

## API Calls: How to use

- Useful links
- Sync calls (REST)
  - swagger-codegen
- Async calls (events and state updates)
  - SQS
    - Integration queues
    - Credentials
    - Subscription
      - Native
      - JMS
  - Data Transfer Model
  - APPENDIX A - Event codes translation
  - APPENDIX B
  - Device Type
  - Device States

### Useful links

Name	Link
CAPI Swagger UI	<a href="https://api.ajax.systems/swagger-ui/#/">https://api.ajax.systems/swagger-ui/#/</a>
CAPI Swagger YAML	<a href="https://api.ajax.systems/api/swagger.yaml">https://api.ajax.systems/api/swagger.yaml</a>

### Sync calls (REST)

CAPI exposes set of REST calls that allow to manage the data.

List of all the latest available calls is provided as Swagger UI page:<https://api.ajax.systems/swagger-ui/#/>

Also there is Swagger specification in YAML format the can be found here:<https://api.ajax.systems/api/swagger.yaml>

### swagger-codegen

Swagger provides a tool that allows to generate a client for almost any programming language based on provided specification.

Link to the project:<https://swagger.io/tools/swagger-codegen/>

The command to generate basic Java client may look like:

#### Generate Java Client

```
swagger-codegen generate -i https://api.ajax.systems/api/swagger.yaml -l  
java -o capi-java-client
```

This will generate a new project with different build systems provided - gradle, maven, sbt etc.

### Async calls (events and state updates)

#### SQS

CAPI uses Amazon SQS (Simple Queue Service) for pushing events/updates information to the clients.

#### Integration queues

Here is a list of available for integration queues.

Queue Name	Region	Description
ajax-events-<IntegrationID>-prod.fifo	eu-west-1	Events (alarms and other important notifications)
ajax-updates-<IntegrationID>-prod.fifo	eu-west-1	State updates

#### Credentials

AWS AccessKey/SecretKey pair is used to access SQS. The values should be provided to the client by email.

#### Keys rotation

Please keep in mind that AccessKey/SecretKey pair may be rotated. New keys will be provided before the old ones get expired.

#### Subscription

There are few ways to subscribe to the SQS queue using Amazon official client library - Native and JMS.

Official code examples can be found here:

<https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-sqs.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/code-examples.html>

#### Native

There is a Java library that provides all the required functionality for receiving SQS messages from the queue.

1. Import the library to your project:

pom.xml

```
<!--  
https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-sqs  
-->  
<dependency>  
  <groupId>com.amazonaws</groupId>  
  <artifactId>aws-java-sdk-sqs</artifactId>  
  <version>_version_</version>  
</dependency>
```

- All the available versions can be found here -<https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-sqs>
2. Use simple code snippet to start receiving messages:

TestMessageReceiver.java

```
public class TestMessageReceiver {  
  public static void main(String[] args) {  
    String region = "eu-west-1";  
    // TODO: specify correct queue name
```

```

String queueName = "ajax-<name>-events-demo.fifo";

// Creating credentials from accessKey/secretKey pair
// Use accessKey/secretKey values provided by Ajax
BasicAWSCredentials credentials = new BasicAWSCredentials(
    "accessKey",
    "secretKey"
);

AWSStaticCredentialsProvider credentialsProvider = new
AWSStaticCredentialsProvider(credentials);
AmazonSQS sqs = AmazonSQSClientBuilder.standard()
    .withCredentials(credentialsProvider)
    .withRegion(region)
    .build();

// Getting queueUrl to connect
String queueUrl = sqs.getQueueUrl(queueName).getQueueUrl();
while (true) {
    ReceiveMessageRequest receiveRequest = new
ReceiveMessageRequest(queueUrl)
        .withMaxNumberOfMessages(10)
        .withWaitTimeSeconds(20);

    List<Message> messages =
sqs.receiveMessage(receiveRequest).getMessages();

    if (messages.isEmpty()) {
        continue;
    }

    System.out.println("Received messages: " + messages);

    // Deleting received and processed messages from the
queue
    List<DeleteMessageBatchRequestEntry> deleteEntries =
messages.stream()
        .map(message -> new
DeleteMessageBatchRequestEntry(message.getMessageId(),
message.getReceiptHandle()))
        .collect(Collectors.toList());

    DeleteMessageBatchRequest deleteRequest = new
DeleteMessageBatchRequest(queueUrl)
        .withEntries(deleteEntries);
    List<BatchResultErrorEntry> failed =
sqs.deleteMessageBatch(deleteRequest).getFailed();

    // Checking if there is any issue with deleting
messages

```

```
        if (!failed.isEmpty()) {
            System.out.println("Failed to delete next messages:
" + failed);
        } else {
            System.out.println("Successfully deleted all the
messages.");
        }
    }
```

```
}  
}  
}
```

3. After running the code you should start seeing messages if any come to the queue.

## JMS

Amazon also provides a way to work with SQS using JMS binding.

1. Import the library to the project:

pom.xml

```
<!--  
https://mvnrepository.com/artifact/com.amazonaws/amazon-sqs-java-me  
ssaging-lib -->  
<dependency>  
  <groupId>com.amazonaws</groupId>  
  <artifactId>amazon-sqs-java-messaging-lib</artifactId>  
  <version>_version_</version>  
</dependency>
```

- All the available versions can be found here -<https://mvnrepository.com/artifact/com.amazonaws/amazon-sqs-java-messaging-lib>
2. Use simple code snippet to start receiving messages:

## TestJmsMessageReceiver.java

```
public class TestJmsMessageReceiver {
    public static void main(String[] args) throws Exception {
        String region = "eu-west-1";
        // TODO: specify correct queue name
        String queueName = "ajax-<name>-events-demo.fifo";

        // Creating credentials from accessKey/secretKey pair
        // Use accessKey/secretKey values provided by Ajax
        BasicAWSCredentials credentials = new BasicAWSCredentials(
            "accessKey",
            "accessKey"
        );
        AWSStaticCredentialsProvider credentialsProvider = new
        AWSStaticCredentialsProvider(credentials);
        SQSConnectionFactory connectionFactory = new
        SQSConnectionFactory(
            new ProviderConfiguration(),
            AmazonSQSClientBuilder.standard()
                .withCredentials(credentialsProvider)
                .withRegion(region)
        );

        SQSConnection connection =
        connectionFactory.createConnection();
        Session session = connection.createSession(false,
        Session.AUTO_ACKNOWLEDGE);
        Queue queue = session.createQueue(queueName);
        MessageConsumer consumer = session.createConsumer(queue);

        consumer.setMessageListener(message ->
        System.out.println("Received message: " + message));

        connection.start();
    }
}
```

3. After running the code you should start seeing messages if any come to the queue.

## Data Transfer Model

Messages in events/updates queues have application/json type. The models for those messages defined in swagger.yaml.

There are two models there:

1. IntegrationEvent - model that is published into ajax-{name}-events-prod.fifo queue. IntegrationEvent is async event with metadata which notify client about alarm/malfunction/arming or other types of events generated by hub that is associated with client service users

```

IntegrationEvent {
  userId      string($hex64)
  hubId       string($hex64)
  hubName     string
  sourceId    string
  sourceName  string
  code        string
  sourceRoomName string
  type        string
              Enum:
                [ ALARM, ALARM_RECOVERED, MALFUNCTION, FUNCTION_RECOVERED, SECURITY, COMMON, USER, LIFECYCLE ]
}

```

Where:

userId - ajax user that is associated with your service

hubId - ajax hub that associated with the user

sourceId - id of the event source (device or other ajax entity)

sourceName - user-defined name of the event source e.g. "entrance door protect"

sourceRoomName - name of the room which contains source object, like "garage". Can be null if event source is not attached to room (like user or hub)

code - event code like "M\_01\_20" which means "Door protect reed open". [See APPENDIX A for event codes translation sample.](#)

type - event type. Examples:

ALARM, ALARM\_RECOVERED - like door protect reed was opened/closed

MALFUNCTION, FUNCTION\_RECOVERED - like sensor battery low/ battery ok

SECURITY - hub/group was armed/disarmed

COMMON - like walls which was on/of

USER - like user was added/removed

LIFECYCLE - like sensor/room, was added/removed

EXAMPLE

```

{ "userId" : "634FDCAF", "hubId" : "0000AE01", "hubName" :
  "0000AE01", "sourceId" : "0000AE01", "sourceName" : "Pajama_Enota",
  "type" : "FUNCTION_RECOVERED", "code" : "M_21_05", "sourceRoomName"
  : null }

```

This information should be sufficient for your service to send notifications for clients instantly, or enrich/ignore it on your side.

- IntegrationUpdate - model that is published into ajax-{name}-updates-prod.fifo queue. IntegrationUpdate indicates that hub associated with service user have an update.

```

IntegrationUpdate {
  id          string($hex64)
  userId      string($hex64)
  hubId       string($hex64)
  updates     {
    }
  type        string
              Enum:
                [ Hub, User, Group, Room, Camera, DoorProtect, MotionProtect, FireProtect, GlassProtect, LeakProtect, MotionProtectCurtain, RangeExtender, CombiProtect,
                  FireProtectPlus, Keypad, SpaceControl, MotionProtectPlus, DoorProtectPlus, UniversalDevice, Transmitter, Socket, WallSwitch, Relay, MotionProtectOutdoor, StreetSiren,
                  HomeSiren ]
}

```

Where:

id - id of the updated object (device or other ajax entity)

userId - ajax user that is associated with your service

hubId - ajax hub that is associated with user

type - type of the updated object (hub, sensor, user etc.)

updates - what exactly was updated. structure of single update is field name and value. Like

name:"new object name"

or

temperature:25

EXAMPLE:

```

{"type": "Hub", "id": "0000AE01", "updates": {"averageNoise": -82}, "userId": "634FDCAF", "hubId": "0000AE01"}

```

You can send this data to user if it's online, or mark that user's hub structure was changed and this will force user client to update all structure using https calls

It means that if there is a client or model classes generated from swagger.yaml then next code snippet helps to understand how to get model object from the message payload:

#### 1. Native

##### MessageDeserialize

```
ObjectMapper objectMapper = new ObjectMapper()

.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES,
false);

sqs.receiveMessage(receiveRequest).getMessages().forEach(message ->
{
    try {
        IntegrationEvent event =
objectMapper.readValue(message.getBody(), IntegrationEvent.class);
        System.out.println("Received event: " + event);
    } catch (Exception e) {
        System.out.println("Failed to process message " + e);
    }
});
```

#### 2. JMS

##### JmsMessageDeserialize

```
ObjectMapper objectMapper = new ObjectMapper()

.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES,
false);

consumer.setMessageListener(message -> {
    try {
        if (message instanceof TextMessage) {
            String payload = ((TextMessage) message).getText();
            IntegrationEvent event =
objectMapper.readValue(payload, IntegrationEvent.class);

            System.out.println("Received event: " + event);
        }
    } catch (Exception e) {
        System.out.println("Failed to process message " + e);
    }
});
```



## APPENDIX A - Event codes translation

Here is an example of event codes translation for EN locale in json format but we can provide translations in a big variety of languages and formats.

Placeholders are

"M\_01\_20": "%3\$s: open, %1\$s in %2\$s",

M - signal type, the static parameter for all signal and doesn't change.

01 - device type. More detailed at APPENDIX B

20 - event signal "open"-description is contained in each event

%3 - hub name

%1 - event source object name (e.g. sensor name)

%2 - event room name

```
"M_01_20": "%3$s: open, %1$s in %2$s",
  "M_01_21": "%3$s: closed, %1$s in %2$s",
  "M_01_22": "%3$s: external contact open, %1$s in %2$s",
  "M_01_23": "%3$s: external contact closed, %1$s in %2$s",
  "M_01_24": "%3$s: alarm is detected, roller shutter %1$s in %2$s",
  "M_01_25": "%3$s: connection lost, roller shutter %1$s in %2$s",
  "M_01_26": "%3$s: connection restored, roller shutter %1$s in %2$s",
  "M_02_20": "%3$s: motion detected, %1$s in %2$s",
  "M_03_20": "%3$s: smoke detected, %1$s in %2$s",
  "M_03_21": "%3$s: no smoke detected, %1$s in %2$s",
  "M_03_22": "%3$s: temperature above the threshold value, %1$s in
%2$s",
  "M_03_23": "%3$s: temperature below the threshold value, %1$s in
%2$s",
  "M_03_24": "%3$s: hardware failure, %1$s in %2$s",
  "M_03_25": "%3$s: reset after hardware failure, %1$s in %2$s",
  "M_03_26": "%3$s: smoke chamber dirty, %1$s in %2$s",
  "M_03_27": "%3$s: smoke chamber clean, %1$s in %2$s",
  "M_03_28": "%3$s: low reserve battery charge, %1$s in %2$s",
  "M_03_29": "%3$s: reserve battery charged, %1$s in %2$s",
  "M_03_2A": "%3$s: rapid temperature rise detected, %1$s in %2$s",
  "M_03_2B": "%3$s: rapid temperature rise stopped, %1$s in %2$s",
  "M_03_2C": "%3$s: faulty detector %1$s in %2$s. Please, contact the
service center listed in the warranty card",
  "M_03_2D": "%3$s: the smoke chamber of %1$s in %2$s is OK",
  "M_04_20": "%3$s: glass break detected, %1$s in %2$s",
  "M_04_22": "%3$s: external contact open, %1$s in %2$s",
  "M_04_23": "%3$s: external contact closed, %1$s in %2$s",
  "M_05_20": "%3$s: water leak detected, %1$s in %2$s",
  "M_05_21": "%3$s: no water leak detected, %1$s in %2$s",
  "M_06_20": "%3$s: motion detected, %1$s in %2$s",
  "M_06_22": "%3$s: masking detected, check the %1$s in %2$s",
  "M_06_23": "%3$s: masking is not detected, %1$s in %2$s",
  "M_07_10": "Updating your %1$s firmware",
  "M_07_11": "The firmware of your %1$s has updated successfully",
  "M_07_20": "%3$s: external power failure, %1$s in %2$s",
  "M_07_21": "%3$s: external power restored, %1$s in %2$s",
  "M_08_20": "%3$s: motion detected, %1$s in %2$s",
  "M_08_21": "%3$s: glass break detected, %1$s in %2$s",
```

"M\_09\_20": "%3\$s: smoke detected, %1\$s in %2\$s",  
"M\_09\_21": "%3\$s: no smoke detected, %1\$s in %2\$s",  
"M\_09\_22": "%3\$s: temperature above the threshold value, %1\$s in %2\$s",  
"M\_09\_23": "%3\$s: temperature below the threshold value, %1\$s in %2\$s",  
"M\_09\_24": "%3\$s: hardware failure, %1\$s in %2\$s",  
"M\_09\_25": "%3\$s: reset after hardware failure, %1\$s in %2\$s",  
"M\_09\_26": "%3\$s: smoke chamber dirty, %1\$s in %2\$s",  
"M\_09\_27": "%3\$s: smoke chamber clean, %1\$s in %2\$s",  
"M\_09\_28": "%3\$s: reserve battery low, %1\$s in %2\$s",  
"M\_09\_29": "%3\$s: reserve battery charged, %1\$s in %2\$s",  
"M\_09\_2A": "%3\$s: rapid temperature rise detected, %1\$s in %2\$s",  
"M\_09\_2B": "%3\$s: rapid temperature rise stopped, %1\$s in %2\$s",  
"M\_09\_2C": "%3\$s: faulty detector %1\$s in %2\$s. Please, contact the service center listed in the warranty card",  
"M\_09\_2D": "%3\$s: the smoke chamber of %1\$s in %2\$s is OK",  
"M\_09\_30": "%3\$s: Carbon monoxide () detected, %1\$s in %2\$s",  
"M\_09\_31": "%3\$s: Carbon monoxide (CO) level is OK, %1\$s in %2\$s",  
"M\_0A\_20": "%3\$s: Disarmed using %1\$s",  
"M\_0A\_21": "%3\$s: Armed using %1\$s",  
"M\_0A\_22": "%3\$s: Night mode activated using %1\$s",  
"M\_0A\_23": "%3\$s: Panic button pressed on %1\$s",  
"M\_0A\_24": "%3\$s: unsuccessful arming attempt using %1\$s",  
"M\_0A\_25": "%3\$s: unsuccessful Night mode activation attempt using %1\$s",  
"M\_0A\_26": "%3\$s: the system has been armed with malfunctions using %1\$s",  
"M\_0A\_27": "%3\$s: Night mode activated with malfunctions using %1\$s",  
"M\_0A\_28": "%3\$s: Night mode deactivated using %1\$s",  
"M\_0A\_29": "%3\$s: %2\$s has been disarmed using %1\$s",  
"M\_0A\_2A": "%3\$s: %2\$s has been armed using %1\$s",  
"M\_0A\_2D": "%3\$s: %2\$s has been armed with malfunctions using %1\$s",  
"M\_0A\_2E": "%3\$s: %2\$s unsuccessful arming attempt using %1\$s",  
"M\_0A\_2F": "%3\$s: %2\$s has been disarmed using %1\$s",  
"M\_0A\_30": "%3\$s: Attempt to break the password on %1\$s",  
"M\_0A\_31": "%3\$s: Disarmed using %1\$s",  
"M\_0A\_32": "%3\$s: Night mode deactivated using %1\$s",  
"M\_0B\_02": "%3\$s: %1\$s battery charged",  
"M\_0B\_03": "%3\$s: low battery level in %1\$s",  
"M\_0B\_20": "%3\$s: Disarmed using %1\$s",  
"M\_0B\_21": "%3\$s: Armed using %1\$s",  
"M\_0B\_22": "%3\$s: Night mode activated using %1\$s",  
"M\_0B\_23": "%3\$s: Panic button pressed on %1\$s",  
"M\_0B\_24": "%3\$s: unsuccessful arming attempt using %1\$s",  
"M\_0B\_25": "%3\$s: unsuccessful Night mode activation attempt using %1\$s",  
"M\_0B\_26": "%3\$s: the system has been armed with malfunctions using %1\$s",

"M\_0B\_27": "%3\$s: Night mode activated with malfunctions using %1\$s",  
"M\_0B\_28": "%3\$s: Night mode deactivated using %1\$s",  
"M\_0B\_29": "%3\$s: %2\$s has been disarmed using %1\$s",  
"M\_0B\_2A": "%3\$s: %2\$s has been armed using %1\$s",  
"M\_0B\_2D": "%3\$s: %2\$s has been armed with malfunctions using %1\$s",  
"M\_0B\_2E": "%3\$s: %2\$s unsuccessful arming attempt using %1\$s",  
"M\_0C\_20": "%3\$s: Panic button pressed on %1\$s",  
"M\_0D\_20": "%3\$s: motion detected, %1\$s in %2\$s",  
"M\_0E\_20": "%3\$s: motion detected, %1\$s in %2\$s",  
"M\_0F\_20": "%3\$s: open, %1\$s in %2\$s",  
"M\_0F\_21": "%3\$s: closed, %1\$s in %2\$s",  
"M\_0F\_22": "%3\$s: external contact open, %1\$s in %2\$s",  
"M\_0F\_23": "%3\$s: external contact closed, %1\$s in %2\$s",  
"M\_0F\_24": "%3\$s: alarm is detected, roller shutter %1\$s in %2\$s",  
"M\_0F\_25": "%3\$s: connection lost, roller shutter %1\$s in %2\$s",  
"M\_0F\_26": "%3\$s: connection restored, roller shutter %1\$s in %2\$s",  
"M\_0F\_30": "%3\$s: shock detected by %1\$s in %2\$s",  
"M\_0F\_31": "%3\$s: tilt detected by %1\$s in %2\$s",  
"M\_0F\_32": "%3\$s: the accelerometer of the %1\$s in %2\$s room doesn't work properly. Please, contact the service center listed in the warranty card",  
"M\_0F\_33": "%3\$s: the accelerometer of the %1\$s in %2\$s is OK",  
"M\_11\_20": "%3\$s: alarm is detected, %1\$s in %2\$s",  
"M\_11\_21": "%3\$s: recovered after alarm, %1\$s in %2\$s",  
"M\_11\_22": "%3\$s: alarm is detected, %1\$s in %2\$s",  
"M\_11\_26": "%3\$s: %1\$s was moved in %2\$s",  
"M\_12\_20": "%3\$s: %1\$s disabled in %2\$s, overheated",  
"M\_12\_21": "%3\$s: Temperature is OK, %1\$s in %2\$s",  
"M\_12\_22": "%3\$s: %1\$s enabled in %2\$s",  
"M\_12\_23": "%3\$s: %1\$s disabled in %2\$s",  
"M\_12\_28": "%3\$s: %1\$s disabled in %2\$s, maximum voltage threshold reached",  
"M\_12\_29": "%3\$s: %1\$s disabled in %2\$s, minimum voltage threshold reached",  
"M\_12\_2A": "%3\$s: Voltage is OK, %1\$s in %2\$s",  
"M\_12\_2C": "%3\$s: Relay for %1\$s in %2\$s is not responding. Please run a signal strength test.",  
"M\_13\_20": "%3\$s: motion detected, %1\$s in %2\$s",  
"M\_13\_22": "%3\$s: masking detected, check the %1\$s in %2\$s",  
"M\_13\_23": "%3\$s: masking is not detected, %1\$s in %2\$s",  
"M\_13\_24": "%3\$s: external power failure, %1\$s in %2\$s",  
"M\_13\_25": "%3\$s: external power restored, %1\$s in %2\$s",  
"M\_14\_20": "%3\$s: %1\$s was moved in %2\$s",  
"M\_14\_21": "%3\$s: external power failure, %1\$s in %2\$s",  
"M\_14\_22": "%3\$s: external power restored, %1\$s in %2\$s",  
"M\_1E\_20": "%3\$s: %1\$s disabled in %2\$s, overheated",  
"M\_1E\_21": "%3\$s: Temperature is OK, %1\$s in %2\$s",  
"M\_1E\_22": "%3\$s: %1\$s enabled in %2\$s",  
"M\_1E\_23": "%3\$s: %1\$s disabled in %2\$s",

"M\_1E\_24": "%3\$s: %1\$s disabled in %2\$s, short circuit",  
"M\_1E\_25": "%3\$s: %1\$s disabled in %2\$s, maximum current threshold reached",  
"M\_1E\_26": "%3\$s: %1\$s disabled in %2\$s, user-defined maximum current threshold reached",  
"M\_1E\_27": "%3\$s: Power usage is OK, %1\$s in %2\$s",  
"M\_1E\_28": "%3\$s: %1\$s disabled in %2\$s, maximum voltage threshold reached",  
"M\_1E\_29": "%3\$s: %1\$s disabled in %2\$s, minimum voltage threshold reached",  
"M\_1E\_2A": "%3\$s: Voltage is OK, %1\$s in %2\$s",  
"M\_1E\_2C": "%3\$s: Socket for %1\$s in %2\$s is not responding. Please run a signal strength test.",  
"M\_1F\_20": "%3\$s: %1\$s disabled in %2\$s, overheated",  
"M\_1F\_21": "%3\$s: Temperature is OK, %1\$s in %2\$s",  
"M\_1F\_22": "%3\$s: %1\$s enabled in %2\$s",  
"M\_1F\_23": "%3\$s: %1\$s disabled in %2\$s",  
"M\_1F\_24": "%3\$s: %1\$s in %2\$s hasn't switched off. The relay stopped functioning. Disconnect electrical appliances and replace the WallSwitch.",  
"M\_1F\_25": "%3\$s: %1\$s disabled in %2\$s, maximum current threshold reached",  
"M\_1F\_26": "%3\$s: %1\$s disabled in %2\$s, user-defined maximum current threshold reached",  
"M\_1F\_27": "%3\$s: Power usage is OK, %1\$s in %2\$s",  
"M\_1F\_28": "%3\$s: %1\$s disabled in %2\$s, maximum voltage threshold reached",  
"M\_1F\_29": "%3\$s: %1\$s disabled in %2\$s, minimum voltage threshold reached",  
"M\_1F\_2A": "%3\$s: Voltage is OK, %1\$s in %2\$s",  
"M\_1F\_2C": "%3\$s: WallSwitch for %1\$s in %2\$s is not responding. Please run a signal strength test.",  
"M\_21\_00": "%1\$s: external power failure",  
"M\_21\_01": "%1\$s: external power restored",  
"M\_21\_02": "%1\$s: battery low",  
"M\_21\_03": "%1\$s: battery charged",  
"M\_21\_04": "%1\$s: lid open",  
"M\_21\_05": "%1\$s: lid closed",  
"M\_21\_06": "%1\$s: GSM signal level poor",  
"M\_21\_07": "%1\$s: GSM signal level OK",  
"M\_21\_08": "%1\$s: radio-frequency interference level is high",  
"M\_21\_09": "%1\$s: radio-frequency interference level is OK",  
"M\_21\_0A": "%1\$s: Hub is offline. Check mobile signal strength and the Ethernet port",  
"M\_21\_0B": "%1\$s: Hub is online again",  
"M\_21\_0C": "%1\$s: turned off",  
"M\_21\_10": "%1\$s: updating firmware...",  
"M\_21\_11": "%1\$s: firmware updated",  
"M\_21\_12": "%1\$s: Malfunction!",  
"M\_21\_13": "%1\$s: connection to the monitoring station is lost.

Check mobile signal strength and the Ethernet port on the Hub",  
"M\_22\_00": "%3\$s: Disarmed by %1\$s",  
"M\_22\_01": "%3\$s: Armed by %1\$s",  
"M\_22\_02": "%3\$s: Night mode activated by %1\$s",  
"M\_22\_03": "%3\$s: %1\$s pressed the panic button",  
"M\_22\_07": "%3\$s: new user %1\$s has been added",  
"M\_22\_08": "%3\$s: user %1\$s has been removed",  
"M\_22\_09": "%2\$s has allowed access to the hub settings to %1\$s  
(%4\$s) for %3\$s hours",  
"M\_22\_0A": "%2\$s has allowed a permanent access to the hub settings  
to %1\$s (%4\$s)",  
"M\_22\_0B": "%2\$s has denied access to the hub settings to %1\$s  
(%4\$s)",  
"M\_22\_0D": "PRO user has requested access to the hub settings",  
"M\_22\_24": "%3\$s: unsuccessful arming attempt by %1\$s",  
"M\_22\_25": "%3\$s: unsuccessful Night mode activation attempt by  
%1\$s",  
"M\_22\_26": "%3\$s: the system has been armed with malfunctions by  
%1\$s",  
"M\_22\_27": "%3\$s: Night mode activated with malfunctions by %1\$s",  
"M\_22\_28": "%3\$s: Night mode deactivated by %1\$s",  
"M\_22\_29": "%3\$s: %2\$s has been disarmed by %1\$s",  
"M\_22\_2A": "%3\$s: %2\$s has been armed by %1\$s",  
"M\_22\_2D": "%3\$s: %2\$s has been armed with malfunctions by %1\$s",  
"M\_22\_2E": "%3\$s: %2\$s unsuccessful arming attempt by %1\$s",  
"M\_22\_2F": "%3\$s: %2\$s has been disarmed by %1\$s",  
"M\_22\_31": "%3\$s: Disarmed by %1\$s",  
"M\_22\_32": "%3\$s: Night mode deactivated by %1\$s",  
"M\_23\_08": "%3\$s: new group %1\$s has been added",  
"M\_23\_09": "%3\$s: group %1\$s has been removed",  
"M\_24\_08": "%3\$s: room %2\$s has been added",  
"M\_24\_09": "%3\$s: room %2\$s has been removed",  
"M\_25\_08": "%3\$s: new camera %1\$s has been added",  
"M\_25\_09": "%3\$s: camera %1\$s has been removed",  
"M\_26\_00": "%3\$s: lid open, %1\$s in %2\$s",  
"M\_26\_01": "%3\$s: lid closed, %1\$s in %2\$s",  
"M\_26\_20": "%3\$s: alarm is detected, %1\$s in %2\$s",  
"M\_26\_21": "%3\$s: recovered after alarm, %1\$s in %2\$s",  
"M\_26\_22": "%3\$s: alarm is detected, %1\$s in %2\$s",  
"M\_26\_23": "%3\$s: external contact is shorted out, %1\$s in %2\$s",  
"M\_26\_24": "%3\$s: external contact is OK, %1\$s in %2\$s",  
"M\_ABS\_00": "%3\$s: lid open, %1\$s in %2\$s",  
"M\_ABS\_01": "%3\$s: lid closed, %1\$s in %2\$s",  
"M\_ABS\_02": "%3\$s: battery charged, %1\$s in %2\$s",  
"M\_ABS\_03": "%3\$s: low battery, %1\$s in %2\$s",  
"M\_ABS\_04": "%3\$s: connection lost, %1\$s in %2\$s",  
"M\_ABS\_05": "%3\$s: connection restored, %1\$s in %2\$s",  
"M\_ABS\_06": "%3\$s: synchronization failure, %1\$s in %2\$s",  
"M\_ABS\_07": "%3\$s: synchronization OK, %1\$s in %2\$s",  
"M\_ABS\_08": "%3\$s: %1\$s has been added successfully",

```
"M_ABS_09": "%3$s: %1$s has been removed",  
"M_ABS_10": "Updaiting your %1$s firmware",  
"M_ABS_11": "The firmware of your %1$s has updated successfully",
```

```
"M_ABS_12": "%3$s: %1$s in %2$s has detected a malfunction",
"M_ABS_13": "%1$s turned off"
```

## APPENDIX B

### Device Type

Device number	Device Type
01	Door Protect
02	Motion Protect
03	Fire Protect
04	Glass Protect
05	Leak Protect
06	Motion Protect Curtain
07	Range Extender
08	CombiProtect
09	Fire Protect Plus
0A	Keypad
0B	Space Control
0C	Button
0D	MotionCam
0E	Motion Protect Plus
0F	Door Protect Plus
11	Transmitter
12	Relay
13	Motion Protect Outdoor
14	Street Siren
15	Home Siren
1F	Wall Switch

### Device States

State	Description
0	PASSIVE
1	ACTIVE
2	DETECTION_AREA_TEST
3	RADIO_CONNECTION_TEST
4	WAIT_RADIO_CONNECTION_TEST_START
5	WAIT_RADIO_CONNECTION_TEST_END
6	WAIT_DETECTION_AREA_TEST_START
7	WAIT_DETECTION_AREA_TEST_END
8	WAIT_REGISTRATION

1E	Socket
22	User
23	Group
24	Room
25	Camera
ABS	Common part for all devices