

Launch Back-end Source Code

1. Sample code: <https://github.com/dji-sdk/DJI-Cloud-API-Demo>
2. Need to install Plugin Lombok
3. Change in IntelliJ the config file. /src/main/ressources/application.yml

```
server:
  port: 6789
spring:
  application:
    name: cloud-api-sample
  datasource:
    druid:
      type: com.alibaba.druid.pool.DruidDataSource
      driver-class-name: com.mysql.cj.jdbc.Driver
      url: jdbc:mysql://localhost:3306/cloud_sample?useSSL=false&allowPublicKeyRetrieval=true
      username: root
      password: admin
      initial-size: 10
      min-idle: 10
      max-active: 20
      max-wait: 60000

  redis:
    host: localhost
    port: 6379
    database: 0
    username: # if you enable
    password:
    lettuce:
      pool:
        max-active: 8
        max-idle: 8
        min-idle: 0

  jwt:
    issuer: DJI
    subject: CloudApiSample
    secret: CloudApiSample
    age: 86400

  mqtt:
    protocol: tcp
    host: 192.168.1.12 # 192.168.1.1
    port: 1883
    username: JavaServer
    password: 123456
    client-id: 123456
    # Topics that need to be subscribed when initially connecting to mqtt, multiple topics are divided by ",".
    inbound-topic: sys/product/+/status,thing/product/+/requests
```

Need to change to localhost

MySQL username and password

Redis - localhost

EMQX, the IP-Address from Vite

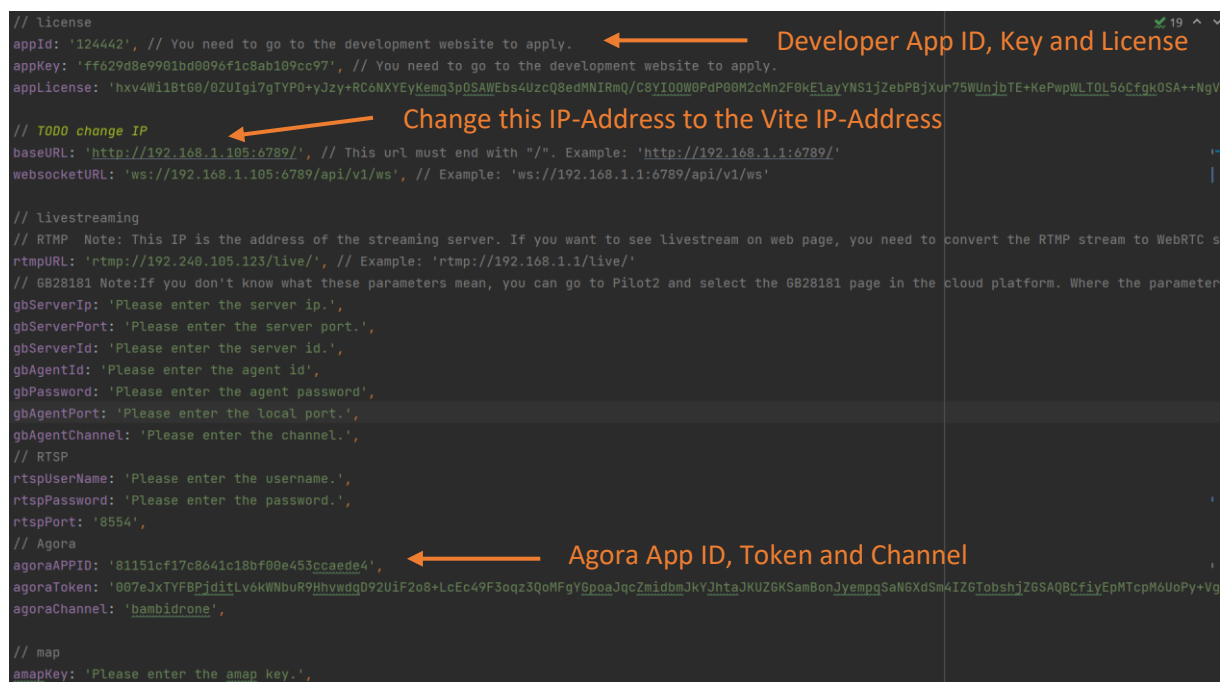
4. Configuring startup.
 - > Add Configuration
 - > +
 - > Application
 - > Main class:
 - > CloudApiSampleApplication

5. Backend started

```
2023-02-02 22:34:31.183 INFO 1348 --- [main] o.s.m.s.b.SimpleBrokerMessageHandler : Started.
2023-02-02 22:34:31.198 INFO 1348 --- [main] c.dji.sample.CloudApiSampleApplication : Started CloudApiSampleApplication in 8.003 seconds (JVM running for 8.621)
```

Launch Front-end Source Code

1. Sample code: <https://github.com/dji-sdk/Cloud-API-Demo-Web>
2. Open the source code using the IDE. /src/api/http/config.ts



The screenshot shows the `config.ts` file in a code editor. It contains configuration for the front-end application, including license information, base URLs, and Agora streaming details. Annotations with orange arrows point to specific fields:

- Developer App ID, Key and License:** Points to the `appId`, `appKey`, and `appLicense` fields in the license section.
- Change this IP-Address to the Vite IP-Address:** Points to the `baseURL` field in the `TODO change IP` section.
- Agora App ID, Token and Channel:** Points to the `agoraAppID`, `agoraToken`, and `agoraChannel` fields in the Agora section.

```
// license
appId: '124442', // You need to go to the development website to apply.
appKey: 'ff629d8e9981bd0096f1c8ab109cc97', // You need to go to the development website to apply.
appLicense: 'hxxv4W118t60/0ZUIgi7gTYP0-yJzy+RC6NXYEyKemq3p0SAWEbs4UzcQ8edMNIrmQ/C8YI00W0PdP00M2cMn2F0kELayYNS1jZebPBjXur75WUUn1bTE+KePwpWLTOL56CfGk0SA++NgV

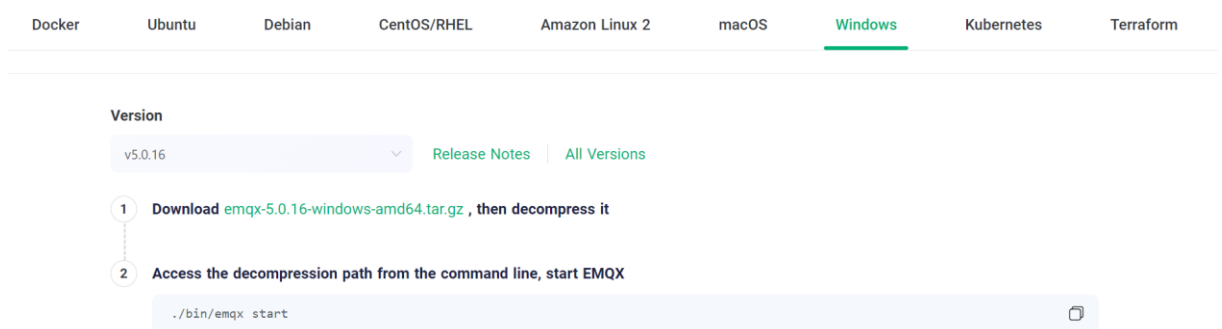
// TODO change IP
baseURL: 'http://192.168.1.105:6789/', // This url must end with "/". Example: 'http://192.168.1.1:6789/'
websocketURL: 'ws://192.168.1.105:6789/api/v1/ws', // Example: 'ws://192.168.1.1:6789/api/v1/ws'

// livestreaming
// RTMP Note: This IP is the address of the streaming server. If you want to see livestream on web page, you need to convert the RTMP stream to WebRTC s
rtmpURL: 'rtmp://192.240.105.123/live/', // Example: 'rtmp://192.168.1.1/live/'
// GB28181 Note: If you don't know what these parameters mean, you can go to Pilot2 and select the GB28181 page in the cloud platform. Where the parameter
gbServerIp: 'Please enter the server ip.',
gbServerPort: 'Please enter the server port.',
gbServerId: 'Please enter the server id.',
gbAgentId: 'Please enter the agent id.',
gbPassword: 'Please enter the agent password.',
gbAgentPort: 'Please enter the local port.',
gbAgentChannel: 'Please enter the channel.',
// RTSP
rtspUserName: 'Please enter the username.',
rtspPassword: 'Please enter the password.',
rtspPort: '8554',
// Agora
agoraAppID: '81151cf17c8641c18bf00e453ccaede4',
agoraToken: '007eJxTYFBPjdiLv6kWNbuR9HhvwDq92UiF2o8+LcEc49F3oqz3QoMFqY6pqaJqcZmidbmJkYJhtaJKUZGKSambonJyempgSaNGXdsM4IZ6TobshjZ6SAQ8CfiyEpHTcpM6UoPy+Vg
agoraChannel: 'bambidrone',

// map
amapKey: 'Please enter the amap key.'
```

4. Open the console at the root of the project and run the command to install the dependencies. (**npm install**)
5. Run the command to start the service. (**npm run serve**)

6. Download EMQX <https://www.emqx.io/downloads?os=Windows>



7. Go in Folder and start EMQX with (`./bin/emqx start`)

8. Test Connections <http://localhost:18083/#/login?to=/connections>

username: admin

password: public

change User to:

username: JavaServer

password: 123456

9. Download Redis

10. Open in Powershell (`wsl -u root`)

11. In Ubuntu download Redis with

```
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list
```

```
sudo apt-get update
```

```
sudo apt-get install redis
```

12. Start Redis with (`sudo service redis-server start`)

13. Install MySQL <https://dev.mysql.com/downloads/installer/>