Android App- Java (Android-Studio)

Flyt-Android SDK

1. Here you are required to download the Flyt-Android-SDK based on Android Studio and build your app using it.
2. The SDK has all the required libraries for making REST calls and a websocket connection to FlytPOD already integrated in it.
3. The mainActivity in it shows a sample of how a REST call and a WebSocket call is to be made.
4. Sample REST call to fetch namespace of the flytpod

```
    private class NamespaceRequest extends AsyncTask<Void, Void, String> {

        @Override

        protected String doInBackground(Void... params) {

            try {

                //Rest url

                final String url = "http://"+IP+":9090/ros/get_global_namespace";

                //params in json

                String requestJson = "{}";

                //headers

                HttpHeaders headers = new HttpHeaders();

                headers.setContentType(MediaType.APPLICATION_JSON);


                HttpEntity<String> entity = new
HttpEntity<String>(requestJson,headers);

                //restTemplate object initialise for rest call

                RestTemplate restTemplate = new RestTemplate();

                restTemplate.getMessageConverters().add(new
```

```
StringHttpMessageConverter());
                // make the rest call and recieve the response in "response"
                String response = restTemplate.postForObject(url,entity, String.class);

                return response;
            } catch (Exception  e) {
                Log.e("MainActivity", e.getMessage(), e);
            }

            return null;
        }
        //function called after a successful rest call
        @Override
        protected void onPostExecute(String response) {
            if (response!="") {

                try {
                    //initialise a JSON object with the response string
                    JSONObject resp = new JSONObject(response);
                    //extract the required field from the JSON object

namespace=resp.getJSONObject("param_info").getString("param_value");
                } catch (JSONException  | NullPointerException e) {
            }
        }
}
```

5. Sample websocket call to view roll pitch yaw of FlytPOD.

```
IP=editTextIP.getText().toString();

//Initialise a ros object with websocket url

ros=new Ros("ws://"+IP+":9090/websocket");

ros.connect();
```

Note

The Ros object initialisation is done only once every time the app is run unless you are planning tp connect to multiple FlytPODs.

```
//the namespace(unique for every FlytPOD) fetched from the rest call is used to subscribe to a web socket topic

//the syntax Topic(<ros>, <topic>, <type>, <throttle rate>optional)

topic=new Topic(ros,"/"+namespace+"/mavros/imu/data_euler" , "geometry_msgs/TwistStamped",200);

topic.subscribe(new CallbackRos(){

        //callback method- what to do when messages recieved.

        @Override

        public void handleMessage(JSONObject message){

            try {


updateRoll(message.getJSONObject("twist").getJSONObject("linear").getDouble("x"));


updatePitch(message.getJSONObject("twist").getJSONObject("linear").getDouble("y") );


updateYaw(message.getJSONObject("twist").getJSONObject("linear").getDouble("z"));



            }catch(JSONException e){}

        }

});
```

Note

For updated documentation visit docs.flytbase.com.