Experiment :: Auth0, Native React, Android, and Firebase Cloud Messaging

Overview

This document provides details about the process taken to learn new technologies and to create a working application. It will help define a target audience and/or persona to create engaging content for an epic blog post.

Viability

The project is certainly viable. The target technologies are will documented and supported by the community. There is plenty of information to learn how to setup a development, configure, and execute development.

Although, there are several new technologies/platforms unfamiliar to me, I have the abilities and skills necessary to learn and implement most if not all of the target requirements within the specified time frame of 8 to 16 hours.

Objective

The objective of the exercise is to learn what it is like to work at and with Auth0, Firebase Cloud Messaging, and React Native. The exercise will target (3) technologies that I do not have any experience using. The exercise should provide Auth0 with details about how I approach learning new technologies. And more importantly, how I will create engaging content for the developer community.

Scope

Create an Android mobile application that provides authentication using Auth0. The application will also have the ability to receive notifications from Firebase Cloud Messaging. The application will use the Native React Framework to implement the features.

Tools and Resource Requirements

- Tools
    - Visual Studio Code
        - React Native Tools for Visual Studio Code
    - React Native
        - Ignite CLI
        - Expo: React Native Toolchain
        - React Navigation
    - Node (Version 8+)
    - Python2
    - Java SE Development Kit 8
    - Android Studio
        - Downloads and CLI
    - Auth0
    - Firebase Cloud Messaging
    - Reactotron: Inspect React and React Native Apps With Ease
        - network event monitoring
        - application state
        - API payload inspection
        - error event monitor
        - logging and log inspection
- Environment Setup
    - React Native Getting Started: Building Projects with Native Code/Quick Start
    - Chocolatey
        - Use to install Java SDK, Python2
        - `choco install -y nodejs.install python2 jdk8`
    - React Native CLI: `npm install -g react-native-cli`
    - Android Studio Setup
    - Auth0
        - Mobile + API: Auth0 Configuration
        - Add Login to Native/Mobile Applications
    - Set up a Firebase Cloud Messaging client app on Android

What I Know?

- OAuth authentication (Google, .NET, and with Angular)
- React Native Items similar to Angular
    - Navigation with stack navigation and actions
    - Environment files for configuration
    - Component-based structure

What I Do Not Know?

- how to setup an Android/React Native solution/project
    - is there a CLI, like Angular?
- can I use Typescript
- how to integrate Auth0 in Android
- Auth0 default token is good for 24 hours (default)

Research

- why are developers using/choosing React (see capabilities)
- what are the available toolchains for this technology stack

Auth0 Capabilities

- authentication
    - user name and password
    - social accounts
    - multi-factor
- single-sign on authentication
- manage Android user profile
- advanced identity solutions
- extended capabilities using Rules

React Native Capabilities

- Use Javascript to create mobile applications for both iOS and Android platforms.
    - Single codebase
    - compiles to native code
- UI Components
- Good community support
- React Native built on top of React
- Widely used by major/large companies
- Component-based, uses environment configuration, stack navigation

Android Capabilities

- requires access to native browser to use Auth0 login form (more secure)

Firebase Cloud Messaging Capabilities

- Send notification messages that are displayed to your user. Or send data messages and determine completely what happens in your application code.

- Distribute messages to your client app in any of 3 ways—to single devices, to groups of devices, or to devices subscribed to topics.
- Send acknowledgments, chats, and other messages from devices back to your server over FCM's reliable and battery-efficient connection channel.

Resources

- React Native: What is it? and, Why is it used?
- React Native Components and APIs
- Here's a list of React Native tools that you can use for your next project
- Auth0: Android Login

Ideation of Concept(s)

- book swap app
- husky/dog meetup app
- fly fishing app

Features

- login page
- account page
    - information about user
    - logout

Target Audience Persona

- a developer who has never developed a mobile app
- a developer that has *not* used React or React Native before (i.e., Angular/Typescript, Microsoft .NET/C#)
- a primary web developer with Javascript and HTML skills
- a developer who has used other/different frameworks (i.e., Ionic, Angular, Vue, etc.)

Content Engagement

- How does a developer that has focused on .NET/C# for the last 18 years and Angular/Typescript for the last 3 years create a mobile application using all new technologies:
    - React Native
    - Android
    - Auth0
    - Firebase Cloud Messaging
- What did I learn exploring new technologies, frameworks, and platforms?
    - React Native has a CLI called Ignite with boilerplate templates

Content Plan (Outline)

- Introduction
    - why web developers need mobile development skills
    - why mobile apps using React Native
- Introduce Application Concept

    - what application will do
    - why authentication and messaging/notifications are necessary
- Overview of tools and resources
    - provide a list of tools
        - required
        - optional
    - provide description of tools and why they are beneficial
- Setup and configuration of development environment
    - review/document sequence of tool setup
    - mostly links to other resources
- Implementation/Execution
    - Auth0
    - Firebase Cloud Messaging
    - Application Features
- Recap
    - Lessons Learned
    - Pros of Chosen Technologies/Frameworks
    - Cons of Chosen Technologies/Frameworks

Plan of Execution

- [ ] crete Auth0 application and credentials
    - [ ] select/use the React Native SDK
    - [ ] configure authentication for social media (target: Github, Google, and Twitter)
- [ ] create new React Native application
    - [ ] clone or use CLI to create new application
    - [ ] define/add the required packages for the application
    - [ ] configure `Android` to allow `WebAuth`
        - [ ] using Auth0 hosted login page via browser
        - [ ] add callback URL to application settings
    - [ ] store Auth0 application credentials in new `environment` file
        - [ ] AUTH0_DOMAIN
        - [ ] AUTH0_CLIENT_ID
        - [ ] AUTH0_AUDIENCE
        - [ ] AUTH0_SCOPE
    - [ ] udpate `App.js` to include
        - [ ] `Root` view/component
        - [ ] any Styles
    - [ ] add navigation to the application's `Root.js` component (stack navigation with actions)
    - [ ] create new `login` component
        - [ ] add styles
        - [ ] retrieve Auth0 configuration
        - [ ] add navigation options
        - [ ] add initialization logic to control view of the `Login`
        - [ ] logic to `render` the Login view

        - [ ] application opens browser with Auth0 hosted page
    - [ ] Auth0 access token storage
        - [ ] add
        - [ ] retrieve
        - [ ] request new token (if token is expired)
            - [ ] retrieve, store,>ma and refresh the `RN` (React Native) application
    - [ ] Auth0
        - [ ] implement `authorization` call
            - [ ] scope
            - [ ] audience
            - [ ] device
            - [ ] prompt
        - [ ] store tokens on successful authentication
            - [ ] access token
            - [ ] refresh token
        - [ ] implement request for user information
    - [ ] implement `logout`
        - [ ] removes tokens from storage

Technical Execution

Work Environment

Repository

Content Execution