Kronos Tools

Various utilities for Kronos.

Git: https://git.arrowconnect.io/kronos/kronos-tools

Name	Description
data-extract	Command line tool to extract telemetry data for multiple devices.
device-and-fwm-job-creation-automation	Device and firmware management job creation automation tool.
simulator-api-client	A tool that simulates multiple API client instances in separate threads.
simulator-gateway	A tool that simulates running multiple gateway instances in separate threads.
social-event-device-import	Command line tool to upload MAC to pin-code spreadsheets to the social_event_device collection.

data-extract

Requirements: ⚠ KR-2882 - JIRA issue doesn't exist or you don't have permission to view it.

Command line tool to extract telemetry data for multiple devices.

The application requests telemetry data through **acn-sdk-java TelemetryApi** by applicationHid, nodeHid or deviceHids and saves it in one of the following formats: comma delimited CSV, JSON, XML.

Build

```
cd data-extract
gradlew bootRepackage
```

Run

```
cd build/libs
java -jar dext-1.0.jar [options]

Options:
 --from=<ISO-8601 formatted date and time, e.g.
2017-09-21T12:42:54.670Z>, mandatory
 --to=<ISO-8601 formatted date and time, e.g. 2017-09-21T12:42:54.670Z>,
mandatory
 --file=<output file path, e.g. dump.csv>, mandatory

 --applicationHid=<application hid>

 --nodeHid=<node hid>

 --deviceHids=<comma delimited list of device hids>

  either applicationHid, nodeHid or deviceHids must be set

 --format=[CSV|JSON|XML], CSV is the default
 --acn.baseUrl=<Kronos API base URL>
 --acn.apiKey=<AES-256 encrypted api key>
 --acn.secretKey=<AES-256 encrypted secret key>
 --spring.config.location=<application.properties file location>
```

Examples

application.properties

```
acn.baseUrl=http://pgsdev01.arrowconnect.io:12001
acn.apiKey=BVE8AQQYEScaDUJiJhlDVA0uOR82AEdLFlRnNiwJTTM3KhQBfBY3OGU4KHsHY
FQmPQI/AUQQAx4FNTkDHAttAxR7Hhp1eS0OAWMxCmEmFTEwcgMjJnlHLgouIm4sHDURDyF8b
zNDDwUQCgIDUBI7S3sOBTtyEgsFHDJRQB43XQBDQgIE
acn.secretKey=DmsaJxwoOUQLVkM5MTkHVn4tExovJkdLJUxbUjo+Q1IWLGANAWwoHFdBOl
sQGAFmRiAYHl4gcSEhQFpjMxwOCjp/QQkVUC4EGXcuDRwFBj8QLz4MehBYBA==
```

CSV

```
java -jar dext-1.0.jar
--spring.config.location=file:./application.properties
--from=2018-11-13T00:00:00.Z --to=2018-11-14T00:00:00.Z
--file=output.csv
--deviceHids=f3db2dba8578f125e6b1cd592ea3393bcaeb3f17,6d5f861a75bdf74345
0ca902e6f6866381da66ae
```

Output format

JSON

```
java -jar dext-1.0.jar
--spring.config.location=file:./application.properties
--from=2018-11-13T12:00:00.Z --to=2018-11-13T13:00:00.Z
--file=output.json
--deviceHids=f3db2dba8578f125e6b1cd592ea3393bcaeb3f17,6d5f861a75bdf74345
0ca902e6f6866381da66ae --format=JSON
```

Output format

XML

```
java -jar dext-1.0.jar
--spring.config.location=file:./application.properties
--from=2018-11-13T12:00:00.Z --to=2018-11-13T13:00:00.Z
--file=output.xml
--deviceHids=f3db2dba8578f125e6b1cd592ea3393bcaeb3f17,6d5f861a75bdf74345
0ca902e6f6866381da66ae --format=XML
```

Output format

device-and-fwm-job-creation-automation

Requirements: ⚠ KNP-275 - JIRA issue doesn't exist or you don't have permission to view it.

Device and firmware management job creation automation tool.

Build

```
cd device-and-fwm-job-creation-automation
gradlew bootRepackage
```

Run

```
cd build/libs
java -jar automatedScript-1.0.jar [options]

Options:
 --acn.baseUrl=<Kronos API base URL>
    --acn.apiKey=<AES-256 encrypted api key>
    --acn.secretKey=<AES-256 encrypted secret key>
    --spring.config.location=<application.properties file location>

 --taskName=HELP - help

 --taskName=CREATE_AND_EXPORT - create and export devices
 --file=<output file path, e.g. devices.csv>, mandatory
 --nodeHid=<node hid>
 --userHid=<user hid>
 --softwareName=<software name>
 --softwareVersion=<software version>
 --gatewayHids=<comma delimited list of gateway hids>
 --deviceTypeName=<device type name>
 --numberOfDevicesPerGateway=<number of devices per gateway>, 1 is the
default

 --taskName=FW - create a job and manage transactions
 --file=<input file path, e.g. devices.csv>, mandatory
 --timeToExpireMinutes=<expiration value for the Job, in minutes>,
mandatory
 --softwareReleaseHid=<software release hid>
 --userHid=<user hid>
 --numberOfDevices=<number of devices to put into new Job>
 --numberOfReceivedTrans=<number of transactions to be in Received
state>
 --numberOfErrorTrans=<number of transactions to be in Error state>
 --numberOfCompletedTrans=<number of transactions to be in Completed
state>

 --taskName=FW_CREATE_ONLY - create a job
 --file=<input file path, e.g. devices.csv>, mandatory
 --timeToExpireMinutes=<expiration value for the Job, in minutes>,
mandatory
 --softwareReleaseHid=<software release hid>
 --numberOfDevices=<number of devices to put into new Job>
 --onDemand=[true, false], true by default
 --scheduledDate=<date and time of starting job, if onDemand=false>, for
example: 2018-07-20T10:15:30.00Z
 --deviceTypeHid=<device type hid>
 --hardwareVersionHid=<hardware version hid>
 --repeat=<number of jobs to create>, 1 by default
```

Examples

CREATE_AND_EXPORT

application.properties

```
taskName=CREATE_AND_EXPORT

# test device generation
file=d:/test.csv
gatewayHids=b53cd90e5847906212312cc8bca5e41fd8534ff5,d4ba62663041b6b245b93a7b2c7f7f174deba469
deviceTypeName=automation-test-asset-type
softwareName=automation-software-product
softwareVersion=1.2.3
userHid=7640f92854c25f8567d9d15a339a6f8f6c2c3f68
nodeHid=dacc67de7e85ee46bfdbb626acb7ab1eee1bccc2
numberOfDevicesPerGateway=4
```

```
java -jar automatedScript-1.0.jar
--spring.config.location=file:./application.properties
```

test.csv

FW

application.properties

```
taskName=FW

# test Job creation and transaction management
file=d:/test.csv
timeToExpireMinutes=2
softwareReleaseHid=8bccb77bfe1d51314a6117fbcccbc57773908ae0
numberOfDevices=1
numberOfReceivedTrans=0
numberOfErrorTrans=1
numberOfCompletedTrans=1
```

FW_CREATE_ONLY

application.properties

```
taskName=FW_CREATE_ONLY

# test job creation
file=d:/test.csv
timeToExpireMinutes=2
softwareReleaseHid=8bccb77bfe1d51314a6117fbcccbc57773908ae0
numberOfDevices=1
deviceTypeHid=80decd12519ba33316c1b0d8e6db641e5bbd1db3
hardwareVersionHid=f535204cdf241b6f6f0c1841d3c7f28b7612b004
onDemand=true
repeat=1
```

simulator-api-client

Requirements: ⚠ KR-2911 - JIRA issue doesn't exist or you don't have permission to view it.

A tool that simulates multiple API client instances in separate threads.

The application uses simulator-gateway to register gateways and devices and send telemetry. API client simulator runs multiple threads each of which runs a loop of randomly chosen tasks:

- query gateways
- query devices
- query telemetries
- delete gateways
- delete devices
- stream telemetries

Build

```
cd simulator-api-client
gradlew bootRepackage
```

Run

```
cd build/libs
java -jar simulator-api-client-1.0.jar
--spring.config.location=<application.properties file location>
```

application.properties

```
# Arrow Connect API configuration
acn.baseUrl=http://pgsdev01.arrowconnect.io:12001
acn.baseWebSocketUrl=ws://pgsdev01.arrowconnect.io:12001
acn.apiKey=B3QRNwEgZh4jIQEXFyxGcB4OIXsVJkdLEAteAnULcBFlCjd/QTkAGXNcLHkgF
ilmDDAEHXxhfDQ1Bgl9DzcBNwAABQsVdzUJF1UKEDYNfDJDbSEBBCRAOCosG3QLBRNQChoPO
jFxDgc+FTQZIXVEAixyJQM0IQwyPAUDOTsIPw8jfHsE
acn.secretKey=MUsgWG4bFz0ZKxxlLWA8X2UHJggJJkdLNFZoUjl1AFcxPGw9QwMGAVlCL1
gFGgQPGxInEnMiFGIHOztDWDxUCToBWzweAnwmZnwzEyFTdRhCclYyFy15BQ==

# Simulator configuration
simulator.iotConnectMqtt=tcp://pgsdev01.arrowconnect.io:1883
simulator.iotConnectMqttVirtualHost=/themis.dev
simulator.threads=100
simulator.maxRegisterTimeoutMs=500
simulator.maxGatewayCheckinTimeoutMs=100
simulator.gatewayHeartbeatTimeoutMs=60000
simulator.gateways=3
simulator.devices=3
simulator.loops=10
simulator.delayBetweenLoops=5000

# Simulator API client configuration
## number of API client threads
simulator.api-client.threads=3
## number of requests each thread sends to the API
simulator.api-client.loops=10
## delay in milliseconds between API requests
simulator.api-client.delayBetweenLoops=5000
## maximum time in milliseconds per websocket subscription to receive
telemetry stream
simulator.api-client.maxStreamTimeoutMs=60000
## maximum number of days which can be used in telemetry requests
simulator.api-client.maxTelemetryDays=7
```

simulator-gateway

Requirements: ⚠ KR-2903 - JIRA issue doesn't exist or you don't have permission to view it.

A tool that simulates running multiple gateway instances in separate threads.

Build

```
cd simulator-gateway
gradlew bootRepackage
```

Run

```
cd build/libs
java -jar simulator-gateway-1.0.jar
--spring.config.location=<application.properties file location>
```

application.properties

```
# Arrow Connect API configuration
acn.baseUrl=http://pgsdev01.arrowconnect.io:12001
acn.apiKey=B3QRNwEgZh4jIQEXFyxGcB4OIXsVJkdLEAteAnULcBFlCjd/QTkAGXNcLHkgF
ilmDDAEHXxhfDQ1Bgl9DzcBNwAABQsVdzUJF1UKEDYNfDJDbSEBBCRAOCosG3QLBRNQChoPO
jFxDgc+FTQZIXVEAixyJQM0IQwyPAUDOTsIPw8jfHsE
acn.secretKey=MUsgWG4bFz0ZKxxlLWA8X2UHJggJJkdLNFZoUjl1AFcxPGw9QwMGAVlCL1
gFGgQPGxInEnMiFGIHOztDWDxUCToBWzweAnwmZnwzEyFTdRhCclYyFy15BQ==

# Simulator configuration
## MQTT server URI
simulator.iotConnectMqtt=tcp://pgsdev01.arrowconnect.io:1883
## RabbitMQ virtual host
simulator.iotConnectMqttVirtualHost=/themis.dev
## number of threads in the therad pool
simulator.threads=100
## maximum amount of time in milliseconds before a gateway register
request is sent to Kronos API
simulator.maxRegisterTimeoutMs=500
## maximum amount of time in milliseconds before a gateway checkin
request is sent to Kronos API
simulator.maxGatewayCheckinTimeoutMs=100
## delay in milliseconds between heartbits that each gateway sends to
the cloud
simulator.gatewayHeartbeatTimeoutMs=60000
## number of gateways to be registered
simulator.gateways=3
## number of devices to be registered
simulator.devices=3
## number of telemetries to be sent to the cloud for each device
simulator.loops=10
## delay in milliseconds between telemetries
simulator.delayBetweenLoops=300
```

social-event-device-import

Requirements: ⚠ KR-4486 - JIRA issue doesn't exist or you don't have permission to view it.

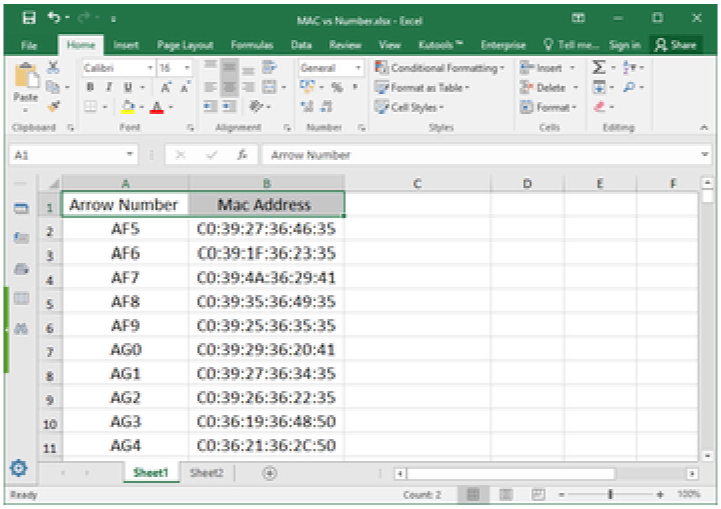
Command line tool to upload MAC to PIN-code spreadsheets to the **social_event_device** collection.

The application sends a POST HTTP request to the hidden **SocialEventDeviceApi** endpoint **<baseUrl>/api/v1/kronos/social/event/devices/import/xls.** An access key used for the request should have **application write privileges**.

Spreadsheet file notes:

- A file containing MAC addresses mapping data should be in .xls (.xlsx) format.
- If it has multiple sheets, only the first one will be handled.
- The first row is expected to be a header and it will be skipped.

- The first column contains Arrow numbers (PIN-codes).
- The second column contains MAC addresses.
- If a particular MAC address already exists in the database, it will be skipped.



Arrow Number	Mac Address
AF5	C0:39:27:36:46:35
AF6	C0:39:1F:36:23:35
AF7	C0:39:4A:36:29:41
AF8	C0:39:35:36:49:35
AF9	C0:39:25:36:35:35
AG0	C0:39:29:36:20:41
AG1	C0:39:27:36:34:35
AG2	C0:39:26:36:22:35
AG3	C0:36:19:36:48:50
AG4	C0:36:21:36:2C:50

Build

```
cd social-event-device-import
gradlew bootRepackage
```

Run

```
cd build/libs
java -jar social-event-device-import-1.0.jar [options]

Options:
 --file=<path to xls file>, example: C:\MAC vs Number.xlsx
 --device-type=<device type name>, example: simba-pro
 --kronos.client.baseUrl=<Kronos API base URL>, example:
http://pgsdev01.arrowconnect.io:12001
 --kronos.client.apiKey=enc:<AES-256 encrypted api key>
 --kronos.client.secretKey=enc:<AES-256 encrypted secret key>
 --spring.config.location=<application.properties file location>
```

Examples

```
java -jar social-event-device-import-1.0.jar --file="C:\MAC vs
Number.xlsx" --device-type=simba-pro
--spring.config.location=file:./application-pgsdev01.properties
```

application-pgsdev01.properties

```
kronos.client.baseUrl=http://pgsdev01.arrowconnect.io:12001
kronos.client.apiKey=enc:P38vF3MEKhcoUVkZChJFXzAEJjh+JkdLTg1AKCwCZSA5Lg4
vQj0WelQcIB0LCgk8D3kyBgY3H385XA1iCShnDFg+OA4AQXItB38uESo1cwgzbTc6cBFfOnA
GIG4BHDcyEy8HEhN5FhgZDhkSNzMIVQQIDD5zRCAfGEddOGIRAA4DTGcE
kronos.client.secretKey=enc:IVooG3A2aggZDkMcUxUWV3oUCn5/AEdLHCFUChwjbVMl
KmkEZhILCFEGPgQbfAcHITEFCVU+ND4KG0dBUjxkUC84BAkPXRwKYXUPcQUwKBA8DjUQNxRk
NA==
```

Logging

```
java -Dlogging.config=file:./log4j2.xml -jar
social-event-device-import-1.0.jar --file="C:\MAC vs Number.xlsx"
--device-type=simba-pro
--spring.config.location=file:./application-pgsdev01.properties
```

log4j2.xml

app.log