

# ICC API Specification

1. ICC APIs	3
1.1. Error Response	3
1.2. Data Object	4
1.3. Filter Management	4
1.3.1. filter_object	4
1.3.2. GET /icc/v1/filters	5
1.3.3. GET /icc/v1/filters/{filter_id}	6
1.3.4. POST /icc/v1/filters	6
1.3.5. POST /icc/v1/filters/preview	8
1.3.6. PUT /icc/v1/filters/{filter_id}	10
1.3.7. DELETE /icc/v1/filters/{filter_id}	11
1.3.8. POST /icc/v1/filters/{filter_id}/copy	12
1.4. Filter Device Management	13
1.4.1. device_object	13
1.4.2. GET /icc/v1/filters/{filter_id}/devices_snapshot	13
1.4.3. POST /icc/v1/filters/{filter_id}/devices/refresh	14
1.4.4. GET /icc/v1/filters/{filter_id}/devices/export	14
1.5. Job Management	15
1.5.1. job_object	15
1.5.2. GET /icc/v1/jobs	15
1.5.3. GET /icc/v1/jobs/{job_id}	17
1.5.4. POST /icc/v1/jobs	17
1.5.5. PUT /icc/v1/jobs/{job_id}	18
1.5.6. DELETE /icc/v1/jobs/{job_id}	20
1.5.7. POST /icc/v1/jobs/{job_id}/copy	20
1.5.8. Schedule Object	21
1.5.9. POST /icc/v1/jobs/{job_id}/schedule	22
1.5.10. GET /icc/v1/jobs/{job_id}/schedule	24
1.5.11. DELETE /icc/v1/jobs/{job_id}/schedule	24

1.5.12. POST /icc/v1/jobs/express	25
1.6. Job Device Management	26
1.6.1. GET /icc/v1/jobs/{job_id}/devices_snapshot	26
1.6.2. GET /icc/v1/jobs/{job_id}/devices/{dsn}/details	27
1.7. Job Operation Management	28
1.7.1. POST /icc/v1/jobs/{job_id}/start	28
1.7.2. POST /icc/v1/jobs/{job_id}/cancel	29
1.7.3. (internal) POST /icc/internal/jobs/{job_id}/execute	29
1.8. Action: Set Properties	30
1.8.1. GET /icc/v1/jobs/{job_id}/properties	30
1.8.2. POST /icc/v1/jobs/{job_id}/properties	31
1.9. Action: File Management	31
1.9.1. file_object & file_property	31
1.9.2. POST /icc/v1/files	32
1.9.3. POST /icc/v1/files/{file_id}/upload	33
1.9.4. GET /icc/v1/files	34
1.9.5. PUT /icc/v1/files/{file_id}	35
1.9.6. DELETE /icc/v1/files/{file_id}	35
1.9.7. GET /icc/v1/files/{file_id}	36
1.9.8. GET /icc/v1/files/{file_id}/download	37
1.9.9. POST /icc/v1/jobs/{job_id}/file_property	37
1.9.10. GET /icc/v1/jobs/{job_id}/file_property	38
1.10. OTA Action	39
1.10.1 OTA Object	39
1.10.2 POST /icc/v1/jobs/{job_id}/ota	40
1.10.3 GET /icc/v1/jobs/{job_id}/ota	41
1.10.4. GET /icc/v1/otas	42
1.10.5. POST /icc/v1/oems/{oem}/otas/{ota_id}/start	43
1.10.6. POST /icc/v1/oems/{oem}/otas/{ota_id}/stop	44
1.11. User Consent	44
1.11.1 Device Job Object	44
1.11.2 GET /icc/v1/jobs/{job_id}/consents	45
1.11.3 Mobile Usage API	46

1.11.3.1 GET /icc/v1/users/device_jobs	46
1.11.3.2 PUT /icc/v1/dsns/{dsn}/device_jobs/{device_job_id}	47
1.11.3.3 PUT /icc/v1/users/device_jobs	48
1.12 Stage Run	49
1.12.1 Stage Object	49
1.12.2 POST /icc/v1/jobs/{job_id}/stages	49
1.12.3 GET /icc/v1/jobs/{job_id}/stages	50
1.12.4 GET /icc/v1/jobs/{job_id}/stages/{stage_number}/devices	51
1.12.5 DELETE /icc/v1/jobs/{job_id}/stages/{stage_number}	52
1.12.6 DELETE /icc/v1/jobs/{job_id}/stages	52
2.1. searchservice	53
2.1.1. (internal) POST /iccsearch/internalv1/advanced_search	53
2.2. datapointservice	55
2.2.1. (internal) POST /dp-svc/internalv1/advanced_search	55
2.3. deviceservice	56
2.3.1. (internal) POST /deviceservice/internal/v1/oems/{oem}/properties/{property_name}/datapoints.json	56
2.4 Schedule service	57
2.4.1 (internal) POST /iccschedule/v1/schedules	58
2.4.2 (internal) DELETE /iccschedule/v1/schedules/{job_id}	59
2.5. otaservice	59
2.5.1. (internal) POST /otaservice/v1/internal/oems/{:oem_str}/otas/{:ota_id}/start	59
2.5.2. (internal) POST /otaservice/v1/internal/oems/{:oem_str}/otas/{:ota_id}/stop	60
2.5.3. (internal) POST /otaservice/v1/internal/otas/{:ota_id}/ota_devices	61
Appendix A Error Code Table	62

# 1. ICC APIs

## 1.1. Error Response

Example:

```
{"errors": [{"code": "ICC-4000", "message": "page: Invalid"}]}
```

,

## 1.2. Data Object

Format:

```
{
  "key": "object key",
  "value": "object value",
  "op": "=", (optional)
  "metadata": [{"key": "m key", "value": "m value"}] (optional)
}
```

op: >, <, =, !=, >=, <=, contain, not\_contain

Attribute keys: firmware\_model, firmware\_version, template\_id, device\_type, device\_name

Status keys: device\_status, connection\_status, connected\_at, activated\_at, provisioned\_at

## 1.3. Filter Management

### 1.3.1. filter\_object

```
{
  "id": 123,
  "oem_model": "linuxex1",
  "oem_version": "v1.2.3.4",
  "name": "V2 Washers",
  "attributes": [
    {"key": "firmware_model", "op": "=", "value": "linuxex1"}
  ],
  "dsns": {
    "match": ["A005"],
    "include": ["AC001", "AC002"],
    "exclude": ["AC003", "AC004"]
  },
  "device_metadata": [{"key": "m key", "value": "m value"}],
  "properties": [{"key": "prop1", "op": "=", "value": true, "metadata": [{"key": "k", "value": "v"}]}],
  "status": [{"key": "connected_at", "op": ">=", "value": "2018-12-20T09:06:51+0000"}],
  "created_at": "2018-12-20T09:06:51+0000",
  "updated_at": "2018-12-20T09:06:51+0000",
  "description": "test",
  "filter_metadata": [{"key": "m key", "value": "m value"}]
}
```

### 1.3.2. GET /icc/v1/filters

## Get the list of filters

OEM::Admin and Ayla users can access.

Query Param:

oem	Optional	For Ayla user only
oem_model	Optional	
page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100
sort_by	Optional	The name of field that the returned list to be sorted by
direction	Optional	Sorting direction (ASC DESC)

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/filters?
page=1&per_page=15&sort_by=name&direction=DESC'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5001 ICC-5101 ICC-5102 ICC-5201	resource: Not found page: Invalid per_page: minimum 5 per_page: maximum 100 Oem: invalid

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end count on page": 10,
```

1

```
{
  "total": 7324,
  "filters": [filter_object, filter_object, filter_object],
}
```

### 1.3.3. GET /icc/v1/filters/{filter\_id}

Get the filter

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/filters/1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5201	resource: Not found oem: invalid

Response example:

[filter\\_object](#)

### 1.3.4. POST /icc/v1/filters

Create a filter

Ayla::Staff, Ayla::Admin can access

Query Param:

oem	Optional	For Ayla user only
-----	----------	--------------------

Input Body:

oem_model	Mandatory	
name	Mandatory	Maximum 100 characters

attributes	Optional	List of <a href="#">data objects</a>
dsns	Optional	<pre>{   "match": ["dsn1"],   "include": ["dsn2"],   "exclude": ["dsn3"] }</pre> <p><b>match:</b> the result of dsns must match to either of them in this list Example: match: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn1"]</p> <p><b>include:</b> the dsns in this list should also be added if they are not present in the result Example: include: ["dsn3"] result: ["dsn1", "dsn2"] return: ["dsn1", "dsn2", "dsn3"]</p> <p><b>exclude:</b> the result of dsns must not have DSNs given in this list Example: exclude: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn2"]</p>
device_metadata	Optional	List of <a href="#">data objects</a>
properties	Optional	List of <a href="#">data objects</a>
status	Optional	List of <a href="#">data objects</a>
description	Optional	
oem_version	Optional	The version defined in property (Host sw version)
filter_metadata	Optional	List of <a href="#">data objects</a>

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"oem_model": "linuxex1",
"name": "V2 Washers", "properties": [{"key": "Backlight", "op": "=", "value": "1"}]}'
'https://<hostname>/icc/v1/filters'
```

Response:

,

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	ICC-5301 ICC-5302 ICC-5401	name: cannot be empty name: must not exceed 100 characters request body is invalid or no given parameters

Response example:

[filter\\_object](#)

### 1.3.5. POST /icc/v1/filters/preview

Preview sample devices result of a filter

Ayla::Staff, Ayla::Admin can access

Query Param:

oem	Optional	For Ayla user only
limit	Optional	Number of entries to be returned. Default:

Input Body:

oem_model	Mandatory	
attributes	Optional	List of <a href="#">data objects</a>
dsns	Optional	<pre>{   "match": ["dsn1"],   "include": ["dsn2"],   "exclude": ["dsn3"] }</pre> <p><b>match:</b> the result of dsns must match to either of them in this list Example: match: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn1"]</p> <p><b>include:</b> the dsns in this list should also be added if they are not present in the result</p>



		<p>Example:  include: ["dsn3"]  result: ["dsn1", "dsn2"]  return: ["dsn1", "dsn2", "dsn3"]</p> <p><b>exclude:</b> the result of dsns must not have DSNs given in this list  Example:  exclude: ["dsn1"]  result: ["dsn1", "dsn2"]  return: ["dsn2"]</p>
device_metadata	Optional	List of <a href="#">data objects</a>
properties	Optional	List of <a href="#">data objects</a>
status	Optional	List of <a href="#">data objects</a>
oem_version	Optional	The version defined in property (Host sw version)

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -
d'{"oem_model": "linuxex1", "name": "V2 Washers", "properties": [{"key": "Backlight", "op": "=",
"value": "1"}]}' 'https://<hostname>/icc/v1/filters/preview'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "total": 7324,
  "devices": [device\_object, device\_object] //the length of list is based on the 'limit' in the
request
}
```

### 1.3.6. PUT /icc/v1/filters/{filter\_id}

Update the filter

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Input Body:

oem_model	Required	Not modifiable if filter is in use
name	Optional	Maximum 100 characters
attributes	Optional	List of <a href="#">data objects</a>
dsns	Optional	<pre>{   "match": ["dsn1"],   "include": ["dsn2"],   "exclude": ["dsn3"] }</pre> <p><b>match:</b> the result of dsns must match to either of them in this list Example: match: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn1"]</p> <p><b>include:</b> the dsns in this list should also be added if they are not present in the result Example: include: ["dsn3"] result: ["dsn1", "dsn2"] return: ["dsn1", "dsn2", "dsn3"]</p> <p><b>exclude:</b> the result of dsns must not have DSNs given in this list Example: exclude: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn2"]</p>
device_metadata	Optional	List of <a href="#">data objects</a>
properties	Optional	List of <a href="#">data objects</a>
status	Optional	List of <a href="#">data objects</a>
oem_version	Optional	The version defined in property (Host sw

		version)
description	Optional	
filter_metadata	Optional	List of <a href="#">data objects</a>

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"name": "V2 Washers",
"properties": [{"key": "Backlight", "op": "=", "value": "1"}]}' 'https://<hostname>/icc/v1/filters/1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-5301 ICC-5302 ICC-5401	name: cannot be empty name: must not exceed 100 characters request body is invalid or no given parameters

Response example:

[filter\\_object](#)

### 1.3.7. DELETE /icc/v1/filters/{filter\_id}

Delete the filter

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Example:

```
curl -v -X DELETE -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/
filters/1'
```

Response:

HTTP Status Code	code	message
204		

304(Not Modified)		No records affected or deleted
401(Unauthorized)		
403(Forbidden)		
422	ICC-2301	filter: in use

Response example:

N/A

### 1.3.8. POST /icc/v1/filters/{filter\_id}/copy

Copy the filter

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/
filters/1/copy'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
404(Not Found)		
422		

Response example:

[filter\\_object](#)

## 1.4. Filter Device Management

### 1.4.1. device\_object

```
{
  "dsn": "AC001",
  "lifecycle": "activated", // "provisioned", "registered"
  "connection_status": "Online", // "Offline"
  "connected_at": "2019-01-01 14:00:00" // YYYY-MM-dd HH:mm:ss
}
```

### 1.4.2. GET /icc/v1/filters/{filter\_id}/devices\_snapshot

Get the list of devices

OEM::Admin and Ayla users can access.

Query Param:

page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edbf78a527742d9380" 'https://<hostname>/icc/v1/filters/1/devices?
page=1&per_page=15'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5001 ICC-5101 ICC-5102	resource: Not found page: Invalid per_page: minimum 5 per_page: maximum 100

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
}
```

```
{
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end_count_on_page": 10,
  "total": 7324,
  "devices": [device_object, device_object]
}
```

### 1.4.3. POST /icc/v1/filters/{filter\_id}/devices/refresh

Search devices with the filter criteria and update the result  
OEM::Admin and Ayla users can access.

Query Param:  
None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/filters/1/refresh'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2001	devices: list is being updated

Response example:  
N/A

### 1.4.4. GET /icc/v1/filters/{filter\_id}/devices/export

download the list of devices  
OEM::Admin and Ayla users can access.

Query Param:  
None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/filters/1/devices/export'
```

,

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

CSV file

## 1.5. Job Management

### 1.5.1. job\_object

```
{
  "id":123,
  "name":"V2 Washers",
  "type_id":1,
  "filter":{}, //filter object
  "schedule":"IMMEDIATE", //"SCHEDULE"
  "status":"CREATED", //"STARTED", "SCHEDULED", "STOPPED",
"CANCELLED", "FAILED"
  "started_at": "2018-12-20T09:06:51+0000",
  "stopped_at": "2018-12-20T09:16:51+0000",
  "created_at":"2018-12-20T09:06:51+0000",
  "updated_at":"2018-12-20T09:16:51+0000",
  "devices_total":3000,
  "devices_processing":1000,
  "devices_succeed":1000,
  "devices_failed":1000,
  "description":"test",
  "job_metadata":[{"key":"m key", "value":"m value"}],
  "retries": 0
}
```

### 1.5.2. GET /icc/v1/jobs

Get the list of jobs

OEM::Admin and Ayla users can access.

Query Param:

oem	Optional	For Ayla user only
page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100
sort_by	Optional	The name of field that the returned list to be sorted by
direction	Optional	Sorting direction (ASC DESC)
job_type	Optional	filter the list by job type
oem_model	Optional	filter the list by oem model

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edbf78a527742d9380" 'https://<hostname>/icc/v1/jobs?page=1&per_page=15&sort_by=name&direction=DESC'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5001 ICC-5101 ICC-5102	resource: Not found page: Invalid per_page: minimum 5 per_page: maximum 100

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end_count_on_page": 10,
  "total": 7324,
  "jobs": [job\_object, job\_object, job\_object],
```



}

### 1.5.3. GET /icc/v1/jobs/{job\_id}

Get the job

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token  
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

[job\\_object](#)

### 1.5.4. POST /icc/v1/jobs

Create a job

Ayla::Staff, Ayla::Admin can access

Query Param:

oem	Optional	For Ayla user only
-----	----------	--------------------

Input Body:

name	Mandatory	
type_id	Mandatory	1: Set properties 2: Reset devices //not yet supported 3: Debug devices //not yet supported 4: OTA

		5: File transfer
filter_id	Mandatory	
exec_method	Mandatory	- ONE_TIME - CONTINUOUS
delivery_option	Mandatory	- SYSTEM_PUSH - USER_CONSENT
description	Optional	
job_metadata	Optional	List of <a href="#">data objects</a>
retries	Optional	Number of retry if one of the action of the device is failure, how many times should retry the action Default: 0 (no retry)
action_payload	Optional	An object that is used for job action, the payload depends on different type of action NOTE: the payload please refer to each action payload in this documentation

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edbf78a527742d9380" -d'{"name": "Washer light off", "type_id": "SET_PROPERTY", "filter_id": 1, "exec_method": "ONE_TIME"}' 'https://<hostname>/icc/v1/jobs'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2205 ICC-6001 ICC-6201 ICC-6901	specific filter not found job_type: not supported exec_method: invalid delivery_option: invalid

Response example:

[job\\_object](#)

### 1.5.5. PUT /icc/v1/jobs/{job\_id}

Update the job. C

cannot be updated once started

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Input Body:

name	Optional	
type_id	Mandatory	1: Set properties 2: Reset devices //not yet supported 3: Debug devices //not yet supported 4: OTA 5: File transfer
filter_id	Mandatory	
exec_method	Mandatory	- CONTINUOUS - ONE_TIME
description	Optional	
job_metadata	Optional	List of <a href="#">data objects</a>
retries	Optional	Number of retry if one of the action of the device is failure, how many times should retry the action Default: 0 (no retry)
action_payload	Optional	An object that is used for job action, the payload depends on different type of action NOTE: the payload please refer to each action payload in this documentation

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d' {"name": "Washer light off", "type_id":2, "filter_id":1}' 'https://<hostname>/icc/v1/jobs/1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		

403(Forbidden)		
422	ICC-2205 ICC-6001 ICC-6201 ICC-6901	specific filter not found job_type: not supported exec_method: invalid delivery_option: invalid

Response example:

[job\\_object](#)

### 1.5.6. DELETE /icc/v1/jobs/{job\_id}

Delete the job

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Example:

```
curl -v -X DELETE -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1'
```

Response:

HTTP Status Code	code	message
204		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2207	job: is running

Response example:

N/A

### 1.5.7 POST /icc/v1/jobs/{job\_id}/copy

Duplicate a same job for further modification.

OEM::Admin and Ayla users can access.

Query Param:

,

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token  
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/copy?re_use=1'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

[job\\_object](#)

## 1.5.8 Schedule Object

**Schedule Object:**

```
{  
  "name": "name of this schedule",  
  "timezone": "GMT-8:00",  
  "start_time": "YYYY-MM-DD HH:mm",  
  "end_time": " YYYY-MM-DD HH:mm",  
  "schedule_type": "daily",  
  "repeats": [  
    {  
      "start_time": {  
        "month_of_year": [10],  
        "day_of_month": [1],  
        "day_of_week": [7],  
        "time": "00:00"  
      },  
      "stop_time": {  
        "month_of_year": [10],  
        "day_of_month": [1],  
        "day_of_week": [7],  
        "time": "23:59"  
      }  
    }  
  ]  
}
```

## 1.5.9 POST /icc/v1/jobs/{job\_id}/schedule

Set a schedule for the job

OEM::Admin and Ayla users can access.

Query Param:

None

Input Body:

name	Mandatory	The name of the schedule
timezone	Mandatory	The timezone that the schedule will run at ex: GMT-8:00, GMT+0:00 etc
start_time	Mandatory	Set the start time of the schedule
end_time	Optional	Set the stop time of the schedule Set null or this field is not given, the schedule will never be stopped
schedule_type	Mandatory	Give it this schedule as DAILY, WEEKLY, MONTHLY or CUSTOM This field affects the validation of field "repeats": <b>DAILY</b> : only time can be set <b>WEEKLY</b> : only time and day_of_week can be set <b>MONTHLY</b> : only time and day_of_month can be set <b>CUSTOM</b> : no restriction
repeats	Optional	Set the schedule that will run the job periodically at particular time between <b>"start_time"</b> and <b>"end_time"</b> , the value should be an array of the objects which have "from" and "to" to specify when the job should be started and stopped  <b><i>day_of_month and day_of_week cannot be set at the same time, only choose either of them to specify the day that the job will be started/stopped</i></b>  month_of_year: 1 – 12 (Jan to Dec) day_of_month: 1 – 31 (depends on each month) day_of_week: 1 – 7 (Mon to Sun)  ex: run the job every day from 2pm to 5pm [{

		<pre> "start_time": {   "time": "14:00" }, "stop_time": {   "time": "17:00" } } ]]  ex: run the job from 2pm to 5pm every Tuesday and Thursday [[   "start_time": {     "day_of_week": [2, 4],     "time": "14:00"   },   "stop_time": {     "day_of_week": [2, 4],     "time": "17:00"   } } ]]  ex: run the job at 8am and stop it at 10am every 10<sup>th</sup> of the month from July and September [[   "start_time": {     "month_of_year": [7, 9],     "day_of_month": [10],     "time": "8:00"   },   "stop_time": {     "month_of_year": [7, 9],     "day_of_month": [10],     "time": "10:00"   } } ]] </pre>
--	--	---

Example:

```

curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" -d"schedule
object""https://<hostname>/icc/v1/jobs/1/schedule"

```

Response:

HTTP Status Code	code	message
201		

401(Unauthorized)		
403(Forbidden)		
422	ICC-6301 ICC-6601 ICC-6701 ICC-6702 ICC-6703 ICC-6704 ICC-6705	timezone: cannot be empty time: cannot be empty only can choose either day of month or day of week empty or unknown schedule type invalid schedule type with the given repeat setting contains duplicate time the start time will never be triggered

Response example:

[schedule\\_object](#)

### 1.5.10 GET /icc/v1/jobs/{job\_id}/schedule

Get the schedule of the job

OEM::Admin and Ayla users can access.

Query Param:

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/copy?re_use=1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

[schedule\\_object](#)

### 1.5.11 DELETE /icc/v1/jobs/{job\_id}/schedule

Remove the schedule from job

OEM::Admin and Ayla users can access.

Query Param:



Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token  
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/copy?re_use=1'
```

Response:

HTTP Status Code	code	message
204		
401(Unauthorized)		
403(Forbidden)		
422		

Response Example:

None

### 1.5.12. POST /icc/v1/jobs/express

Create a job

Ayla::Staff, Ayla::Admin can access

Query Param:

oem	Optional	For Ayla user only
-----	----------	--------------------

Input Body:

type_id	Mandatory	SET_PROPERTY(1): Set properties OTA(4): OTA FILE_TRANSFER(5): File transfer  Can be either "name" or its "id" For example: Can pass either "SET_PROPERTY" or "1"
oem_model	Mandatory	
dsns	Mandatory	An array of dsns to be deployed
action_payload	Mandatory	An object that is used for job action, the payload depends on different type of action NOTE: the payload please refer to each action payload in this documentation

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"oem_model": "ledevb", "type_id": "SET_PROPERTY", "dsns": [ "dsn1", "dsn2" ], "action_payload": {...}}' 'https://<hostname>/icc/v1/jobs'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

[job\\_object](#)

## 1.6. Job Device Management

### 1.6.1. GET /icc/v1/jobs/{job\_id}/devices\_snapshot

Get the list of devices

OEM::Admin and Ayla users can access.

Query Param:

page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/devices?page=1&per_page=15'
```

Response:

HTTP Status Code	code	message
200		

401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5001 ICC-5101 ICC-5102	resource: Not found page: Invalid per_page: minimum 5 per_page: maximum 100

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end_count_on_page": 10,
  "total": 7324,
  "devices": [device\_object, device\_object]
}
```

### 1.6.2. GET /icc/v1/jobs/{job\_id}/devices/{dsn}/details

Get the status details of a device

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380"
'https://<hostname>/icc/v1/jobs/1/devices/A0001/details?page=1&per_page=15'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2202	job has not been started yet

Response example:

,

```

{
  "task_id": 1,
  "job_type": "SET_PROPERTY", //OTA, FILE_TRANSFER
  "actions": [
    {
      "action_content": "property_name1", // or an action identity (e.g. ota id)
      "status": "SUCCEED", // or SENT, FAILED
      "created_at": "2018-12-20T09:06:51+0000"
    },
    {
      "action_content": "property_name2", // or an action identity (e.g. ota id)
      "status": "SUCCEED", // or SENT, FAILED
      "created_at": "2018-12-20T09:06:51+0000"
    }
  ],
}

```

## 1.7. Job Operation Management

### 1.7.1. POST /icc/v1/jobs/{job\_id}/start

Start the job

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/start'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2201 ICC-2502 ICC-2211	job: failed to start job: job has started job: unable to start job, the schedule is invalid

Response example:

N/A

### 1.7.2. POST /icc/v1/jobs/{job\_id}/cancel

Cancel the job

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/cancel'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

N/A

### 1.7.3. (internal) POST /icc/internal/jobs/{job\_id}/execute

Trigger task execution

For scheduler to call

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/internal/jobs/1/execute'
```

Response:

HTTP Status Code	code	message
------------------	------	---------

201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

N/A

### 1.7.4. PUT /icc/v1/jobs/{job\_id}/resume

ONLY for resuming the job paused by stages

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/internal/jobs/1/resume'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

N/A

## 1.8. Action: Set Properties

### 1.8.1. GET /icc/v1/jobs/{job\_id}/properties

Get the properties to set for the job

OEM::Admin and Ayla users can access.

,

Query Param:  
None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/1/properties'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{  
  "properties": [data object, data object, data object]  
}
```

### 1.8.2. POST /icc/v1/jobs/{job\_id}/properties

Set properties for the job

Ayla::Staff, Ayla::Admin can access

Query Param:  
None

Input Body:

properties	Optional	List of <a href="#">data objects</a>
------------	----------	--------------------------------------

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"properties": [{"key": "Backlight", "value": "1"}]}' 'https://<hostname>/icc/v1/jobs/1/properties'
```

Response:

HTTP Status Code	code	message
201		

401(Unauthorized)		
403(Forbidden)		
422	ICC-2101 ICC-2207	properties: not a SET_PROPERTY job job: is running

Response example:

```
{
  "properties": [data object, data object, data object]
}
```

## 1.9. Action: File Management

### 1.9.1. file\_object & file\_property

file\_post (for POST & PUT)

```
{
  "name": "Latte Recipe",
  "description": "A new flavor of latte",
  "url": "http://123.com/abc123",
  "file_metadata": [{"key": "m key", "value": "m value"}],
}
```

file\_object:

```
{
  "id": 123,
  "name": "Latte Recipe",
  "description": "A new flavor of latte",
  "url": "http://123.com/abc123",
  "created_at": "2018-12-20T09:06:51+0000",
  "updated_at": "2018-12-20T09:06:51+0000",
  "file_metadata": [{"key": "m key", "value": "m value"}],
  "file_size": 100000
}
```

file\_property:

```
{
  "file_id": 1234,
  "name": "recipe1"
}
```

,



### 1.9.2. POST /icc/v1/files

Create a file transfer record

OEM::Admin and Ayla users can access.

Query Param:

oem	Optional	For Ayla user only
-----	----------	--------------------

Input Body (application/json):

name	Mandatory	
description	Optional	
file_metadata	Optional	
url	Optional	A given external file url for file property, or upload the binary file later

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"name": "Latte recipe", "description": "A new flavor of latte"}' 'https://<hostname>/icc/v1/files'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2401	file: failed to save

Response example:

File\_object

### 1.9.3. POST /icc/v1/files/{file\_id}/upload

Upload a file to the file record

OEM::Admin and Ayla users can access.

,

Query Param:  
None

Input Body (multipart/form-data):

file	Mandatory	The binary file to be uploaded
------	-----------	--------------------------------

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -F"file=@/home/user1/Desktop/file1.bin" 'https://<hostname>/icc/v1/files/upload'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2401	file: failed to save

Response example:  
File\_object

#### 1.9.4. GET /icc/v1/files

Get all the file transfer records

Query Param:

oem	Optional	For Ayla user only
page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100
sort_by	Optional	The name of field that the returned list to be sorted by
direction	Optional	Sorting direction (ASC DESC)

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/files?page=1&per_page=15'
```

,

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end_count_on_page": 10,
  "total": 7324,
  "files": [file_object, file_object, file_object]
}
```

### 1.9.5. PUT /icc/v1/files/{file\_id}

Update a file transfer record

OEM::Admin and Ayla users can access.

Query Param:

None

Input Body (application/json):

name	Mandatory	
description	Optional	
file_metadata	Optional	
url	Optional	A given external file url for file property

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"name": "Latte recipe", "description": "A new flavor of latte"}' 'https://<hostname>/icc/v1/files'
```

,

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

File\_object

### 1.9.6. DELETE/icc/v1/files/{file\_id}

Delete a file transfer record

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X DELETE -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/files/1'
```

Response:

HTTP Status Code	code	message
204		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

N/A

### 1.9.7. GET /icc/v1/files/{file\_id}

Get a file transfer record

,

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/files/1'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-0301 ICC-5001 ICC-5101 ICC-5102 ICC-5201	resource: Not found page: Invalid per_page: minimum 5 per_page: maximum 100 Oem: invalid

Response example:

File\_object

### 1.9.8. GET /icc/v1/files/{file\_id}/download

Download the file to a file transfer record

Query Param:

None

Input Body:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/files/1/download'
```

Response:

HTTP Status Code	code	message
201		

401(Unauthorized)		
403(Forbidden)		
422	ICC-2504	unable to read file

Response example:

HTTP/1.1 200 OK

Content-Type: application/octet-stream

### 1.9.9. POST /icc/v1/jobs/{job\_id}/file\_property

Set file\_property for the job

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Input Body:

properties	Mandatory	List of <a href="#">file_property</a>
------------	-----------	---------------------------------------

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"properties": [{"file_id":
1234, "name": "recipe1"}]}' 'https://<hostname>/icc/v1/jobs/{job_id}/file_property'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "properties": [file\_property, file\_property, file\_property]
}
```

### 1.9.10. GET /icc/v1/jobs/{job\_id}/file\_property

Get file\_property for the job

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Input Body:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/{job_id}/file_property'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "properties": [file_property, file_property, file_property]
}
```

## 1.10. OTA Action

### 1.10.1 OTA Object

```
{
  "type": "module" or "host",
  "ayla_model": "linuxex1",
  "version": "appd_demo 1.1.1-beta 2016-11-25 04:05:47 /",
  "command": "label:123",
  "description": "test"
}
```

Ota Object (List)

```
{
  "filter_id": 123, //given if the ota has a job created through ICC
  API
  "job_id": 123, //it's given if the ota is created through ICC API
  "base_mod_img_model": "linuxex1",
  "base_mod_img_version": "appd_demo 1.1.1-beta 2016-11-25
04:05:47 /",
  "command": "label:123",
  "created_at": "2018-01-01 00:00:00",
  "description": "test",
  "errors": 3,
  "file_size": 11222000,
  "from_host_img_version": "appd_demo 1.1.1-beta 2016-11-25
04:05:47 /",
  "from_mod_img_version": "appd_demo 1.1.1-beta 2016-11-25
04:05:47 /",
  "homekit": true,
  "id": 123,
  "intelligent": true,
  "lan_ota": true,
  "model": "linuxex1",
  "oem": "0dfc7900",
  "oem_model": "oem123",
  "pending": 4,
  "started_on": "2018-01-01 00:00:00",
  "status": "initialized", "started", "finished", "canceled",
  "stopped_on": "2018-01-01 00:00:00",
  "type": "module", "host",
  "updated_at": "2018-01-01 00:00:00",
  "version": "appd_demo 1.1.1-beta 2016-11-25 04:05:47 /",
  "is_online": true,
  "is_registered": true,
}
```

### 1.10.2 POST /icc/v1/jobs/{job\_id}/ota

Set an OTA action to the job

Ayla::Staff, Ayla::Admin can access

Query Param:

None

Input Body:

,



type	Mandatory	host   module
ayla_model	Mandatory if type=module	Ayla Module
version	Mandatory	Image version
description	Optional	
command	Optional	Additional string type info passing into device Length: 32

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"type": "module", "ayla_model": "linuxex1", "version": "appd_demo 1.1.1-beta 2016-11-25 04:05:47 /", "description": "OTA from 1.1 to 1.2", "command": "label part1"}' https://<hostname>/icc/v1/jobs/{job_id}/ota'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	AOS-200030 AOS-200130  AOS-210140 AOS-200090	Image: From version is same to upgraded version Image: Image is not existed. You need to upload the image first. OTA: Job is duplicate. Permission: Image owner is not belonging to login user.

Response example:

[ota\\_object](#)

### 1.10.3 GET /icc/v1/jobs/{job\_id}/ota

Get an OTA action from the Job

Ayla::Staff, Ayla::Admin can access

Query Param:

None

,

Input Body:  
None

```
curl -v -X GET -H"Content-type: application/json" -H"Content-type: application/json" -  
H"Authorization: auth_token 5a119ef6285c4edbf78a527742d9380"  
'https://<hostname>/icc/v1/jobs/{job_id}/ota'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:  
[ota\\_object](#)

#### 1.10.4. GET /icc/v1/otas

Get the list of the OTAs  
OEM::Admin and Ayla users can access.

Query Param:

oem	Optional	For Ayla user only. Required if type=host
type	Mandatory	module   host
model	Optional	Ayla model
oem_model	Optional	Oem model
page	Optional	page number. Default 1
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 100
order_by	Optional	Module OTA -> status   model   version   started_on   stopped_on  Host OTA -> status   from_host_img_version   version started_on   stopped_on

		Default: order by ID
order	Optional	asc   desc Default: desc

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Authorization: auth_token
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/otas?type=host'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	AOS-100515 AOS-100050 AOS-100160 AOS-100616	oem: can't be blank order_by: Invalid order: Invalid type: Missing

Response example:

```
{
  "previous_page": null,
  "next_page": 2,
  "current_page_number": 1,
  "start_count_on_page": 1,
  "end_count_on_page": 10,
  "total": 7324,
  "otas": [ota\_object \(GET\), ota\_object \(GET\), ota\_object \(GET\)],
}
```

#### 1.10.5. POST /icc/v1/oems/{oem}/otas/{ota\_id}/start

Start OTA

OEM::Admin and Ayla users can access.

Query Param:

None

,

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token  
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/otas/1234/start'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	AOS-2000103 AOS-1000100 AOS-1001100 AOS-1000101 AOS-1001101	ota: Image owner is not belonging to login user. Submit a Spark job failed. Job has been started. Submit an intelligent OTA failed. No match rule for intelligent OTA.

Response example:

N/A

#### 1.10.6. POST /icc/v1/oems/{oem}/otas/{ota\_id}/stop

Stop OTA

OEM::Admin and Ayla users can access.

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Authorization: auth_token  
5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/otas/1234/start'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	AOS-2000103 AOS-2000100	ota: Image owner is not belonging to login user. Delete a Spark job failed.

	AOS-2000101	Delete an intelligent OTA failed.
--	-------------	-----------------------------------

Response example:

N/A

## 1.11. User Consent

### 1.11.1 Device Job Object

Device Object for GET (get\_device\_object):

```
{
  "dsn": "A0001",
  "device_jobs": [
    {
      "id": "qQ0IbCd7",
      "name": "V2 Washers",
      "job_type": "SET_PROPERTY", //FILE_TRANSFER, OTA
      "delivery_option": "SYSTEM_PUSH", //USER_CONSENT
      "started_at": "2018-12-20T09:06:51+0000",
      "stopped_at": "2018-12-20T09:16:51+0000",
      "description": "test",
      "job_metadata": [{"key": "m key", "value": "m value"}],
      "device_status": "PENDING"
    }
  ]
}
```

Device Object for PUT (put\_device\_object):

```
{
  "dsn": "A0001",
  "device_jobs": [
    { //schedule the action
      "id": "qQ0IbCd7",
      "schedule_timestamp": 1565078393 //Optional
    },
    { //do action now
      "id": "gdd033g4"
    },
    { //Cancel or Ignore the action
      "id": "badI3Cg2",
      "cancel_job": true
    }
  ]
}
```

,

#### Delivery Option

Option	Description
SYSTEM_PUSH	The action of the job is forced to push to the device
USER_CONSENT	The action of the job needs to wait for user's consent

#### Device Status

Status	Description
PENDING	The device is being queued on backend, waiting to be processed
CONSENT	The device is waiting for user to decide when it goes on <b>(Only this status of the device can be sent an "PUT" to the cloud)</b>
SCHEDULED	The device is being scheduled to take the action
PROCESSING	The device is being processed
SUCCEED	The device is done with the assigned action by the job, and it succeeded
FAILED	The action assigned to the device has been failed
PARTIALY_FAILED	There are some of assigned action that are failed, but some of them succeeded
CANCELLED	The action has been cancelled by user or OEM user/admin

#### 1.11.2 GET /icc/v1/jobs/{job\_id}/consents

Get all the running consent jobs with the same type from the target job

**OEM::Admin and Ayla users can access.**

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Content-type: application/json" -  
H"Authorization: auth_token 5a119ef6285c4edbf78a527742d9380"  
'https://<hostname>/icc/v1/jobs/{job_id}/consents'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
```

```
,
```

```
    "jobs": [job_object, job_object]
}
```

### 1.11.3 Mobile Usage API

#### 1.11.3.1 GET /icc/v1/users/device\_jobs

Get all the jobs for all the devices belong to the user  
ENDUSER

Query Param:

from	Optional	Unix timestamp (in second)
to	Optional	Unix timestamp (in second)
status	Optional	Device status
oem_model	Optional	
dsns	Optional	DSNs that are separated by comma ex: AC001,ACC002
page	Optional	page number. Default 1. Page cannot be smaller than 1.
per_page	Optional	Number of entries returned. Default 10. Minimum 5. Maximum: 200
order_by	Optional	The name of field that the returned list to be sorted by
order	Optional	Sorting direction (ASC DESC)

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -  
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/  
users/device_jobs'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422	ICC-6902	dsn: invalid or not exists

	ICC-6903	dsn: maximum 200 devices
--	----------	--------------------------

Response example:

```
{
  "devices": [get_device_object, get_device_object, ...]
}
```

### 1.11.3.2 PUT /icc/v1/dsns/{dsn}/device\_jobs/{device\_job\_id}

Put the decision for each devices and device jobs

ENDUSER

Input Body (application/json):

schedule_timestamp	Optional	Unix timestamp (in seconds) (given if the device is scheulded by this timestamp) 0: run the job immediately
cancel_job	Mandatory	true: if the device doesn't need to be processed. false (or not being given) confirm the device will be processed by the job

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d'{"schedule_timestamp": 1565078393}' 'https://<hostname>/icc/v1/dsns/A001/jdevice_jobs/qQ0lbCd7'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2801 ICC-2209	device_job: the given DSN is unavailable job: invalid schedule time

Response example:

**get\_device\_object**

### 1.11.3.3 PUT /icc/v1/users/device\_jobs

Put the batch of devices for each devices and device jobs

ENDUSER

,



Input Body:

devices	Mandatory	List of “Device Job Object for PUT”
---------	-----------	-------------------------------------

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -d '{"devices": [{"dsn": "A0001", "device_jobs": [{"id": "qQ0lbCd7", "schedule_timestamp": 1565078393}, {"id": "gddO33g4"}, {"id": "badl3Cg2", "cancel_job": true}]}]}' https://<hostname>/icc/v1/users/device_jobs'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422	ICC-2801 ICC-2209	device_job: the given DSN is unavailable job: invalid schedule time

Response example:

```
{
    "succeeded": [put_device_object, put_device_object, ...],
    "failed": [put_device_object, put_device_object, ...]
}
```

## 1.12 Stage Run

### 1.12.1 Stage Object

POST

```
{
    "total": 1000,
    "failure_threshold": 300, //The last should be 0
    "timeout": 600 //in seconds
}
```

GET

```
{
  "stage_number": 1, //1 - 5
  "total": 1000,
  "failure_threshold": 300,
  "devices_executed": 650,
```

1

```

    "devices_succeeded":200,
    "devices_failed":50,
    "status": "initialized", "processing", "succeed", "failed"
}

```

All Remaining Stage (returned if job started)

```

{
    "stage_number": 4, //2 - 5
    "total": 0, //the "All Remaining" Stage doesn't have total
    "failure_threshold": -1,
    "devices_executed": 650, //number of remaining devices
    "devices_succeeded": 0,
    "devices_failed": 0,
    "status": "FINISHED"
}

```

**PS.** All the status detail of remaining devices should be retrieved by calling API "1.6 Job Device Management"

### 1.12.2 POST /icc/v1/jobs/{job\_id}/stages

Add stages to the job

**OEM::Admin and Ayla users can access.**

Query Param:

None

Input Body:

stages	Mandatory	all the stages
--------	-----------	----------------

Example:

```

curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" -D'{"stages": [{"stage_type
":"number","devices_to_target":1000,"failure_threshold":300}]}' 'https://<hostname> /icc/v1/jobs/
4325/stages'

```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		

403(Forbidden)		
422		

Response example:

```
{
  "stages": [stage\_object, stage\_object, stage\_object]
}
```

### 1.12.3 GET /icc/v1/jobs/{job\_id}/stages

Get all stage of the job

**OEM::Admin and Ayla users can access.**

Query Param:

None

Example:

```
curl -v -X GET -H"Content-type: application/json" -H"Content-type: application/json" -
H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname> /icc/v1/
jobs/4325/stages'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "stages": [stage\_object, stage\_object, stage\_object]
}
```

### 1.12.4 DELETE /icc/v1/jobs/{job\_id}/stages/{stage\_number}

Delete an existing stage of the job

“all remaining” stage cannot be deleted.

**OEM::Admin and Ayla users can access.**

,

Query Param:

None

Example:

```
curl -v -X DELETE -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/{job_id}/stages/4'
```

Response:

HTTP Status Code	code	message
204		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

None

### 1.12.5 DELETE /icc/v1/jobs/{job\_id}/stages

Delete stages of the job, the job won't be run in stages

**OEM::Admin and Ayla users can access.**

Query Param:

None

Example:

```
curl -v -X DELETE -H"Content-type: application/json" -H"Content-type: application/json" -H"Authorization: auth_token 5a119ef6285c4edfbf78a527742d9380" 'https://<hostname>/icc/v1/jobs/{job_id}/stages'
```

Response:

HTTP Status Code	code	message
204		
401(Unauthorized)		
403(Forbidden)		

422		
-----	--	--

Response example:  
None

## 2.1. searchservice

### 2.1.1. (internal) POST /iccsearch/internalv1/advanced\_search

Search devices

Query Param:

None

Input Body:

oem	Mandatory	
oem_model	Mandatory	
oem_version	Optional	It's equal to the host sw version, service will look up the exact property name and match its value to this field
match_oem_version	Optional	Default: true <ul style="list-style-type: none"><li>true: OEM version should be equal to input. Device Search service should pass operator '=' to DPS</li><li>false: Device Search service should pass operator '!=' to DPS</li></ul>
limit	Optional	Number of devices in response -1 – return total and all dsns 0 – only return total and no dsns Default – return total and 20 dsns
attributes	Optional	List of <a href="#">data objects</a>
device_metadata	Optional	List of <a href="#">data objects</a>
properties	Optional	List of <a href="#">data objects</a>
status	Optional	List of <a href="#">data objects</a>
dsns	Optional	<pre>{   "match": ["dsn1"],   "include": ["dsn2"],   "exclude": ["dsn3"] }</pre> <p><b>match:</b> the result of dsns must match to either of them in this list Example: match: ["dsn1"]</p>

		<p>result: ["dsn1", "dsn2"] return: ["dsn1"]</p> <p><b>include:</b> the dsns in this list should also be added if they are not present in the result Example: include: ["dsn3"] result: ["dsn1", "dsn2"] return: ["dsn1", "dsn2", "dsn3"]</p> <p><b>exclude:</b> the result of dsns must not have DSNs given in this list Example: exclude: ["dsn1"] result: ["dsn1", "dsn2"] return: ["dsn2"]</p>
--	--	--

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=iccservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=afe388244ed5df9abf456277b787d2d946dd8ee26212f5687f6a7de70ec1a161" -H "x-
sso-date: 20181226T090236Z" -H "x-ayla-origin-host: deviceservice.dev" -d'{"oem": "afaf1100",
"oem_mode":"linuxex1":"properties": [{"key":"Backlight", "op":"=", "value":"1", "metadata":
[{"key":"metadata1", "value":1}], {"key":"Backlight", "op":"=", "value":"1"}], "dsns":{"match":
["dsn1"], "include": ["dsn2"], "exclude": ["dsn3"]} }'
'https://<hostname>/iccservice/internalv1/advanced_search'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
422		

Response example:

```
{
  "devices": [device\_object, device\_object],
  "total": 2334
}
```

## 2.2. datapointservice

### 2.2.1. (internal) POST /dp-svc/internalv1/advanced\_search

Search devices

Query Param:

None

Input Body:

oem	Mandatory	
oem_model	Mandatory	
dsns	Mandatory	Array of dsns to be filtered. Example: ["AC001", "AC002"]
properties	Mandatory	List of <a href="#">data objects</a>

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=iccservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=afe388244ed5df9abf456277b787d2d946dd8ee26212f5687f6a7de70ec1a161" -H "x-
sso-date: 20181226T090236Z" -H "x-ayla-origin-host: deviceservice.dev" -d'{"oem": "afaf1100",
"oem_model": "foobar0001", "dsns": ["AC000W0000002175", "AC000W0000004132",
"AC000W0000007190", "AC000W0000008117"], "properties": [{"key": "Backlight1", "op": "=",
"value": "1"}, {"key": "Backlight2", "op": ">", "value": "1"}]}'
'https://<dps_hostname>/dp-svc/api/internalv1/advanced_search'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
422		Missing arguments, or invalid argument values

Response example:

```
{
  "dsns":["AC000W0000004132","AC000W0000007190","AC000W0000008117"]
}
```

,



## 2.3. deviceservice

### 2.3.1. (internal) POST

/deviceservice/internal/v1/oems/{oem}/properties/{property\_name}/datapoints.json

Set datapoint

Query Param:

None

Input Body:

dsns	Mandatory	
internal_info	Optional	Format: { "icc": { "task_id": 21 "file_property_url": " <a href="http://example.com/test.txt">http://example.com/test.txt</a> ", "file_property_url": "s3://obj_key" } }
datapoint:	Mandatory	
value	Mandatory	

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=iccservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=afe388244ed5df9abf456277b787d2d946dd8ee26212f5687f6a7de70ec1a161" -H "x-
sso-date: 20181226T090236Z" -H "x-ayla-origin-host: deviceservice.dev" -d'{"datapoint":
{"value":1}}' 'https://<deviceservice>/deviceservice/internal/v1/oems/0dfc7900/properties/
oem_host_version/datapoint.json'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		

404(Not Found)		
422		

Response example:

```
{
  "errors":[{"AC001":"error1"}, {"AC002":"error2"}]
}
```

## 2.4 Schedule service

Schedule object

```
{
  "job_id": 2,
  "schedules": [{
    "days_of_week": "1,2,3,4,5,6,7",
    "hours": 11,
    "minutes": 22
  },
  {
    "days_of_week": "1,2,4,5,6",
    "hours": 22,
    "minutes": 10
  }
]
}
```

### 2.4.1 (internal) POST /iccschedule/v1/schedules

Create schedules for ICC job.

Input Body:

job	Mandatory	The id of ICC job
schedules	Mandatory	List of schedule object

Example:

```
curl -X POST \
  http://localhost:8089/icc/v1/schedules/ \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 66db0bf8-f00d-4645-88ba-4c89987976cf' \
  -H 'cache-control: no-cache' \
```

```
-d '{
  "job_id": 2,
  "schedules": [
    {
      "days_of_week": "1,2,3,4,5,6,7",
      "hours": 11,
      "minutes": 22
    },
    {
      "days_of_week": "1,2,4,5,6",
      "hours": 22,
      "minutes": 10
    }
  ]
}'
```

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		
409 (Conflict)		
422		

## 2.4.2 (internal) DELETE /iccschedule/v1/schedules/{job\_id}

Delete a schedule by ICC job ID.

Input Body:

N/A

Response:

HTTP Status Code	code	message
201		
401(Unauthorized)		
403(Forbidden)		

204 (No_Content)		
422		

## 2.5. otaservice

### 2.5.1. (internal) POST

/otaservice/v1/internal/oems/{:oem\_str}/otas/{:ota\_id}/start

Start an OTA

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=otaservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=42513eefb29d270c3414c793c70c2bab85083bf5f1334320d7b2451bd8953c02" -H "x-
sso-date: 20180816T060721Z" -H "x-ayla-origin-host: deviceservice.dev"
'https://<hostname>/otaservice/v1/internal/oems/0dfc7900/otas/123/start'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
404(Not Found)		
422		

Response example:

N/A

### 2.5.2. (internal) POST

/otaservice/v1/internal/oems/{:oem\_str}/otas/{:ota\_id}/stop

Stop an OTA

Query Param:

None

Example:

```
curl -v -X POST -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=otaservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=42513eefb29d270c3414c793c70c2bab85083bf5f1334320d7b2451bd8953c02" -H "x-
sso-date: 20180816T060721Z" -H "x-ayla-origin-host: deviceservice.dev"
'https://<hostname>/otaservice/v1/internal/oems/0dfc7900/otas/123/stop'
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
404(Not Found)		
422		

Response example:

N/A

### 2.5.3. (internal) POST /otaservice/v1/internal/otas/{:ota\_id}/ota\_devices

Internal Service API. Add devices for an OTA

Input Body:

oem	Mandatory	OEM of the OTA job
dsns	Mandatory	DSNs to be added

Json example:

```
{
  "dsns": [ "dsn1", "dsn2" ]
}
```

}

Example:

```
curl -v -X PUT -H"Content-type: application/json" -H"Content-type: application/json" -H
"Authorization: HMAC-SHA256 Credential=otaservice/s2s/v1, SignedHeaders=x-ayla-origin-
host;x-sso-date,
Signature=42513eefb29d270c3414c793c70c2bab85083bf5f1334320d7b2451bd8953c02" -H "x-
sso-date: 20180816T060721Z" -H "x-ayla-origin-host: deviceservice.dev" -d'{"oem": "0dfc7900",
"dsns": [ "dsns1", "dsn2"]}' https://<hostname>/otaservice/internal/v1/otas/123/ota\_devices
```

Response:

HTTP Status Code	code	message
200		
401(Unauthorized)		
403(Forbidden)		
404(Not Found)		
422		

Response example:

N/A

## Appendix A Error Code Table

Code	Message
ICC-0001	Server Internal Error
ICC-0101	Wrong property type
ICC-0102	type: Cannot be null
ICC-0103	type: Cannot be blank
ICC-0201	path: Not Found
ICC-0301	resource: Not found
ICC-0401	sort: unsupported sorting field
ICC-0501	data: unable to read data
ICC-1001	Job/Filter/File owner is not belonging to login user.
ICC-2001	devices: list is being updated
ICC-2101	properties: not a SET_PROPERTY job
ICC-2102	properties: job has started
ICC-2201	job: failed to start
ICC-2202	job: job has not been started yet
ICC-2203	job: job has stopped

ICC-2204	job: unable to stop job
ICC-2205	job: specific filter not found
ICC-2206	job:specific job not found
ICC-2207	job: is running
ICC-2208	job: scheduler not found
ICC-2209	job: invalid schedule time
ICC-2210	job: no schedule found
ICC-2211	job: unable to start job, the schedule is invalid
ICC-2212	job: had been started, unable to update job
ICC-2301	filter: in use
ICC-2302	filter: has been used by started jobs, unable to modify oem model
ICC-2401	file: failed to save
ICC-2402	file: not found
ICC-2403	file: is in use
ICC-2501	file_property: not a FILE_TRANSFER job
ICC-2502	file_property: job has started
ICC-2503	file_property: not contain a file or uri
ICC-2504	file_property: unable to read file
ICC-2505	file_property: specific file not found
ICC-2601	upload: exceeds maximum allowed file size
ICC-2701	ota: unable to create OTA
ICC-5001	page: invalid
ICC-5101	per_page: minimum 5
ICC-5102	per_page: maximum 100
ICC-5201	oem: invalid
ICC-5301	name: cannot be empty
ICC-5302	name: must not exceed 100 characters
ICC-5401	filter: invalid filter criteria (maximum 10 properties or no duplicate dsns)
ICC-5402	filter: given filter_id is invalid
ICC-5403	filter: description exceed maximum length
ICC-5501	limit: maximum 100
ICC-5601	oem_model: cannot be empty
ICC-5602	oem_model: maximum 255
ICC-5701	ota_type: cannot be empty
ICC-5801	ayla_model: cannot be empty if type is 'module'
ICC-5901	ota_version: cannot be empty
ICC-6001	job_type: not supported
ICC-6101	properties: cannot be empty
ICC-6201	exec_method: invalid
ICC-6301	timezone: cannot be empty
ICC-6401	start_time: cannot be empty
ICC-6501	stop_time: cannot be empty
ICC-6601	time: cannot be empty
ICC-6701	schedule: only can choose either day of month or day of week
ICC-6702	schedule: empty or unknown schedule type
ICC-6703	schedule: invalid schedule type with the given repeat setting
ICC-6704	schedule: contains duplicate time
ICC-6705	schedule: the start time will never be triggered
ICC-6801	job_retry: maximum 10

ICC-6901	delivery_option: invalid
----------	--------------------------