Introduction

The ios SDK should be a simple client wrapper that wraps the Inplayer's REST API endpoints in an idiomatic way and provide an abstraction that should require little to no knowledge regarding APIs and REST.

Requirements

The SDK should be easily installable via a package manager such as cocoa pods, carthage etc. It should follow Swift conventions and it should feel natural to the developer using the library.

The API reference contains the needed information on how to get started implementing the endpoints:

https://s3-eu-west-1.amazonaws.com/docs.inplayer.com/latest/index.html

The api relies on JWT (Json web tokens) for identifying and authorizing the user who calls some specific endpoint. Once a user authenticates using the authentication endpoint, the API returns an access token which can then be used to call certain endpoints.

The JWT is sent in the header of each request that requires authorization, in this format:

Authorization: Bearer %token%

Endpoints

API URLs:

- Production - https://services.inplayer.com
- Staging - https://staging-v2.inplayer.com

The SDK should be configurable to either use the production or the staging environment.

Here is a list of endpoints that the SDK should implement:

Accounts

- create
- authenticate
- logout
- get
- update
- erase
- changePassword
- requestForgotPasswordToken
- setNewPassword

Assets

- getAccess

- getItem
- getAccessFees

Payments

- **validateReceipt** - /v2/external-payments/apple/validate (currently not documented)

End-user Payment Flow

The SDK should allow the developer to create a complete purchase flow. The following is an example of a

real world application:

- Access check - once the user clicks the buy button, another access checkup call should be sent to the InPlayer system to confirm the user hasn't purchased access to the asset in the meantime.
- Payment process initiated - the application should proceed with the payment flow via the Apple payment system. That process should result with a confirmation Receipt sent by the Apple system.
- Forward receipt to InPlayer - the Receipt received from the Apple system should be forwarded to the InPlayer system so that information for the payment can be recorded there. The URL where the Receipt should be forwarded is the following - **/v2/external-payments/apple/validate**
- Content display - once the InPlayer system completes the process, access will be granted to the end-user and a notification will be sent to the application. In order to follow this flow, the application should have a websocket established with the InPlayer system through which the notification will be sent.

Notifications

The platform provides real-time notifications via websockets, that are essentialy events that happened in the platform after a certain action. The SDK should provide the developer an easy way to subscribe and listen to notifications.

We rely on the AWS IoT service in order to provide the notifications functionality. The SDK should use the aws-sdk-ios library:

- https://github.com/aws/aws-sdk-ios

The JWT access token is needed in order to fetch temporary credentials to be used when authenticating with the AWS IoT service. Through this endpoint you can get the needed credentials that last 1 hour:

- Production: https://eynmuj2g26.execute-api.eu-west-1.amazonaws.com/prod/iot/keys
- Staging: https://o3871l8vj7.execute-api.eu-west-1.amazonaws.com/staging/iot/keys

You'll need to call this endpoint with the authorization header as it is explained above.