Android SDK for SportsTalk 1.0

Introduction

The Android SDK was ported based on the JavaScript version. It has got all the functionalities exposed by the JS SDK. Some of the main functionalities are listed below:

- Join Room
- List Users
- List Participants
- Send command

SDK Documentation

The main SDK component is SportsTalkClient.java. This class has a constructor which will accept configuration needed to set up this SDK.

```
Final SportsTalkConfig sportsTalkConfig = new SportsTalkConfig();
sportsTalkConfig.setAppId("");
sportsTalkConfig.setApiKey("your sports talk API Key goes here...");
sportsTalkConfig.setApiCallBack()   // sets the ApiCall back here
sportsTalkConfig.setEventHandler() // sets the event handler
sportsTalkConfig.setUser() //user
Final SportsTalkClient sportsTalkClient = new SportsTalkClient(sportsTalkConfig);
```

In order to start this API, you need to add a user to this API with the help of SportsTalkConfig.java as shown below.

Note: SportsTalk SDK, at the time of development does not support creating a main user. So it is the responsibility of the implementation team supply an authenticated user.

Polling

The SDK makes use of polling to get the latest events from the SportsTalk server. The events are captured by the callback methods. The developer needs to register event handler as shown above.

Polling started once you join a room. The developer does not have to start the polling mechanism explicitly.

How to use this SDK

The SDK is exposed as an Android Library and this can be added as a module to your project.

In your MainActivity, please do the following:

```
private void setupSportsTalkAPI() {
        // creates a test user
        User user = new User();
        user.setUserId("001864a867604101b29672e904da688a");
        user.setDisplayName("Aldo");

        SportsTalkConfig sportsTalkConfig       = new SportsTalkConfig();

sportsTalkConfig.setApiKey("vfZSpHsWrkun7Yd_fUJcWAHrNjx6VRpEqMCEP3LJV9Tg"
);
        sportsTalkConfig.setContext(MainActivity.this.getApplicationContext());
        sportsTalkConfig.setUser(user);


         // register events
        eventHandler = new EventHandler() {
           @Override
           public void onEventStart(Event event) {

           }

           @Override
           public void onReaction(Event event) {

           }

           @Override
           public void onAdminCommand(Event event) {

           }

           @Override
           public void onPurge(Event event) {
              Log.d(TAG, " onPurge start ...");
```

```
    }

    @Override
    public void onSpeech(Event event) {
        Log.d(TAG, "Onspeech start...." +  event.getBody());
    }

    @Override
    public void onChat(Event event) {

    }
};

apiCallback = new APICallback() {
    @Override
    public void execute(ApiResult<JSONObject> apiResult, String action) {
        if ("listRooms".equals(action)) {
        } else if ("joinRoom".equals(action)) {

        } else if ("sendCommand".equals(action)) {

        }else if("sendReply".equals(action)) {

        }
        else if ("user".equals(action)) {
         } else if ("listUserMessages".equals(action)) {

        }
    }

    @Override
    public void error(ApiResult<JSONObject> apiResult, String action) {
    }
};
```

The *action* variable is used to identify which is the last action performed by the SDK so that appropriate action can be taken.

You are good to go.

Example Project

There is an Android chat demo using this SportsTalk SDK which you can find in the repository. Please find the screenshot below: