

Chat Interface Connection Development Steps

Subscription to WSS Endpoint

1. Get the subscription (realtime) & publishing (request) endpoint
2. In order to subscribe, open up websocket connection to:

```
1 wss://<realtime_endpoint>/header=<B64-ENCODED-HEADER>&payload=e30=
```

a. `realtime_endpoint` is:

```
1 If realtime_endpoint is wss://xmowuyieu5dexm754oebdhs7u.appsync-realtime-api.us-east-1.amazonaws.com/graph
2
3 We will utilize xmowuyieu5dexm754oebdhs7u.appsync-realtime-api.us-east-1.amazonaws.com
```

b. `B64-ENCODED-HEADER` is:

```
1 Convert this JSON to Base64:
2
3 {
4   "host": "<domain of request_api>" # xmowuyieu5dexm754oebdhs7u.appsync-api.us-east-1.amazonaws.com,
5   "x-api-key": "<api_key>"
6 }
```

3. After establishing the connection, the first message to send should be:

```
1 {
2   "type": "connection_init"
3 }
```

As a response you will get:

```
1 {
2   "type": "connection_ack",
3   "payload": {
4     "connectionTimeoutMs": 300000
5   }
6 }
```

4. Now you can establish subscription via sending the GRAPHQL command

```
1 {
2   "id": "<random_uuid>",
3   "type": "start",
4   "payload": {
5     "data": "<GRAPHQL_COMMAND>",
6     "extensions": {
7       "host": "<domain of request_api>" # xmowuyieu5dexm754oebdhs7u.appsync-api.us-east-1.amazonaws.com,
8       "x-api-key": "<api_key>"
9     }
10  }
11 }
```

`GRAPHQL_COMMAND` is:

```
1 {"query": "subscription SubscribeToData {\n  subscribe(name: \"<channel_name>\") {\n    name\n    data\n  }\n}"
```

 Use the `GRAPHQL_COMMAND` as is!

Publishing via GraphQL Endpoint

1. Get the subscription (realtime) & publishing (request) endpoint
2. Do POST request to `request_endpoint`
3. Put `x-api-key` header variable
4. Payload must be like this:

```
1 {
2   "operationName":"sendmessage",
3   "query":"mutation sendmessage { client_publish(message: \"{\\\"channel\\\":\\\"<channel_name>\\\",\\\"message\\\":\\\"<mes
4   "variables":{}}
5 }
```

5. As like subscription, do not change the `query`
6. As response you should get this:

```
1 {
2   "data":{
3     "client_publish":true
4   }
5 }
```