 iOS 15.0
Visual Studio Enterprise 2019 for Mac
13
Search solution
Solution
XamarinIdentity101 (main)
Mobile
OidcSample
Connected Services
Dependencies
Services
ViewModels
MainViewModel.cs
Views
MainPage.xaml
Annotations.cs
App.xaml
App.xaml.cs
AssemblyInfo.cs
OidcSample.Android
OidcSample.iOS
OidcSample.UITests
Server
App.xaml.cs
No selection
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;
namespace OidcSample
{
public partial class App : Application
{
// oidcxamarin101:/authenticated
public const string CallbackUri = "oidcxamarin101";
public static readonly string CallbackScheme = $"{CallbackUri}:/authenticated";
public static readonly string SignoutCallbackScheme = $"{CallbackUri}:/signout-callback-oidc";
public App()
{
InitializeComponent();
MainPage = new NavigationPage(new MainPage());
}
protected override void OnStart()
{
}
protected override void OnSleep()
{
}
protected override void OnResume()
Source
Changes
Blame
History
Merge
Errors
Errors: 0
Warnings: 13
Messages: 0
Filter
Line
Description
File
Project
Path
Category
Tasks
Package Console
Tool Output
Application Output - OidcSample.iOS
Vim Output
Build Output
Terminal
XAML Hot Reload
Toolbox
Properties
Document Outline
Tests

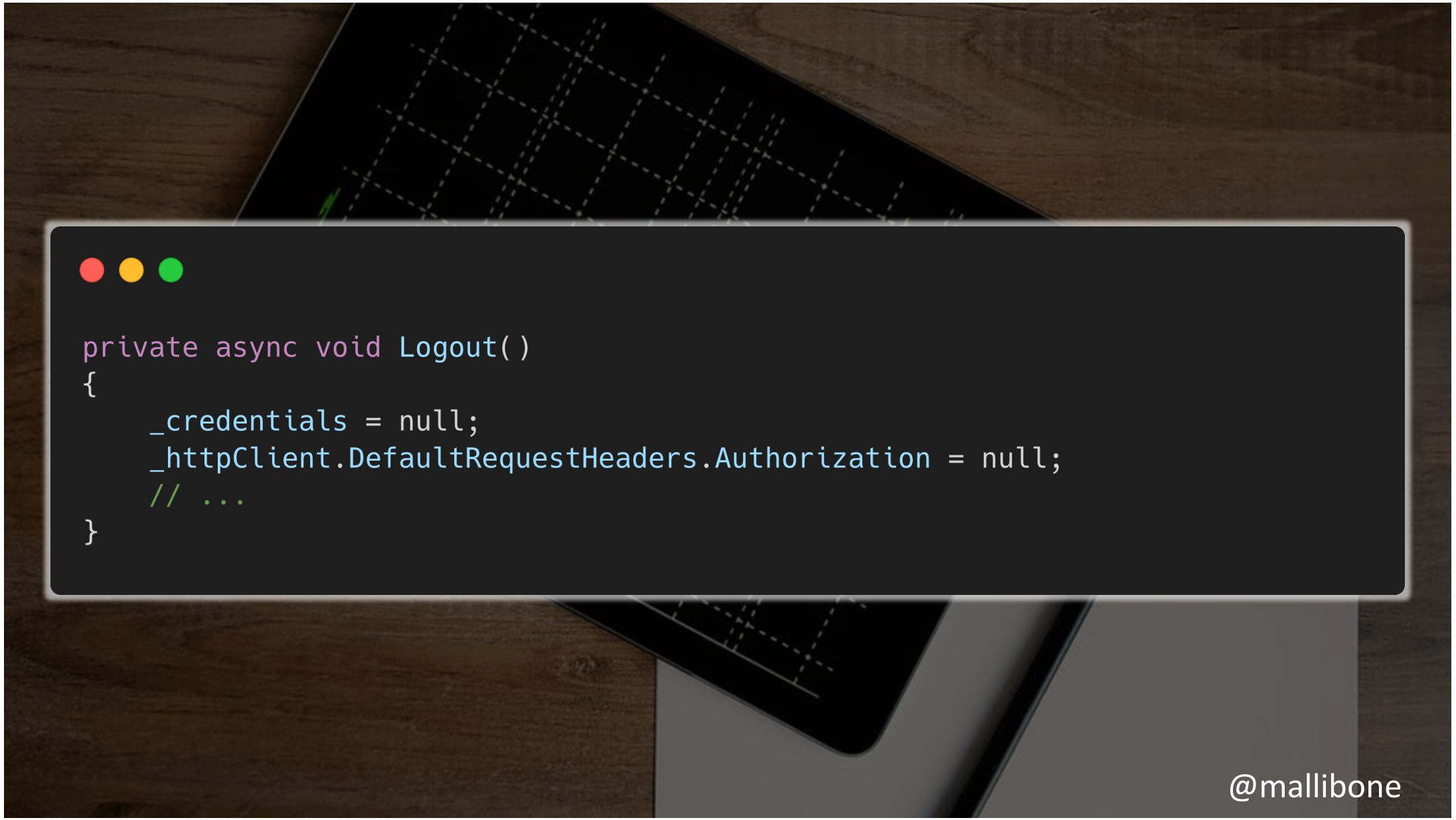
@mallibone

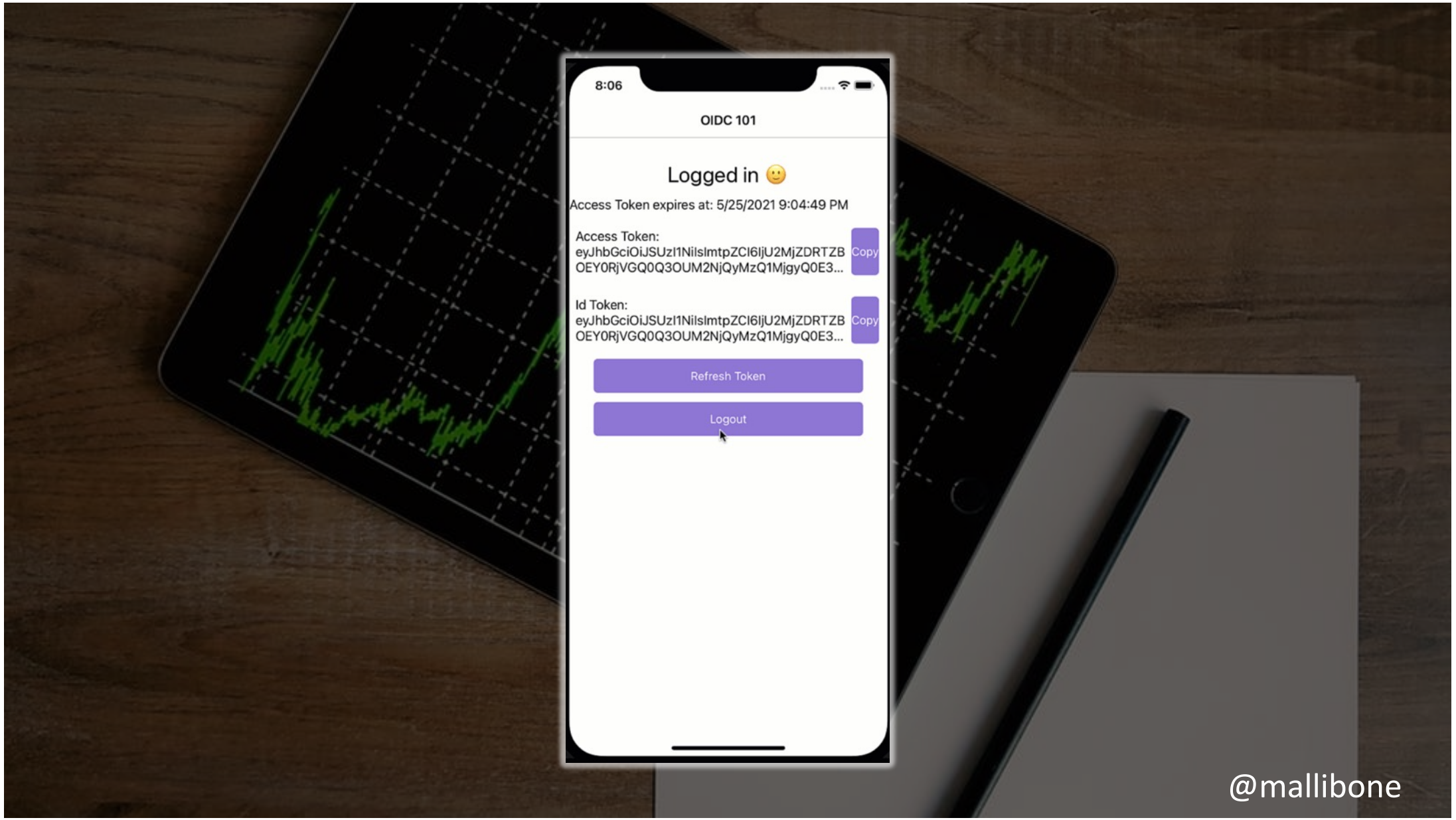


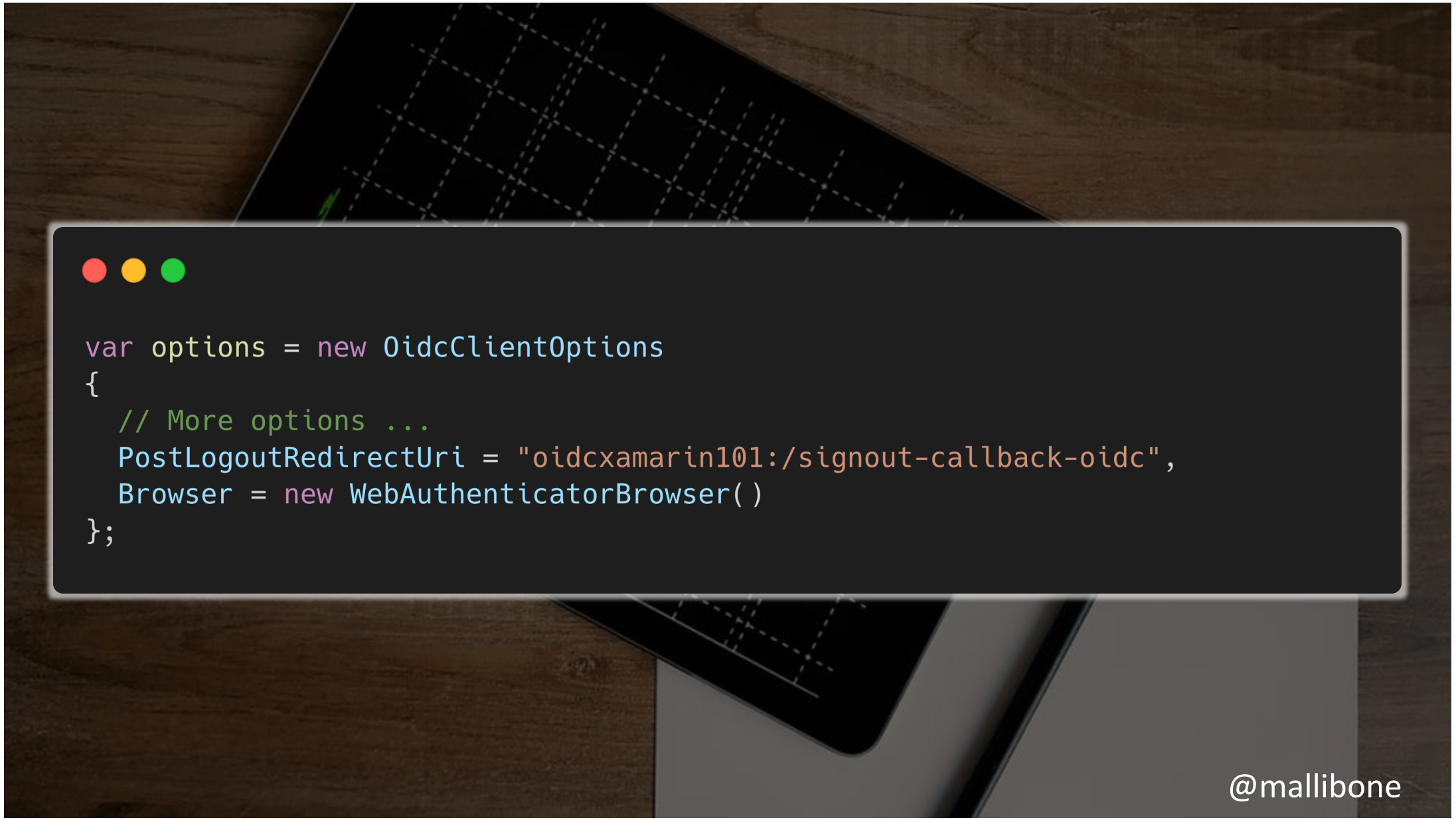
Recap – Refresh Token

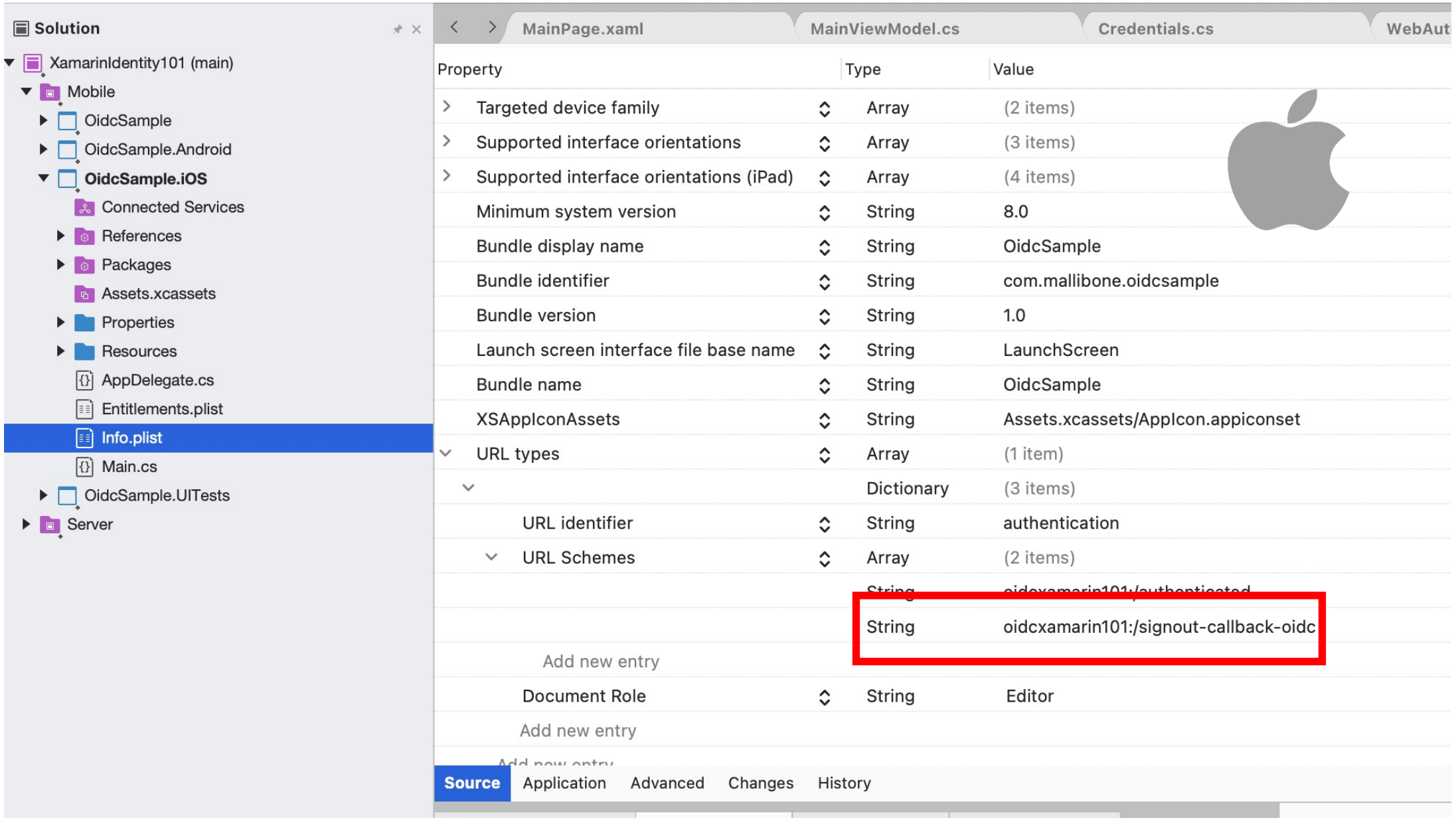
- Refresh Tokens for automated log in
- Don’t let people steal your Refresh Token
- Refresh Tokens and a bad internet connection can lead to a login window…

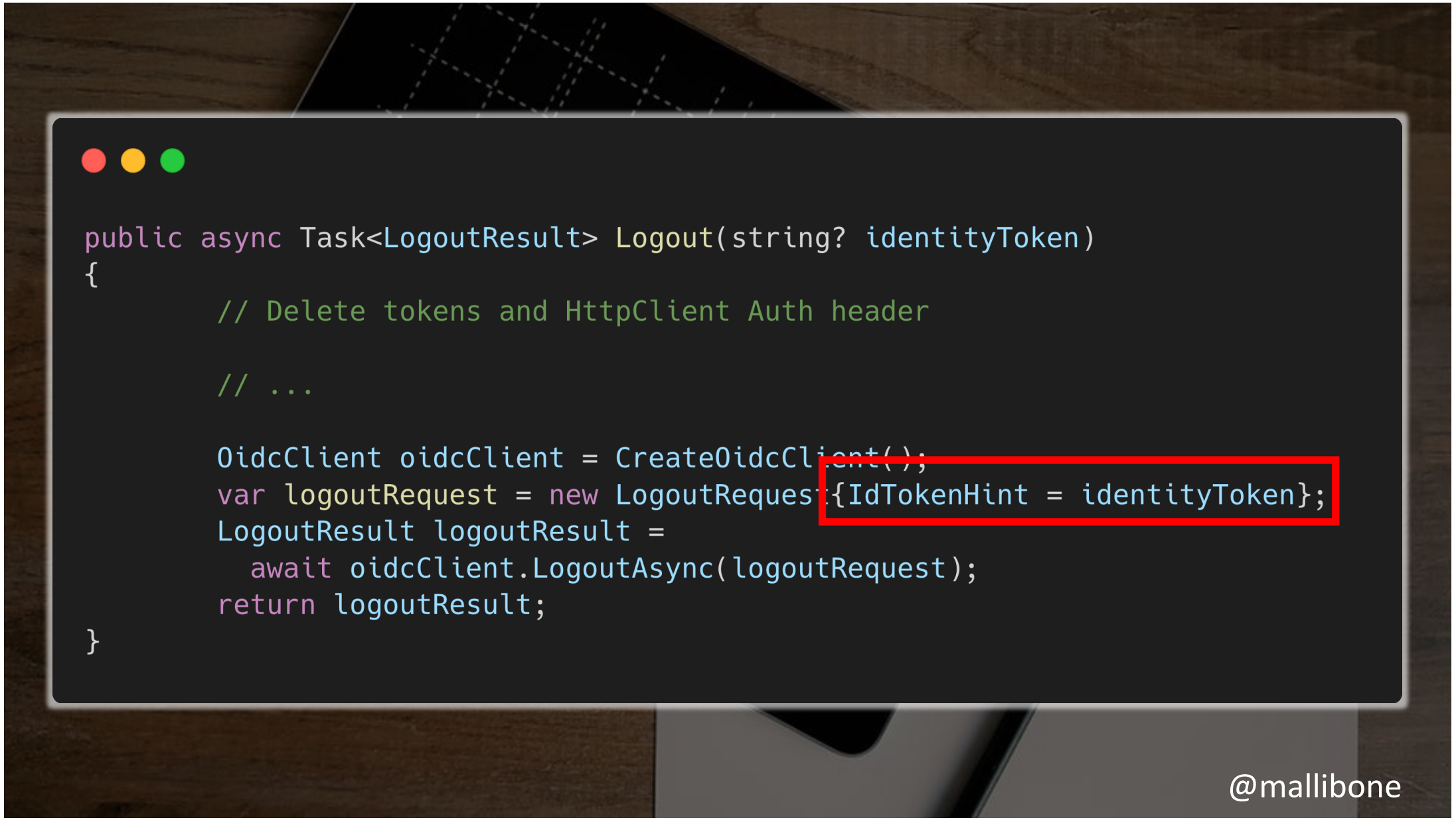

Logging out

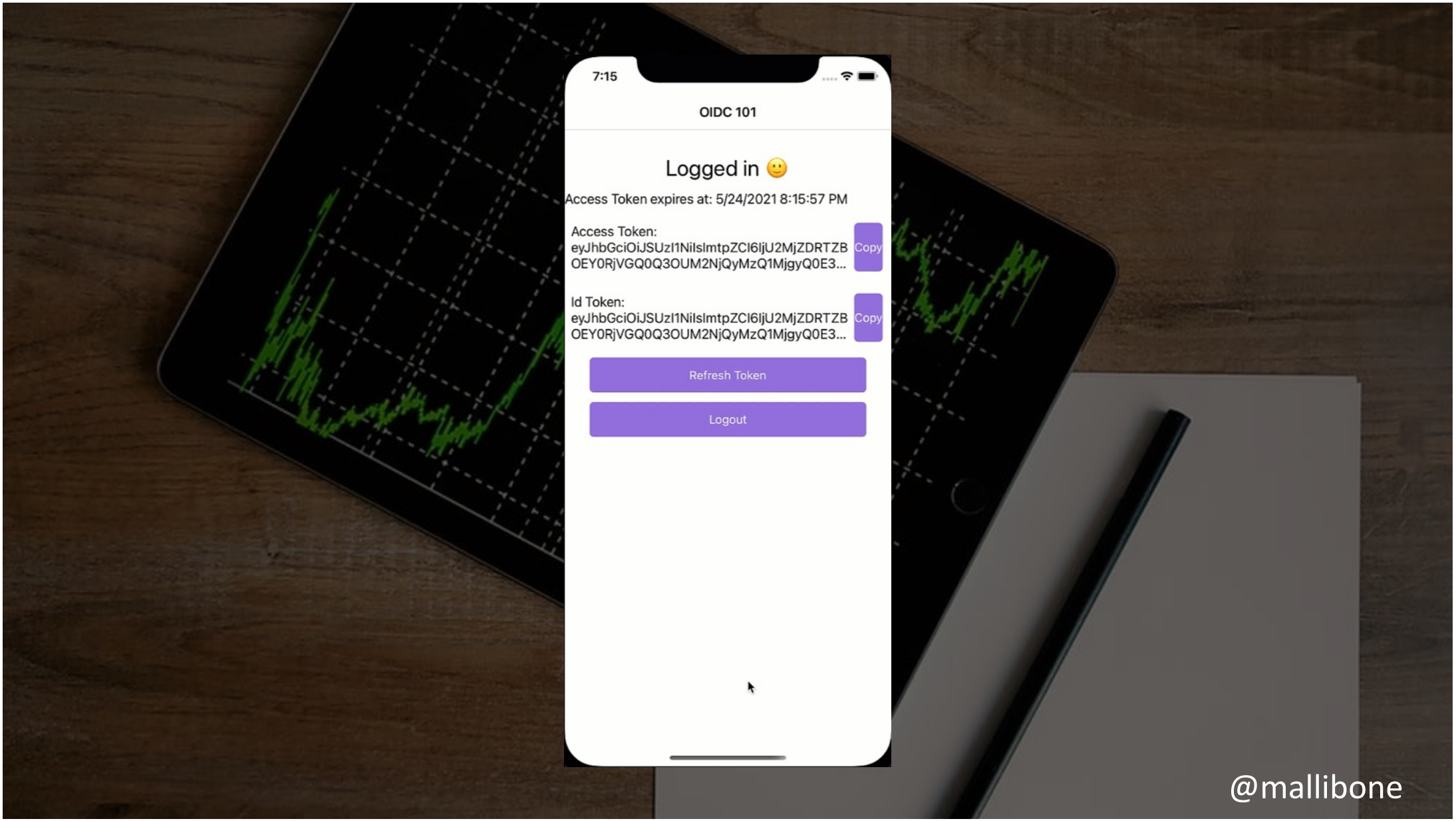

private async void Logout()
{
_credentials = null;
_httpClient.DefaultRequestHeaders.Authorization = null;
// ...
}
@mallibone

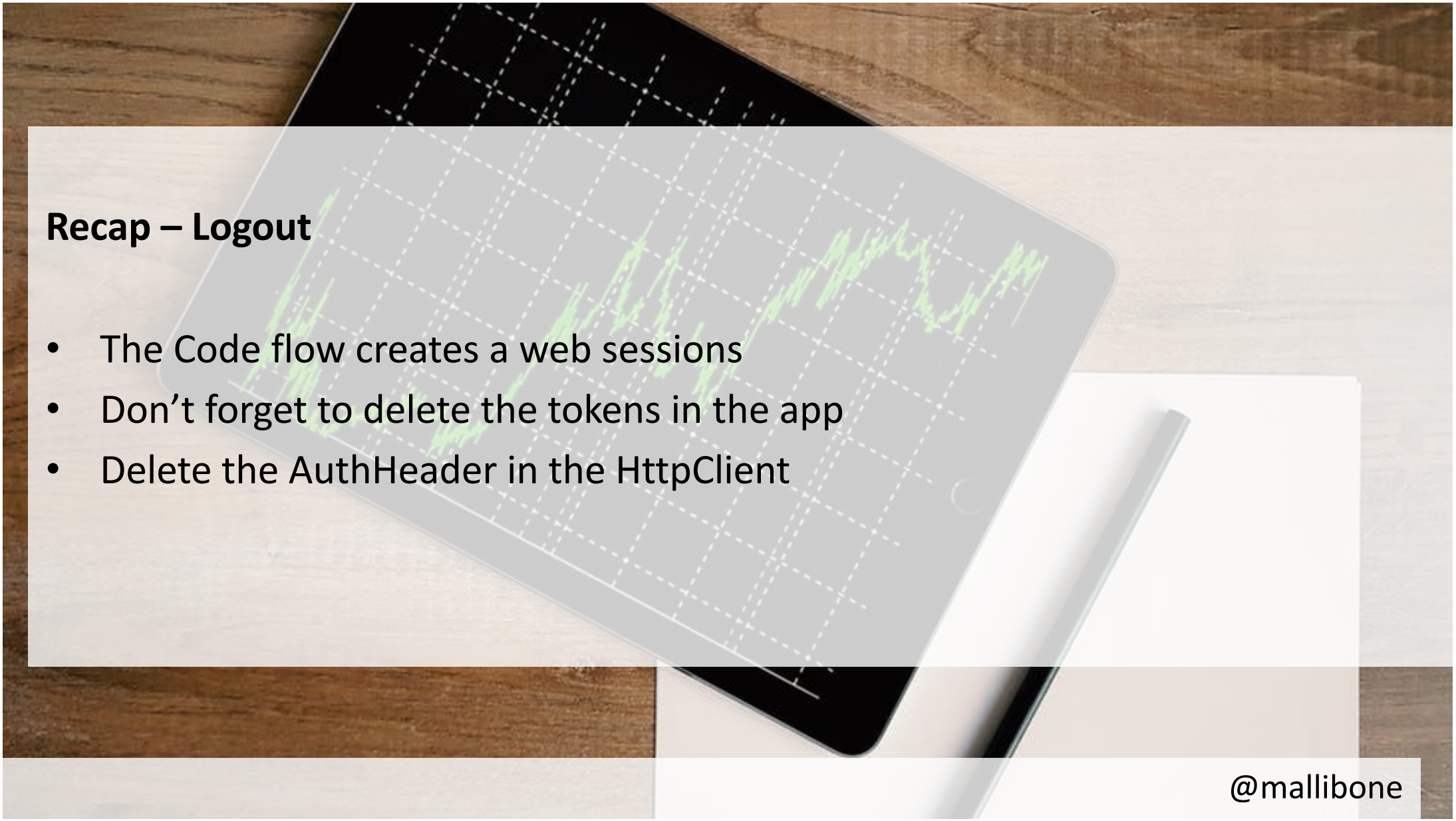

8:06
OIDC 101
Logged in 🙂
Access Token expires at: 5/25/2021 9:04:49 PM
Access Token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2MjZDRTZB
OEY0RjVGQ0Q3OUM2NjQyMzQ1MjgyQ0E3...
Copy
Id Token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2MjZDRTZB
OEY0RjVGQ0Q3OUM2NjQyMzQ1MjgyQ0E3...
Copy
Refresh Token
Logout
@mallibone


var options = new OidcClientOptions
{
// More options ...
PostLogoutRedirectUri = "oidcxamarin101:/signout-callback-oidc",
Browser = new WebAuthenticatorBrowser()
};
@mallibone


Solution
XamarinIdentity101 (main)
Mobile
OidcSample
OidcSample.Android
OidcSample.iOS
Connected Services
References
Packages
Assets.xcassets
Properties
Resources
AppDelegate.cs
Entitlements.plist
Info.plist
Main.cs
OidcSample.UITests
Server
MainPage.xaml
MainViewModel.cs
Credentials.cs
WebAut
Property
Type
Value
Targeted device family
Array
(2 items)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
Minimum system version
String
8.0
Bundle display name
String
OidcSample
Bundle identifier
String
com.mallibone.oidcsample
Bundle version
String
1.0
Launch screen interface file base name
String
LaunchScreen
Bundle name
String
OidcSample
XSAppIconAssets
String
Assets.xcassets/AppIcon.appiconset
URL types
Array
(1 item)
Dictionary
(3 items)
URL identifier
String
authentication
URL Schemes
Array
(2 items)
String
oidcxamarin101:/signout-callback-oidc
Add new entry
Document Role
String
Editor
Add new entry
Source
Application
Advanced
Changes
History

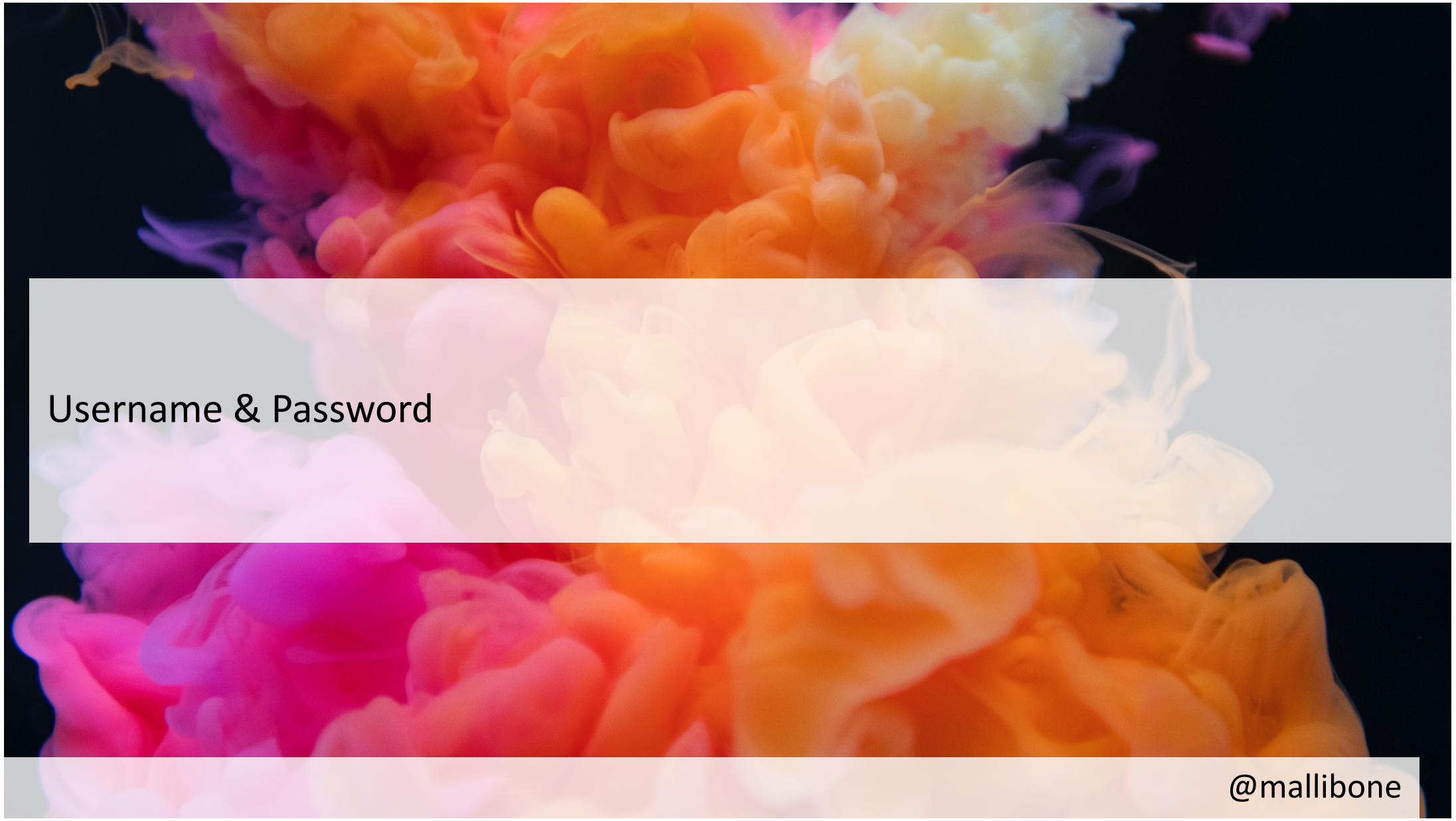

public async Task<LogoutResult> Logout(string? identityToken)
{
// Delete tokens and HttpClient Auth header
// ...
OidcClient oidcClient = CreateOidcClient();
var logoutRequest = new LogoutRequest{IdTokenHint = identityToken};
LogoutResult logoutResult =
await oidcClient.LogoutAsync(logoutRequest);
return logoutResult;
}
@mallibone

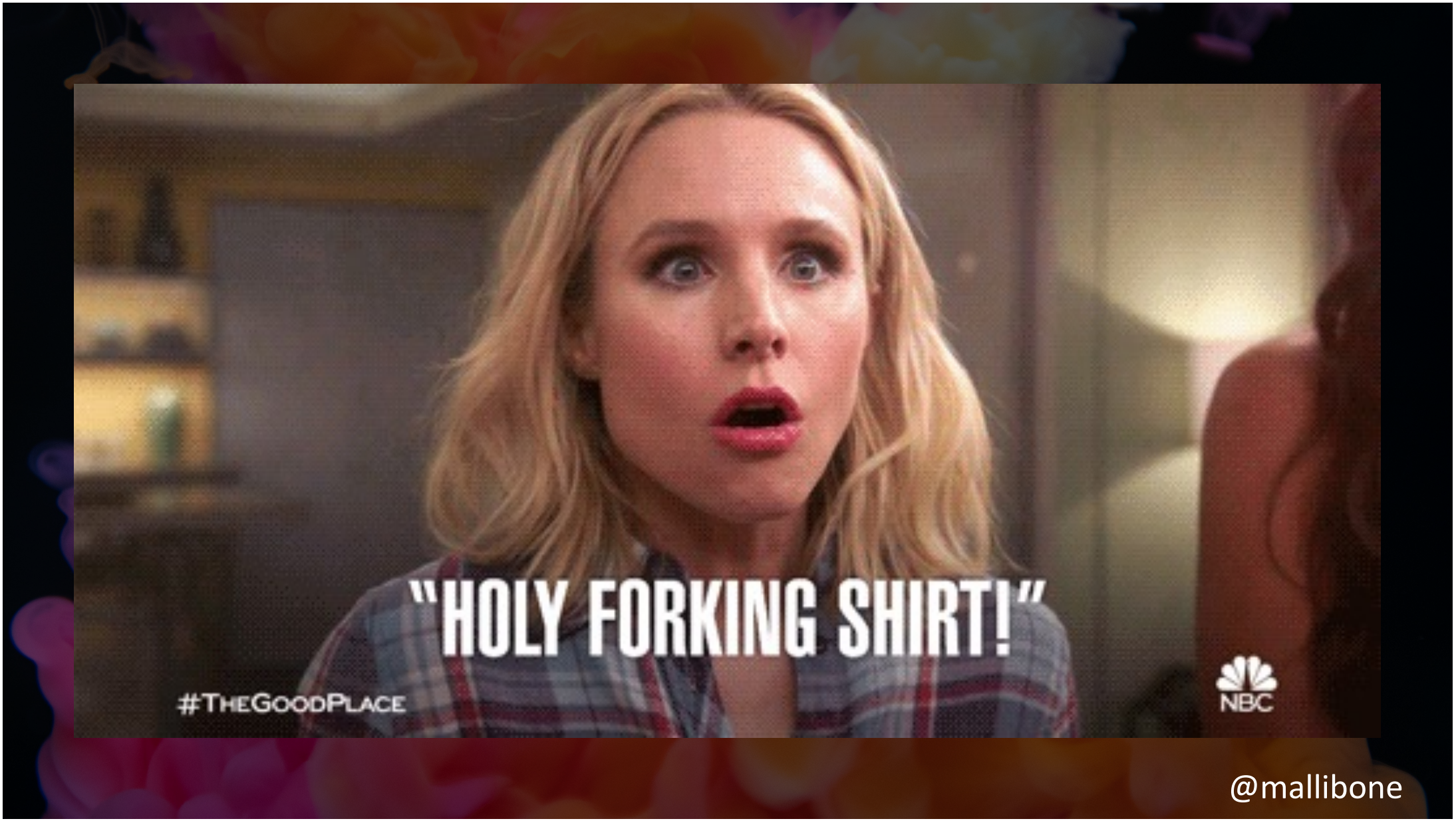

7:15
OIDC 101
Logged in 🙂
Access Token expires at: 5/24/2021 8:15:57 PM
Access Token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2MjZDRTZB
OEY0RjVGQ0Q3OUM2NjQyMzQ1MjgyQ0E3...
Copy
Id Token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2MjZDRTZB
OEY0RjVGQ0Q3OUM2NjQyMzQ1MjgyQ0E3...
Copy
Refresh Token
Logout
@mallibone



Recap – Logout

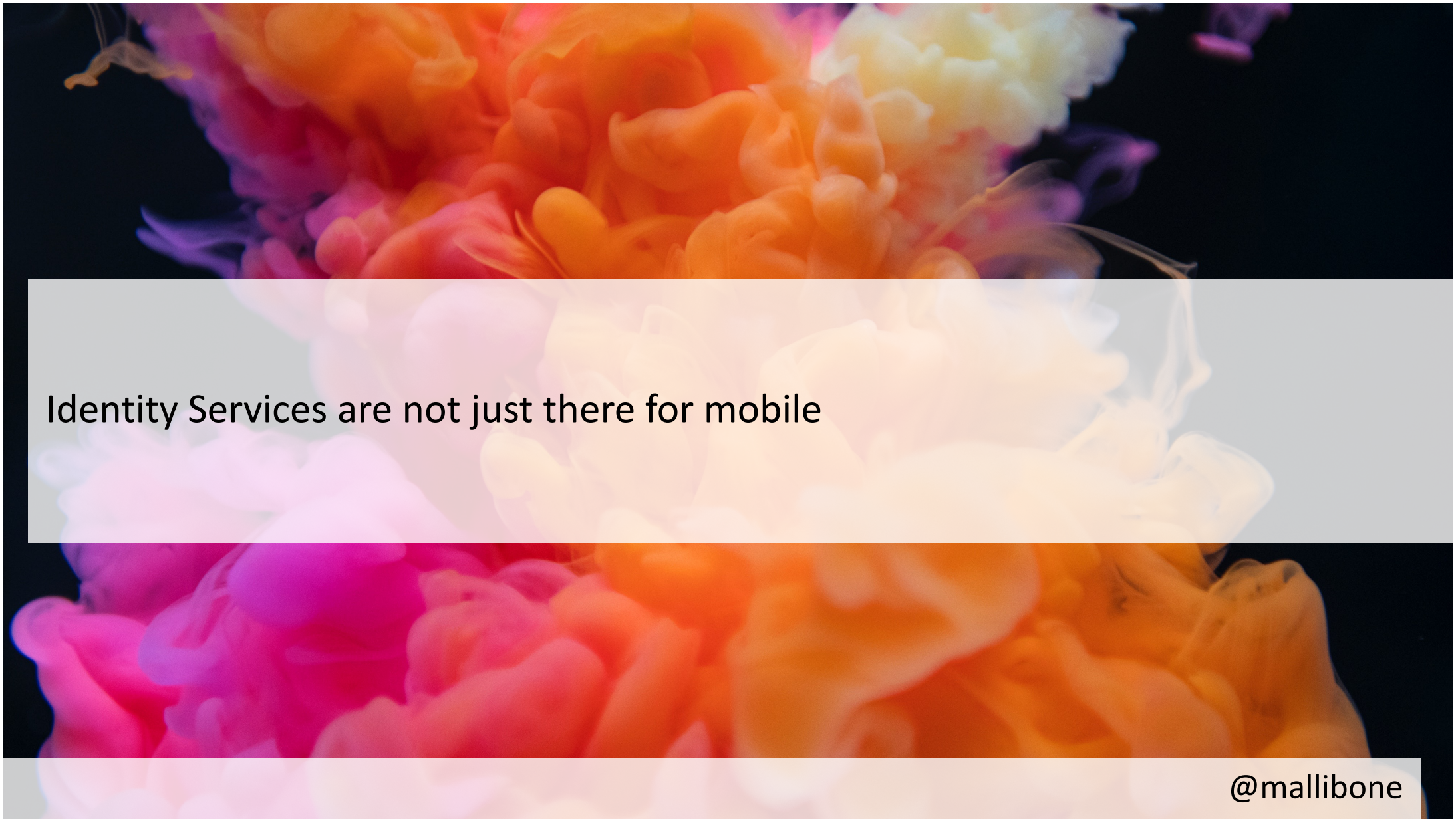
- The Code flow creates a web sessions
- Don’t forget to delete the tokens in the app
- Delete the AuthHeader in the HttpClient

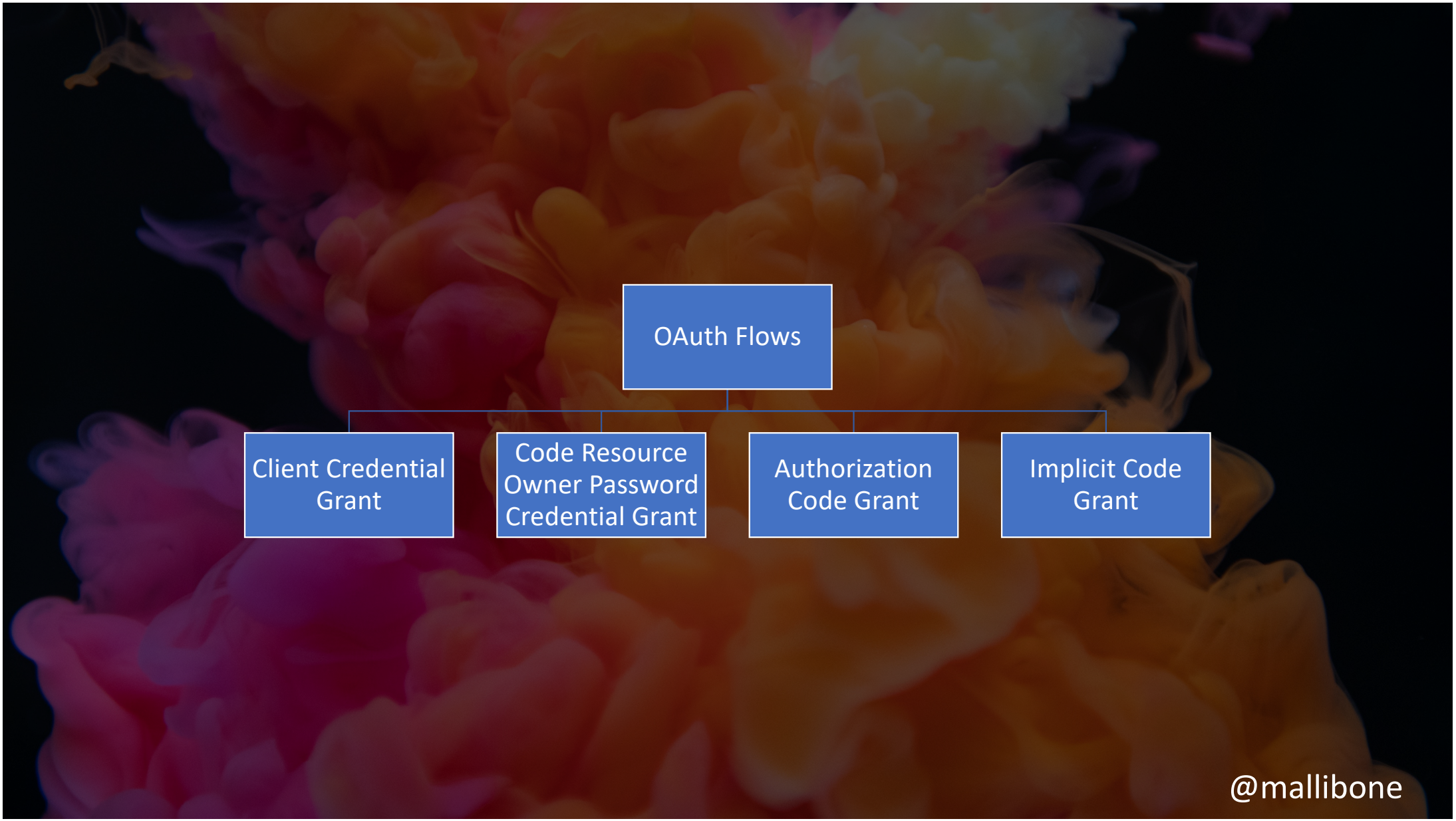

Username & Password
@mallibone

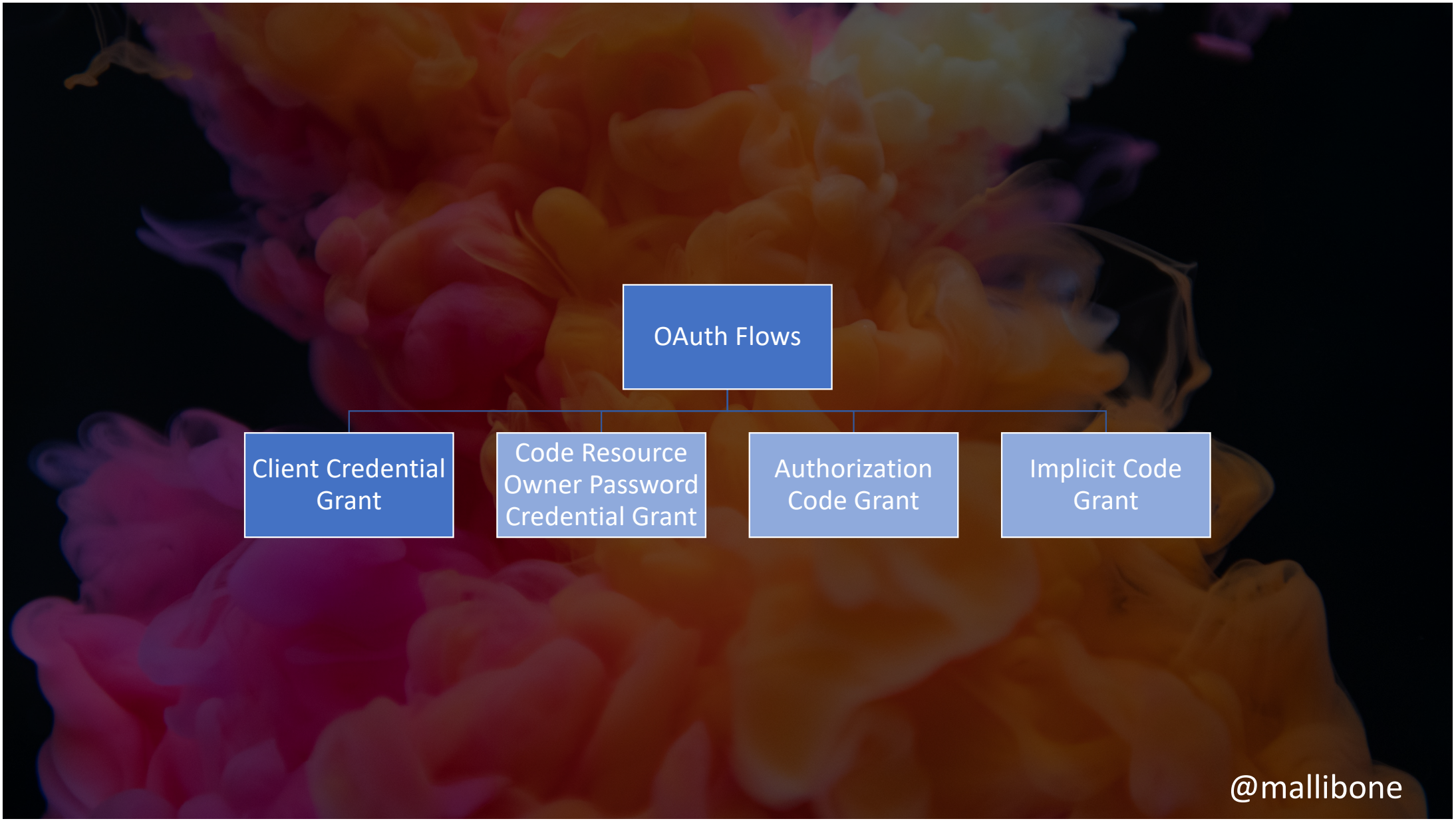


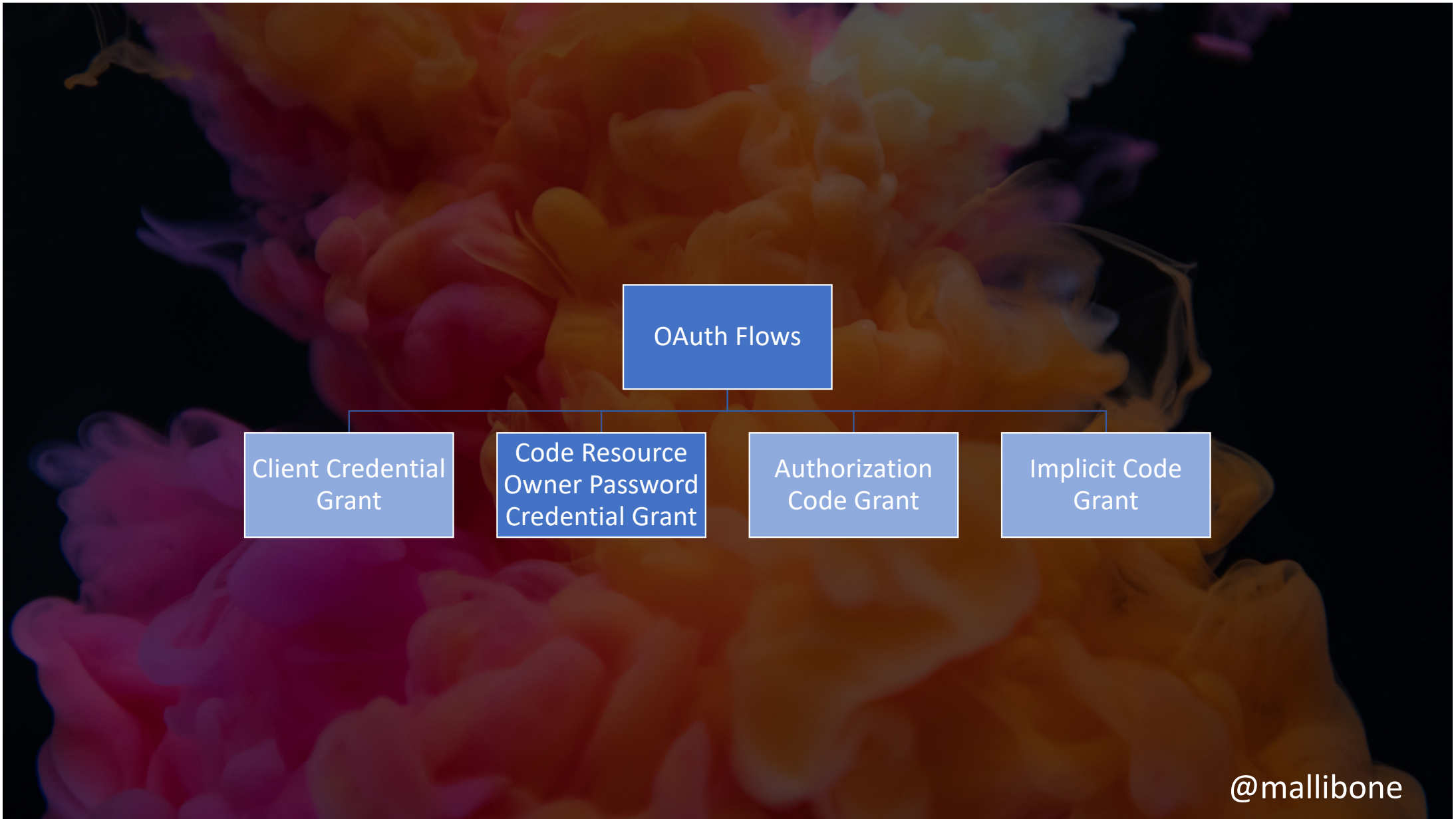
@mallibone

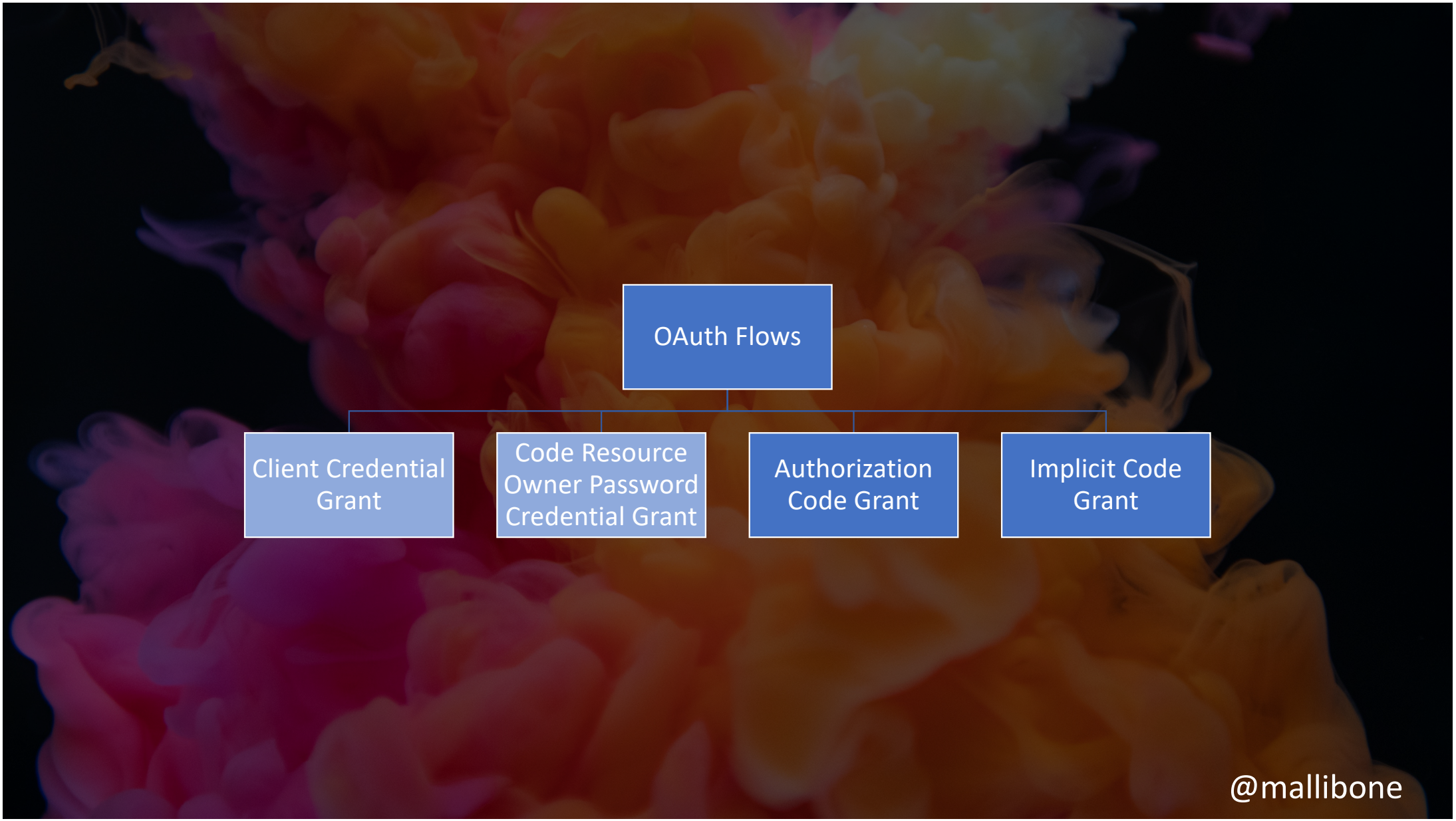
Is it worth it?


Identity Services are not just there for mobile
@mallibone


OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone

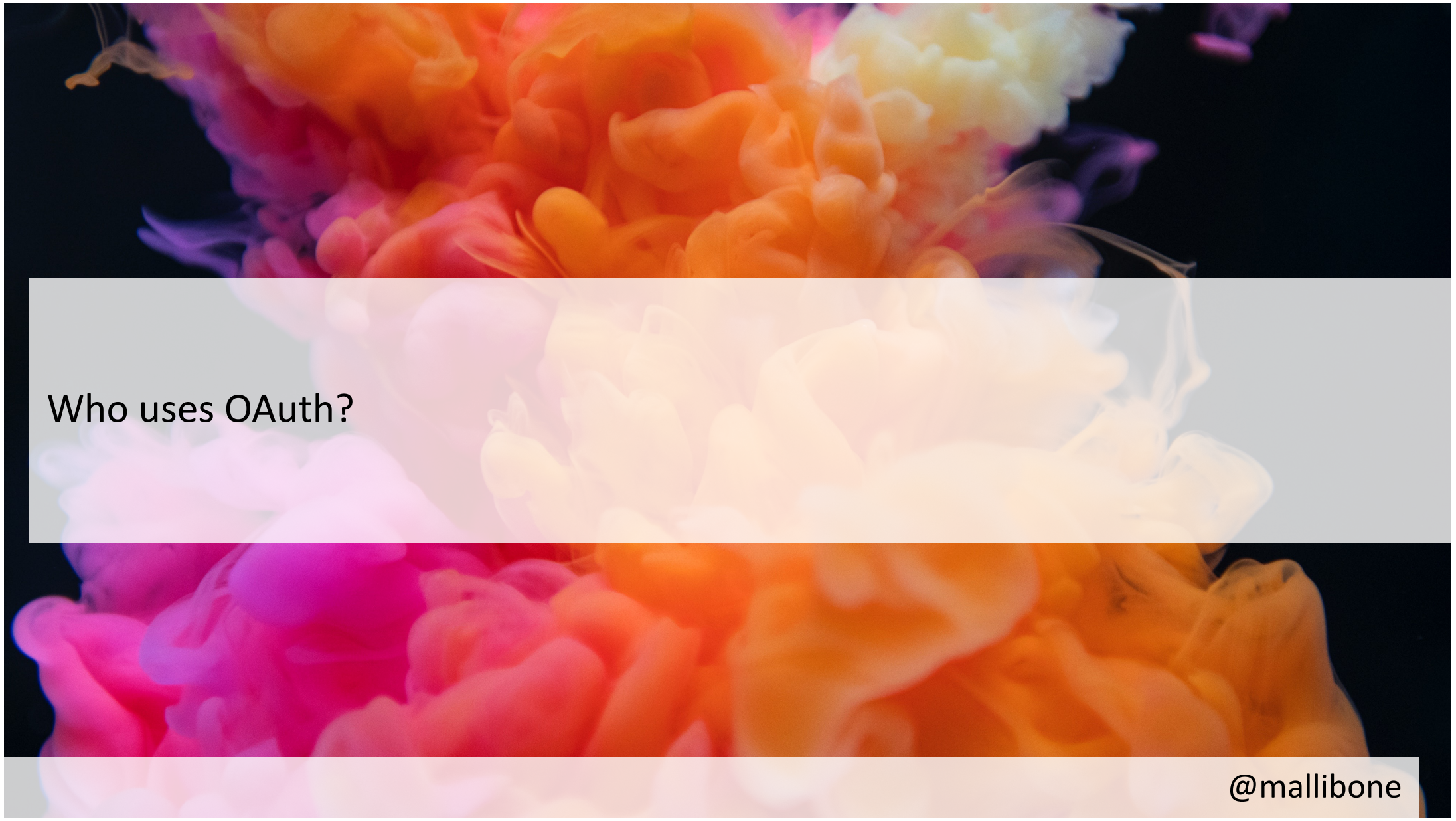

OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone

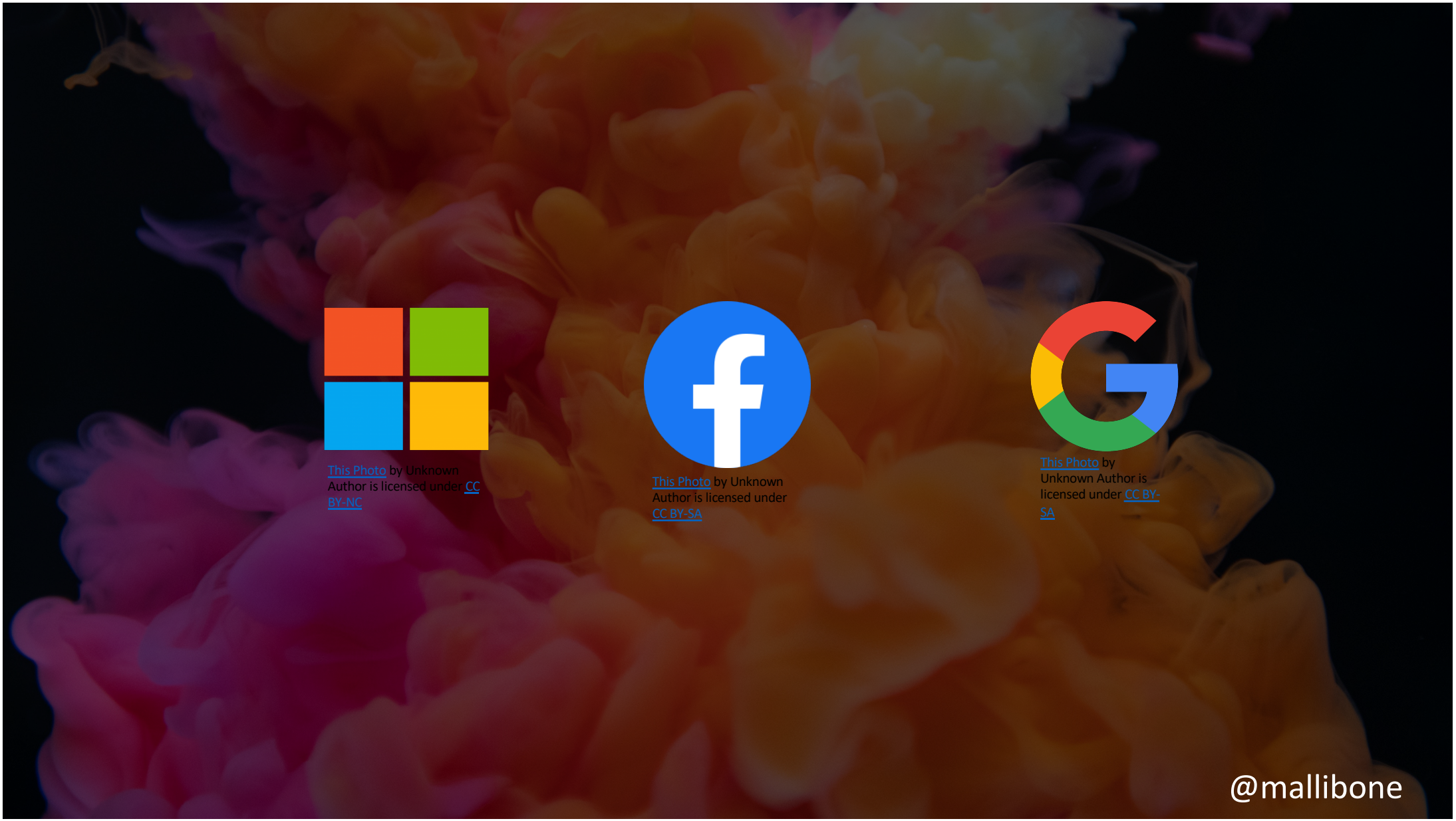

OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone


OAuth Flows
Client Credential Grant
Code Resource Owner Password Credential Grant
Authorization Code Grant
Implicit Code Grant
@mallibone


DEVELOPER
CODE RESOURCE OWNER
PASSWORD CREDENTIAL GRANT
SECURITY
BEST PRACTICES
imgflip.com

@mallibone


Who uses OAuth?
@mallibone

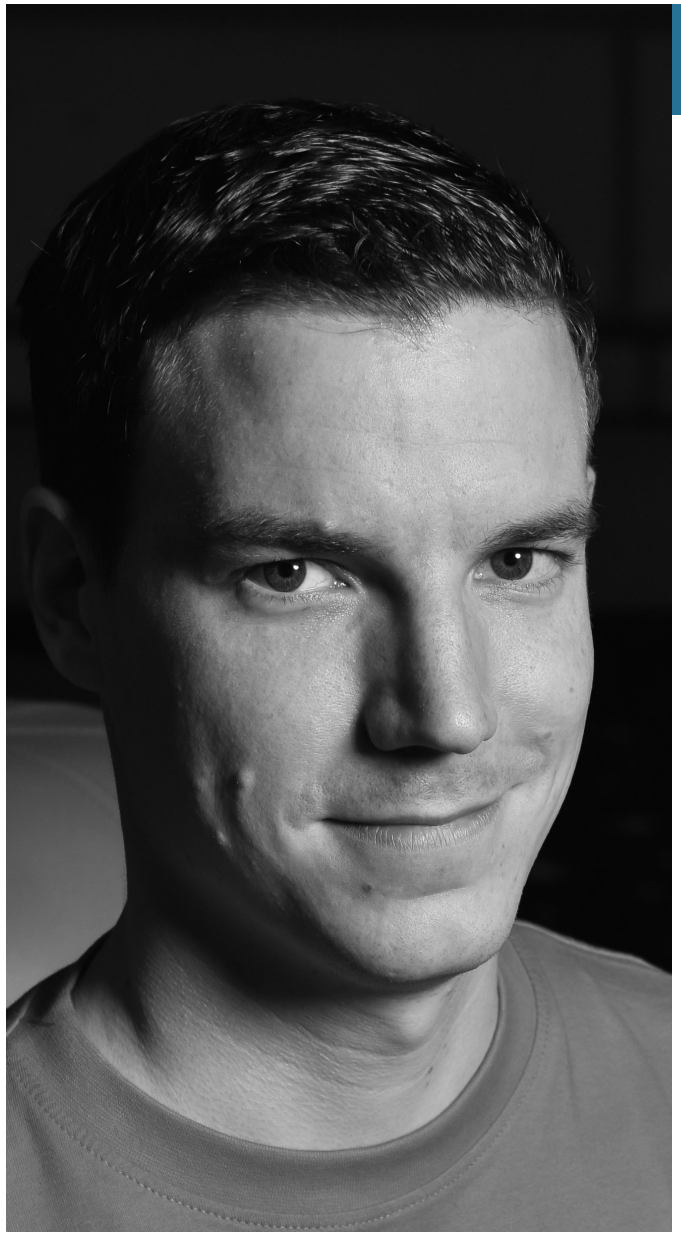

This Photo by Unknown Author is licensed under CC BY-NC
This Photo by Unknown Author is licensed under CC BY-SA
This Photo by Unknown Author is licensed under CC BY-SA
@mallibone

And what about MAUI?

Takeaways

Takeaways

- OAuth 2 and OpenId Connect are a wide spread industry standard
- If you don’t know the password, you can’t be blamed for loosing it.
- With Xamarin Essentials and OIDC Client NuGet packages the basic building blocks are there.



REY TECHNOLOGY

Thank you for your time!

Mark Allibone @mallibone

Rey Technology

https://mallibone.com

https://nullpointers.io

https://bit.ly/3ojDyVr

Microsoft® MVP Most Valuable Professional