

# iOS Developer Challenge

## Problem:

Implement an iOS app that presents a part of [SoundCloud](#) user profile. This might be any user including yourself or the app may give an option to specify user id on launch.

Running the app should bring the user to the screen that includes two sections:

1. Profile information (first name, last name, avatar).
2. List of user's favorite tracks.

Data for both sections can be obtained publicly from the [SoundCloud API](#). Please use the following client\_id for of API calls authorization:  
c23089b7e88643b5b839c4b8609fce3b.

## Requirements:

1. The app must perform HTTP requests for filling up profile sections in parallel.
2. The app must update both sections simultaneously. It is not acceptable to update the UI in 2 separate moments.
3. The data shown must be refreshed automatically every 2 minutes.
4. There should be a way to trigger a manual update of the data.
5. The user should be informed if an error occurred while fetching the data.

## Third party libraries:

Try to minimize using of third party libraries. If you really want to, keep in mind that their usage should not affect the core logic of the app and these libraries should be replaceable. However, you may use Reactive Cocoa or any other FRP library without limitation. Moreover, their usage is highly encouraged (though not necessary!).

## Language:

Please use Objective-C. Swift is a great language, but we avoid using it for various reasons (at least until Swift 4), so we want to test your Objective-C skills.

### **Coding Guide:**

1. Please use [style-guide.pdf](#) as a coding style reference. Be consistent with the specified guidelines as much as possible – that’s very important!
2. Pay attention to the quality of the code, and the overall design of your app.

### **UI design:**

App UI is completely up to you. We absolutely love animations and UI effects, but please pay attention to the core part of your app. It is perfectly fine if your app has a simplistic UI, while it’s engine is well thought and tested.

### **Tests:**

Though definitely a plus, we do not require you to write unit tests. But please note that we expect your code be testable (hint: dependency injection can be really helpful!).

### **Submission:**

Create a zip file including the source code, build scripts (if any) and any other relevant content, and send it to us. Please do not share your solution.