Zello Work Server API Libraries

Project Structure

There are five client libraries included in this repository:

1. `PHP`
2. `Swift`
3. `Objective C`
4. `Java`
5. `C#`

In addition, we offer our Swift and Objective C libraries available as `CocoaPods.`

Each library provides a ZelloAPI class and an test for the ZelloAPI. For the Swift, Objective C, Java and C# libraries, this test comes in the form of a project titled `APITest`. These projects will output the results of the `APITest` to the console.

PHP Library

The `PHP` library includes a `zello_server_api.class.php` file and a `api_test.php` script to test the functionality of the `zello_server_api.class.php` class.

To use `api_test.php`, replace the $host variable, the $apikey variable and replace the username and password strings in the `auth` method. Then, simply run the script and view the output.

Swift Library

CocoaPod

The `Swift` CocoaPod creates a `ZelloAPISwift` module that can be imported into any Swift file that wishes to access the Zello Work API.

To install using Swift 3, add `pod 'ZelloAPISwift'` to your Podfile. To use Swift 2.2, add `pod 'ZelloAPISwift', '1.0.3'` to your Podfile. For more information, please see the `Example Project.`

Dependencies

- Swift 2.2 or higher.
- Minimum iOS Version: 8.0

Manual Installation

The `Swift` library includes a `ZelloAPI.swift` file and a test project `APITest` to test the functionality of the `ZelloAPI.swift` class.

`APITest` is an iOS app project that can be run using Xcode on macOS. Open `ViewController.swift` and replace the `APITest` constructor Strings with the hostname, API key, username, and password. Then, simply run the project and view the output.

Dependencies

- The Swift library includes a reference to `CommonCrypto`, a C library, for the MD5 hashing of login credentials. Unfortunately, due to Swift limitations, C libraries cannot be simply imported. Instead, Swift provides a method of importing C code through `Bridging Headers`. **Any project using the `ZelloAPI.swift` class will need to having a bridging header with the following import:** `#import <CommonCrypto/CommonCrypto.h>`
- Swift 3. For those wishing to target Swift 2.2, the source code can be found `here.`
- Minimum iOS Version: 7.0

Objective C Library

CocoaPod

The `Objective C` CocoaPod creates a `ZelloAPIObjC` module that can be imported into any Objective C file that wishes to access the Zello Work API.

To install, add `pod 'ZelloAPIObjC'` to your Podfile. For more information, please see the `Example Project.`

Dependencies

- Minimum iOS Version: 8.0

Manual Installation

The `Objective C` library includes `ZelloAPI.h` and `ZelloAPI.m` files and a test project `APITest` to test the functionality of the `ZelloAPI` class.

`APITest` is an iOS app project that can be run using Xcode on macOS. Open `ViewController.m` and replace the `APITest` constructor NSStrings with the hostname, API key, username, and password. Then, simply run the project and view the output.

Dependencies

- Minimum iOS Version: 7.0

Java Library

The `Java` library includes a `ZelloAPI.java` file and a test project `APITest` to test the functionality of the `ZelloAPI.java` class.

`APITest` is an Android app project that can be run using Android Studio. Open `MainActivity.java` and replace the `APITest` constructor Strings with the hostname, API key, username and password. Then, simply run the project and view the output.

C# Library

The `C#` library includes a `ZelloAPI.cs` file and a test project `APITest` to test the functionality of the `ZelloAPI.cs` class.

`APITest` is a Visual Studio console project that can be run using Visual Studio on Windows or Xamarin Studio on macOS. Open `Program.cs` and replace the `APITest` constructor strings with the hostname and API key for your network. Then, replace the `Authenticate` method strings with the administrative username and password. Lastly, run the project and view the output.

Dependencies

A reference to the `System.Web.Extensions` component is required for any project adding the `ZelloAPI.cs` class.

See also

- Zello Work API reference
- Zello Work client SDK for Android