Podio Mobile Developer Assignment

The Assignment

The task is to implement a small app that authenticates a user against the Podio API using email and password and displays a list of the user's workspaces, grouped by organization.

The goal of this exercise is to demonstrate your ability to write good and well structured code for the platform. You are free to use any libraries, tools or structure you want to implement this app, but **you are not allowed to use Podio's client libraries or SDKs** (i.e. the libraries available at developers.podio.com/clients).

Writing unit tests are not required for the purpose of this exercise, but if you would like to demonstrate your abilities in writing tests you are very welcome to do so.

Make sure you read this description of the assignment carefully or you might miss some important detail.

Documentation

A complete documentation of the Podio API is available at https://developers.podio.com. All API requests return JSON data.

A few tips before starting:

- The base URL for all API requests is **api.podio.com**. The authentication request also works with podio.com as the base URL, but fetching the organization does not.
- Remember that some strings might need to be URL encoded when passed as URL parameters.
- The API returns a very descriptive JSON response for any error that might occur, so make sure to have a close look at the response body JSON if your request fails.

Authentication

Podio uses OAuth2 for authentication, as described here https://developers.podio.com/authentication

As described in the link above, you need an API key and secret for your app to talk to the API. You can make one yourself on Podio (Under "Account Settings" > "API keys"), or use this pre-defined one:

API key: **podio-puzzle**
API secret: **2u8c5SbylhvJ1uzeYMlsNS9fePA6hlkAyGtGjlWaN2r9FrThhcmwBh67EPHUpCHd**

For the purpose of this exercise, use the email/password flow outlined here to obtain an access token https://developers.podio.com/authentication/username_password. Please note that the **redirect_uri** URL parameter is not required for this exercise, only **username**, **password**, **client_id** and **client_secret**.

You do not have to care about the token expiring or take care of refreshing it for this assignment, you are only interested in the **access_token** part of the response.

Once you have obtained an access token for the user, any subsequent API call (such as the one to get all organizations) can be authenticated by setting the **Authorization** HTTP header to be "OAuth2 <access_token>".

To test the log in, use your own Podio account or this demo one:

Email: **sebrehtest+puzzle@gmail.com**
Password: **password1234**

Getting the organizations and workspaces

To fetch the organizations that the user is a member of from the Podio API, use the “Get organizations” API call, documented here https://developers.podio.com/doc/organizations/get-organizations-22344

The response from the server contains a list of all organizations the user is a member of, and each organization has a list of workspaces (the “spaces” attribute of each organization).

The Interface

The app should consist of two main screens, the ***login screen*** and the ***workspaces list screen***.

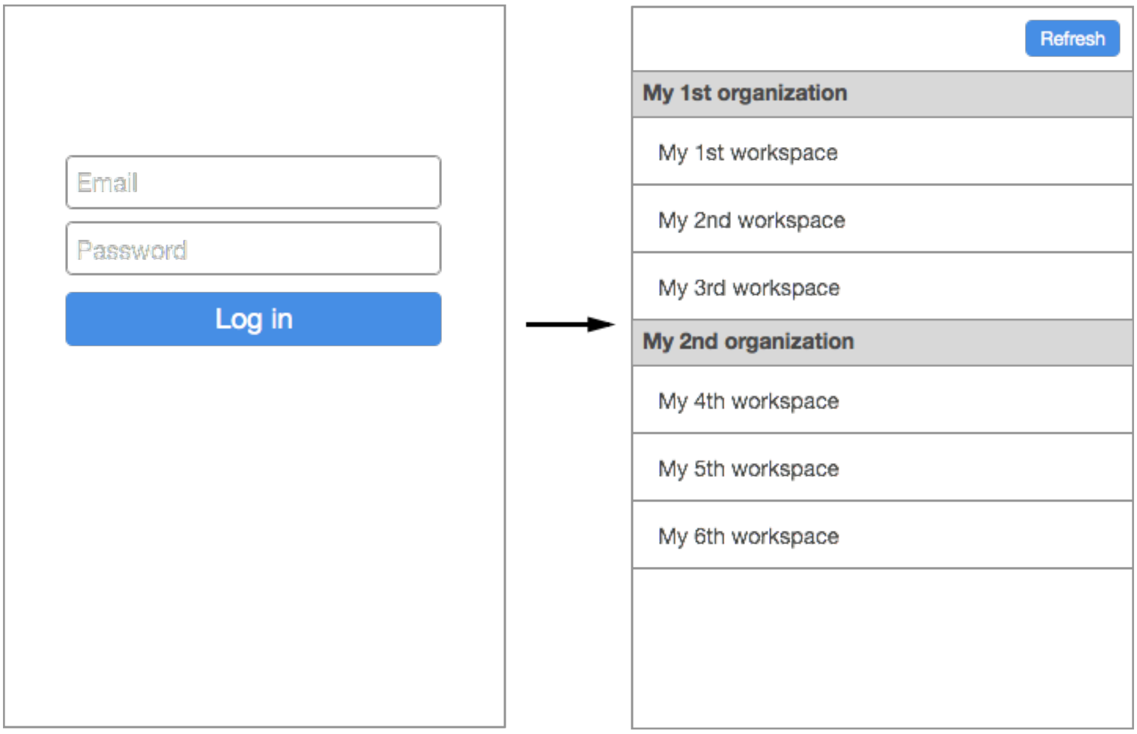
The login screen should contain:

- An email text field
- A password text field
- A “Log in” button that logs the user in of the user once clicked, and then shows the workspace lists

The workspace list screen should:

- Display a list of the names of every workspace that the user is a member of, grouped by the organization, where each group header displays the name of the organization.
- Contain a button to refresh the list of organizations from the API

Below is a wireframe of how it could look:



That’s it :) We are looking forward to your solution. Good luck!