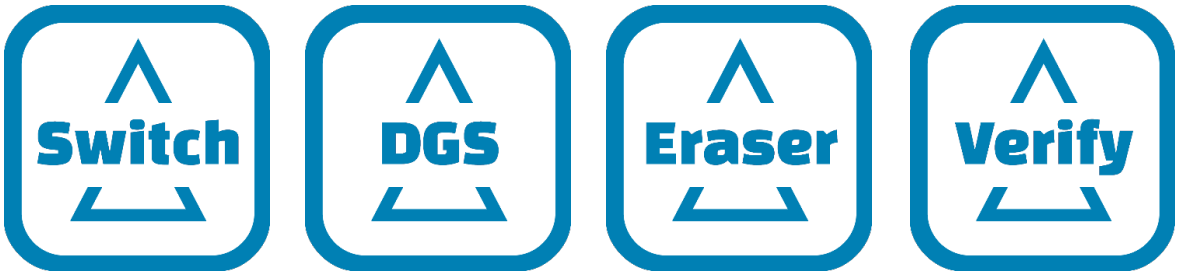

PiceaServices Reporting API

June 2019

v.4.5.6

Contents

1 Overview 2
1.1 Security 3
2 Report API 5
2.1 Fair API Usage 7
2.2 get_transactions / get_transactions_ex 8
2.2.1 Request 8
2.2.2 Response 9
2.3 get_transactions_for_imeis / get_transactions_for_imeis_ex 11
2.3.1 Request 11
2.3.2 Response 11
2.4 get_transactions_for_serials 13
2.4.1 Request 13
2.4.2 Response 13
2.5 get_transaction_details 15
2.5.1 Request 15
2.5.2 Response 15
2.6 get_transaction_details_ex 25
2.6.1 Request 25
2.6.2 Response 25
2.7 get_device_details 26
2.7.1 Request 26
2.7.2 Response 26
2.8 get_devices 28
2.8.1 Request 28
2.8.2 Response 28
2.9 get_pdf_report 30
2.9.1 Request 30
2.9.2 Response 30
2.10 get_file 31
2.10.1 Request 31
2.10.2 Response 31
3 Definitions 32

3.1 Device formats 32
3.2 Operation types 32
3.3 Content types 32
3.4 Transaction status values 33
3.5 Verify result values 33
3.6 Diagnostics test cases 34
3.7 Result values 39
3.7.1 Success codes 39
3.7.2 Failure codes 40
4 Sample Code 41

1 Overview

PiceaServices Reporting API provides a REST API to access Piceasoft reporting data. It can be used to query information about device transaction history and device details.

This API provides access only to the data generated by products licensed by the customer.

1.1 Security

To secure access and to validate caller identity, user needs to have a Piceasoft provided Online account to access services. When account is created, user will get own client ID and a private key. User is responsible to keep these values as a private information and to use those only as agreed with Piceasoft. Piceasoft is not responsible of any cost caused by lost or incorrectly used authentication data.

Client ID needs to be added to all requests as a HTTP header:

```
X-Piceasoft-Client-Id: <client id>
```

In addition to client id value, user needs to add a valid signature to all requests. This is generated using the private key provided by Piceasoft together with the client id. This needs to be added to all requests as a HTTP header:

```
X-Piceasoft-Signature: <signature>
```

Signature is a dynamically generated sha-256 value. It needs to be generated with the following pseudo algorithm.

```
sha256(<private key>+<client id>+JSONRequestData)
```

Note that the produced hexstring characters must be lowercase characters.

Curl/bash example:

```
#!/bin/bash

HOST="https://api.piceasoft.com"

CLIENT_ID="F4E7D508058344F1A6E67B041508F21B"
CLIENT_KEY="yBnrVLkPskJnWPtte97zTQfU"

function _hash()
{
  echo -n "$CLIENT_KEY$CLIENT_ID$1" | shasum -a 256 | cut -d " " -f 1
}

DATA='{"device_uid":"046e76c2-f4ef-51d6-360e-9db08300b4e1" }'
HASH=`_hash "$DATA"`
curl -s  -H "Content-type: application/json" \
   -H "X-Piceasoft-Client-Id: $CLIENT_ID"    \
   -H "X-Piceasoft-Signature: $HASH"         \
   -X POST -d "$DATA" $HOST/reporting/v1/get_transactions_ex
```

PHP example:

```
// Private key and client ID shall be read from protected
// storage instead of hardcoding them
$privateKey = "G8nbQcdtzNwG5PUs";
$clientID = "0A0A9F08941E441890D380C43FE149E4";

// Construct request array
```

```
$request['imei'] = "356938035643809";

// Encode the request to JSON
$requestJSON = json_encode($request);

// Calculate signature
$signature = hash('sha256', $privateKey . $clientID . $requestJSON);

// Construct request...
$ch = curl_init('https://api.piceasoft.com/reporting/v1/get_transactions_ex');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $requestJSON);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'X-Piceasoft-Client-Id: ' . $clientID,
    'X-Piceasoft-Signature: ' . $signature,
    'Content-Type: application/json',
    'Content-Length: ' . strlen($requestJSON))
);

// ...and submit
$result = curl_exec($ch);
```

Private key is always customer specific and it needs to be kept as a secret value. Locate it always in the server side so that any unwanted party can't read it.

2 Report API

Request data is sent in HTTP POST method where the request data is in JSON format (HTTP header `Content-Type` must be set to `application/json`). HTTP header `Content-Length` must be set as well.

The request URL shall be in the following format:

```
https://api.piceasoft.com/reporting/v1/<command>
```

Response is always sent as JSON (content type `application/json`) and HTTP status code is 200 ("OK") when the request was processed; other HTTP status codes indicate HTTP-level failures, such as invalid method, wrong content type, missing mandatory headers and too long POST data.

Mandatory HTTP headers are as follows:

- `X-Piceasoft-Client-Id`
- `X-Piceasoft-Signature`
- `Content-Type` (expected **`application/json`**)
- `Content-Length`

Note also that `Transfer-Encoding` (such as chunked) is not supported.

The response generic format is as below:

```
{
  "status": <STATUS_CODE>,
   . . .  command specific response if <STATUS_CODE> is success
```

Status code values are as follow (value 0 is always success):

0	Success
-1	Internal error
1000	Signature error
1001	Authentication error (unknown client id)
1002	Invalid parameters in request JSON (mandatory parameters missing etc.)
1003	Database error
1004	Unknown command
1005	Invalid request content (incorrect JSON formatting)
1006	Server not ready yet
1007	Object not found

1008	Service is busy, retry later
1009	Access is denied
1010	Invalid PDF
1011	Language not supported
	More to be added

In case of error, `error_details` may be present to help troubleshooting:

```
{
 "status": 1002,
 "error_details": "'uid' parameter is missing from the request"
}
```

New error status codes may be added any time and the API client shall consider all nonzero status code values as errors.

Also new elements and new enumerations in the response can be added and the API client shall ignore those unknown to it.

For each element value, `null` means unknown. For example imei below:

```
"device_info": {
    "device_uid": "25ccdbfc-2f27-5eb0-8843-01ad8b10aaa6",
    "manufacturer": "Apple",
    "model_name": "iPhone",
    "model_code": "MRJN2",
    "os_version": "11.3",
    "serial_number": "F9HW6MDHJMVR",
    "memory_internal": 32,
    "imei": null
}
```


2.1 Fair API Usage

Please follow these guidelines when using the Reporting API.

- Do not issue several requests in parallel, wait until the first becomes ready before sending the next request.
- Use always the "startindex" method with `get_transactions(_ex)` to incrementally getting new transactions. Do not use the "startdate" parameter.
- Do not call `get_transactions(_ex)`, `get_transactions_with_imeis(_ex)` or `get_transactions_with_serials` more often than every few minutes.
- Do not use `get_transaction_details_ex` unless really needed.
- Always save data in your local data store got with `get_transaction_details_*`, the data won't change! The same applies to PDFs got with `get_pdf_repor`t and device information got with `get_device_data`.

2.2 get_transactions / get_transactions_ex

These commands return the list of transactions for a selected device or for all devices. Device needs to be identified by **uid, serial** or **imei** code value. Only one of these values shall be present, none if all transactions are listed.

Command get_transactions_ex – when using without type parameter – returns also transaction types stolen check, workflows, trade-in and sw update.

By default, this command returns all transactions within one month. Default start date can be modified by **startdate** parameter. For more accurate listings it is recommended to use **startindex** parameter instead. `startdate` refers to the timestamp set by the reporting client and it may take few seconds (even more, depending on the network quality and load in the server); therefore using `startdate` may result that some reports returned by the API are skipped.

2.2.1 Request

```
{
    "imei": "356938035643809",
    "device_uid": "002ebf12-a125-5ddf-a739-67c3c5d20177",
    "serial_number": "RZ8G80BHK3B",
    "product_ids":["d750c309-196b-99ba-8d3a-d14bf1f8d92d",
                   "745e77cd-d361-33dc-1c37-97f918cc5ecf"],
    "startdate": "2017-10-12",
    "startindex": 54325,
    "type": 1,
    "client_id": "ff2ebf12-a125-5ddf-a739-67c3c5d20177"
}
```

Parameter	Value	Mandatory	Description
device_uid	Device unique UID	No **1)**	Device unique UID generated by Piceasoft.
imei	Device IMEI code	No **1)**	Device IMEI code
serial_number	Device serial	No **1)**	Device serial number
startdate	Start date	No 2)	Optional start date for results. Default value is one month from calling date (unless startindex is used). Time needs to be given in the following date format YYYY-MM-DD Also more specific time can be given by using format YYYY-MM-DD hh:mm:ssZ (for example 2018-03-27 16:48:44Z). Either startdate or startindex shall be

			present; startindex is the preferred way.
startindex	Start index	No 2)	Each transaction is associated with a growing index integer number. To get initial list of transactions from the epoc, use number 0; the next listing request should use the index of the last transaction got from the previous listing, plus value 1.
type	Operation type	No	Optional operation type for filtering results. See list of supported operation types from Definitions section. By default, this command returns all operation types; note the difference between **`get_transactions_ex`** and **`get_transactions`**. Note also that **`get_transactions_ex`** can, in the future releases return also new transaction types.
client_id	Client unique ID	No	List only transactions generated by this client only. Note that the reporting API server has no knowledge of existing clients before one report for a particular client has been received.
product_ids	Array of product IDs	No	Include only products named in the array

1) Only one of these values needs to be present in the request, neither if transactions for all devices are listed.
2) Only one of these values needs to be present in the request; startindex is recommended

2.2.2 Response


```
{
 "status": 0,
 "transactions": [
    {
       "index":1011047,
       "type":1,
       "timestamp": "2017-10-12 12:00:00",
       "result": 0,
       "transaction_status": 1,
       "uid": "e84e610d-7d52-7521-8705-28ee199c5a89",
       "product_id": "745e77cd-d361-33dc-1c37-97f918cc5ecf",
       "client_id": "bbc27058-2a3f-8f1e-8c56-6f70c1f0e627",
       "client_name": "Shop Y",
       "client_address": "A City"
```



```
        },
        {  "index":1012047,
           "type": 2,
           "timestamp": "2017-10-12 12:00:00",
           "result": 0,
           "transaction_status": 1,
           "uid": "e84e610d-7d52-7521-8705-28ee199c5a89"",
           "product_id": "745e77cd-d361-33dc-1c37-97f918cc5ecf"",
           "client_id": "bbc27058-2a3f-8f1e-8c56-6f70c1f0e627",
           "client_name": "Shop Y",
           "client_address": "A City"
        },
        {  "index":1012048,
           "type": 3,
           "timestamp": "2017-10-12 12:00:00",
           "result": 0,
           "transaction_status": 1,
           "uid": "e84e610d-7d52-7521-8705-28ee199c5a89"",
           "product_id": "745e77cd-d361-33dc-1c37-97f918cc5ecf",",
           "client_id": "bbc27058-2a3f-8f1e-8c56-6f70c1f0e627",
           "client_name": "Shop Y",
           "client_address": "A City"
           "imei": "123456"
        }
     ]
  }
```


Value descriptions:

Tag	Description	Possible values / notes
index	Operation index	Unique, growing index number
type	Operation type	See list of supported operation types from **Definitions** section.
uid	Operation UID	Unique id of the operation. This can be used to query operation specific data.
timestamp	Operation timestamp in UTC time	
result	Operation result	See **Definitions** for possible result values.
transaction_status	Operation status	See **Definitions** for possible status values.
product_id	Originating product	
imei	IMEI of the device	Only if known, may be null
serial_number	Device serial number	Only if known, may be null
manufacturer	Device manufacturer	Only if known, may be null
model_name	Device model name	Only if known, may be null
model_code	Device model code	Only if known, may be null
client_id	Unique identifier of the client (PC)	
client_name	Given name of the client	
client_address	Given address of the client	
client_group	Given group of the client	
client_region	Given region of the client	
client_country	Given country of the client	

2.3 get_transactions_for_imeis / get_transactions_for_imeis_ex

These commands return the list of transactions for a list of devices identified by **imei** code value.

By default, this command returns all transactions within one month. Default start date can be modified by **startdate** parameter.

Command get_transactions_for_imeis_ex returns also transaction types stolen check, workflows, trade-in and sw update.

2.3.1 Request

```
{
    "imeis":["355808080985945","359217076041570","12345"],
    "startdate": "2017-01-01"
}
```

Parameter	Value	Mandatory	Description
imeis	Device IMEI codes	Yes	Array of device IMEI codes
startdate	Start date	No	Optional start date for results. Default value is one month from calling date. Time needs to be given in the following date format YYYY-MM-DD. Also more specific time can be given by using format YYYY-MM-DD hh:mm:ssZ (for example 2018-03-27 16:48:44Z)

2.3.2 Response

```
{
  "status": 0,
  "imeis": [
    {
      "imei": "355808080985955",
      "transactions": [
        {
          "type": 1,
          "uid": "714b4359-8de5-1044-43d0-cc7615b8fabb",
          "timestamp": "2017-08-15 04:08:41",
          "result": 0,
          "transaction_status": 1,
          "product_id": "56168609-365a-4584-aea2-0257f35deecc",
          "client_id": "d7688317-4d3c-c0a5-b4e4-df02dcda1eb0"
        }
      ]
    },
    {
```



```
      "imei": "359217076041507",
      "transactions": [
        {
          "type": 1,
          "uid": "854b867c-89d0-30f7-5aa7-7d36716bee91",
          "timestamp": "2017-01-09 09:24:46",
          "result": 0,
          "transaction_status": 1,
          "product_id": "56168609-365a-4584-aea2-0257f35deecc",
          "client_id": "04706fe0-c29f-032e-90b7-a359ead61bd5"
        },
        {
          "type": 1,
          "uid": "ed48922c-cca1-75a6-55c9-71969b0f8682",
          "timestamp": "2017-01-11 11:49:13",
          "result": 0,
          "transaction_status": 1,
          "product_id": "56168609-365a-4584-aea2-0257f35deecc",
          "client_id": "04706fe0-c29f-032e-90b7-a359ead61bd5"
        },
        {
          "type": 1,
          "uid": "0846cbf7-c0b8-e084-b2d3-e94b31d85f80",
          "timestamp": "2017-03-03 06:04:14",
          "result": 0,
          "transaction_status": 1,
          "product_id": "56168609-365a-4584-aea2-0257f35deecc",
          "client_id": "04706fe0-c29f-032e-90b7-a359ead61bd5"
        }
      ]
    },
    {
      "imei": "12345",
      "error": 1007
    }
  ]
}
```


For value descriptions refer to `get_transactions`. Note `"error"` in the example above; the value is the generic status code, in the example 1007 means that no transactions for that IMEI found.

Tag	Description	Possible values / notes
type	Operation type	See list of supported operation types from **Definitions** section.
uid	Operation UID	Unique id of the operation. This can be used to query operation specific data.
timestamp	Operation timestamp in UTC time	
result	Operation result	See **Definitions** for possible result values.
transaction_status	Operation status	See **Definitions** for possible status values.
product_id	Originating product	

2.4 get_transactions_for_serials

These commands return the list of transactions for a list of devices identified by **serial_number** value.

By default, this command returns all transactions within one month. Default start date can be modified by **startdate** parameter.

2.4.1 Request

```
{
    "serials":["RZ8G80BHK3B","F71W63GTHG7G"],
    "startdate": "2017-01-01"
}
```

Parameter	Value	Mandatory	Description
serials	Device serial number	Yes	Array of device serial numbers
startdate	Start date	No	Optional start date for results. Default value is one month from calling date. Time needs to be given in the following date format YYYY-MM-DD. Also more specific time can be given by using format YYYY-MM-DD hh:mm:ssZ (for example 2018-03-27 16:48:44Z)

2.4.2 Response

```
{
  "status": 0,
  "serials": [
    {
      "serial_number": " RZ8G80BHK3B",
      "transactions": [
        {
          "type": 1,
          "uid": "714b4359-8de5-1044-43d0-cc7615b8fabb",
          "timestamp": "2017-08-15 04:08:41",
          "result": 0,
          "transaction_status": 1,
          "product_id": "56168609-365a-4584-aea2-0257f35deecc",
          "client_id": "d7688317-4d3c-c0a5-b4e4-df02dcda1eb0"
        }
      ]
    },
    {
      "serial_number": "F71W63GTHG7G",
      "error": 1007
    }
  ]
}
```


For value descriptions refer to `get_transactions`. Note `"error"` in the example above; the value is the generic status code, in the example 1007 means that no transactions for that serial number found.

Tag	Description	Possible values / notes
type	Operation type	See list of supported operation types from **Definitions** section.
uid	Operation UID	Unique id of the operation. This can be used to query operation specific data.
timestamp	Operation timestamp in UTC time	
result	Operation result	See **Definitions** for possible result values.
transaction_status	Operation status	See **Definitions** for possible status values.
product_id	Originating product	

2.5 get_transaction_details

This command returns details of selected transaction.

2.5.1 Request

```
{
   "uid": "002ebf12-a125-5ddf-a739-67c3c5d20177",
   "type": 3,
   "get_custom_fields": false
}
```

Parameter	Value	Mandatory	Description
uid	Operation unique id	Yes	Unique identifier for the operation.
type	Transaction type	Yes	See list of supported operation types from **Definitions** section.
get_custom_fields	Boolean	No	true if custom fields are returned in the response, default false.

2.5.2 Response

All the responses contain `client_id`, `client_name`, `client_group`, `client_region`, `client_country` and `client_address` as in get_transactions(_ex). If get_custom_fields was true and custom fields exists, the response contain top-level custom fields key/value array as below:

```
"custom_fields": [
  {
    "key": "Key 1",
    "value": "Value 1"
  },
  {
    "key": "Key 2",
    "value": "Value 2"
  }
]
```


2.5.2.1 Switch

```
{
   "status": 0,
   "transaction_type":1,
   "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
   "duration":3,
   “result":0,
   "source":{
        "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
        "manufacturer":"General",
        "model_name":"One",...
   },
   "target":{
        "device_uid":"8ccf2662-c05c-bfd5-203f-d15961cfaca7",
        "manufacturer":"General",
        "model_name":null,...
   },
   "switch_data": {
      "content": [
         {“content_type”:2,"result":0,"added":356,"failed":0},
         {“content_type”:4,"result":0,"added":140,"failed":0},
         {“content_type”:7,"result":0,"added":1,"failed":0}
      ]
   }
}
```

Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Notification type	Switch. See notification types from the **Definitions**.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible result values from the **Definitions**.
source	Source device (see device_info of ‘get_device_details’	
target	Target device (see device_info of ‘get_device_details’	
content	Content types in transfer	

content_type	Value indicating the content type. See list of possible types from **Definitions**.

result	Operation result
added	Number of added items
updated	Number of updated items
duplicates	Number of duplicate items
notsupported	Number of not supported items
failed	Number of failed items

2.5.2.2 Erasure


```
{
    "status": 0,
    "transaction_type":2,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "duration":320,
    "result":0,
    "source":{
       "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "manufacturer":"Samsung",
       "model_name":"Galaxy A3",...
    },
    "erasure_type": 11,
 }
```


Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Erasure. See Definitions.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible result values from the Definitions.
source	Source device (see device_info of 'get_device_details')	
erasure_type	Type of erasure	Refer to document PICEASOFT REPORTING FORMATS

2.5.2.3 Diagnostics


```
{
    "status": 0,
    "transaction_type":3,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "duration":320,
    "result":0,
    "source":{
       "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "manufacturer":"Samsung",
       "model_name":"Galaxy A3", ...
    },
    "diagnostics_data": {
       "tests": [
          {"case_id":1,"result":0,"status":0},
          {"case_id":2,"result":0,"status":0}
       ]
    }
}
```


Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Diagnostics. See Definitions.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible result values from the Definitions.
failure_rate	Percentage of failed tests	0..100. NOTE: "canceled" is not considered to be a failure.
case_data	Additional case-specific data (optional)	See Definitions.
source	Source device (see device_info of 'get_device_details')	

2.5.2.4 Verify


```
{
    "status": 0,
    "transaction_type":4,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "duration":320,
    "result":0,
    "source":{
       "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "manufacturer":"Samsung",
       "model_name":"Galaxy A3",...
    },
    "verify_data": {
       "checks": [
          {“type”:1,"result":0,"status":0},
          {“type”:2,"result":0,"status":140}
       ]
    }
}
```


Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Verify. See Definitions.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible verify result values from the Definitions.
failure_rate	Percentage of failed checks	0..100
source	Source device	(see device_info of ‘get_device_details’)
checks/type	Type of the check	0 Unknown 1 LockCode 2 Memory card 3 SIM card 4 Device protection 5 Find my iPhone 6 Stolen check 7 Accounts 8 Root check 9 Carrier lock 10 Device genuiness 11 Battery genuiness 12 eSIM activation status

2.5.2.5 Stolen check


```
{
    "status": 0,
    "transaction_type":5,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "duration":320,
    "result":0,
    "source":{
       "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "manufacturer":"Samsung",
       "model_name":"Galaxy A3",...
    }
}
```


Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Stolen check. See **Definitions**.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible result values from the **Definitions**.
source	Device	(see device_info of 'get_device_details')

2.5.2.6 Workflow/Trade-in

```
{
    "status": 0,
    "transaction_type":6,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "source": {
        "device_uid": "ce80bf77-984e-45dd-b1ad-cbe8eb6d2d55",
        "manufacturer": "Apple",
        "model_name": "iPhone 5",
        "model_code": "MD298",
        "os_version": "10.3.3",
        "serial_number": "F2NK28MDDTWF",
        "memory_internal": 16,
        "imei": "123456"
    },
    "transactions":[{
       "uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "type”:2
    }]
}
or
{
    "status": 0,
    "transaction_type":10,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "source": {
        "device_uid": "ce80bf77-984e-45dd-b1ad-cbe8eb6d2d55",
        "manufacturer": "Apple",
        "model_name": "iPhone 5",
        "model_code": "MD298",
        "os_version": "10.3.3",
        "serial_number": "F2NK28MDDTWF",
        "memory_internal": 16,
        "imei": "123456"
    },
    "transactions":[{
       "uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "type”:2
    }]
}
```

Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Workflow/Trade-in workflow. See **Definitions**.
uid	Operation UID	
source	The device	The following properties may be present: - device_uid - manufacturer - model_name - model_code - os_version - serial_number - memory_internal

		- imei
transactions	Table of related transactions	Object contains the transaction ID (**uid**) and **type**. Note that if type has value null, it means that no details for the transaction is available. If uid is null, the workflow phase was skipped.

2.5.2.7 Software update

```
{
    "status": 0,
    "transaction_type":7,
    "uid":"e84e610d-7d52-7521-8705-28ee199c5a89",
    "duration":320,
    "result":0,
    "source":{
       "device_uid":"e45bd7fd-7efa-d7b9-8add-313d78c864bd",
       "manufacturer":"Samsung",
       "model_name":"Galaxy A3",...
    }
}
```

Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	SW update check. See **Definitions**.
uid	Operation UID	
duration	Duration is seconds	
result	Operation result	See list of possible result values from the **Definitions**.
source	Device	(see device_info of 'get_device_details')

2.5.2.8 Trade-In


```
{
  "status": 0,
  "transaction_type": 9,
  "uid": "bbe8b388-ac00-462f-93c8-864d4482c80d",
  "trading_data": {
    "timestamp": "2018-09-26T10:10:10.000Z",
    "deal_accepted": 1,
    "vendor_id": "894ce586-126b-4b1c-b100-f5c4d3bc0f19",
    "vendor_name": "A trader,
    "price": "20.75",
    "currency_code":"EUR",
    "offer_id": "107cb06a-c1c8-4d10-a26f-593ee2eca445",
    "valid_until": "2018-09-17T10:32:16.000Z",
    "device_uid": "CCCAA2FF-B6DD-40B1-A53F-E46BBA93F711",
    "imei": "123456",
    "client_id": "14e2d132-18a4-4e35-a455-b3cd73c44a22",
    "client_name": "A client",
    "client_address": "Address",
    "customer_id": "b6163faf-5678-4864-1234-303d4a605200"
  }}
```


Value descriptions:

Tag	Description	Possible values / notes
transaction_type	Report type	Trade-in. See content **Definitions**.
uid	Operation UID	
timestamp	Date	
deal_accepted	Status of the deal	1 if the deal was accepted, 0 if not and null in case of error
vendor_id	Unique ID of the vendor	
vendor_name	Name of the vendor	
price	Price of the device paid to the vendor	
currency_code	Currency	
offer_id	Unique ID of the offer	
valid_until	Date, how long the deal is valid	
device_uid	ID of the device	See get_device_details
imei	IMEI of the device	
customer_id	ID of the customer	

2.6 get_transaction_details_ex

This command returns the original JSON report of the selected transaction. This command should be used only if the data returned in **get_transaction_details** is not complete enough.

2.6.1 Request

```
{
   "uid": "002ebf12-a125-5ddf-a739-67c3c5d20177",
   "type": 3
}
```

Parameter	Value	Mandatory	Description
uid	Operation unique id	**Yes**	Unique identifier for the operation.
type	Transaction type	**Yes**	See list of supported operation types from **Definitions** section.

2.6.2 Response

If the JSON report was found, the original report is returned in "`transaction_details`" property. The JSON format is specified in document PICEASOFT REPORTING FORMATS.
Example:

```
{
  "status": 0,
  "transaction_details": {
    "uid": "B7E8B388-AC00-462F-93C8-864D4482C800",
    "application": {
      "application_id": "56168609-365A-4584-AEA2-0257F35DEECC",
      "name": "PiceaOne",
      "version": "2.2.221.38965",
      "type": "RD",
      "language": "de",
      "package_name": "com.piceasoft.piceaone",
      "libraries": [
        {
          "library_id": "56168609-365A-4584-AEA2-0257F35DEECC",
          "name": "Verify",
          "version": "2.2.221.38965",
          "type": "RD",
          "package_name": "com.piceasoft.piceaone"
        }
      ]
. . .
```

If the JSON report is not found, the response is the same as in `get_transaction_details`. Note that due to the complexity of the result JSON it is not recommended to use this command unless the data needed is not available in the `get_transaction_details` command.

2.7 get_device_details

This command returns details of a selected device.

2.7.1 Request

```
{
   "device_uid":"002ebf12-a125-5ddf-a739-67c3c5d20177"
}
```

Parameter	Value	Mandatory	Description
device_uid	Device unique id	Yes	Unique identifier for the device.

2.7.2 Response

```
{
   "status": 0,
   "device_info": {
       "device_uid": "ae45f4c0-f33b-41ca-b787-a33bf14ec8ee",
       "manufacturer": "Apple",
       "model_name": "iPhone 7",
       "model_code": "MN8Y2",
       "os_version": "12.1.4",
       "serial_number": "C7KSWNJWGH7G",
       "memory_internal": 32,
       "imei": "1234567",
       "device_model_code": "MN8Y2B/A",
       "color_code": "2",
       "color": "Silver",
       "battery": "Non-removable Li-Ion 1960 mAh battery (7.45 Wh)",
       "product_code": "B/A",
       "model_number": "A1778",
       "hw_model": "D101AP",
       "has_gsm": true,
       "alternative_model_codes": [
         "FN8Y2",
         "NN8Y2",
         "PN8Y2",
         "3N8Y2"
       ],
       "fid": "49cffeff-b43b-46b7-ac2b-da9c7ea741be"
   }
}
```

Value descriptions:

Tag	Description	Possible values / notes
device_uid	Device UID	Unique device id
manufacturer	Device manufacturer	
model_name	Device model name	
model_code	Device model code	
model_region_code	Model code with region info (*	Only if known, for example (Apple iPhone 62) model_code is MN112 and

		model_region_code is MN112RU/A
brand_name	Device brand name (*	Optional
product_code	Device product code (*	Optional
model_number	Device model number (*	Optional
hw_model	Device HW model (*	Optional
alternative_model_codes	Alternative model codes (*	Optional, array
os_version	Device software version	Optional
serial_number	Device serial number	Optional
imei	Device IMEI code	Optional
imei2	Device IMEI2 code (*	Optional
memory_internal	Internal memory	Internal memory size in GBs, optional
color	Device color (*	Optional
color_code	Device color code (*	Optional
fid	Device format id (*	Optional. See Device formats in Definitions.
battery	Info on device battery (*	Optional, example: "battery": "Non-removable Li-Po 1810 mAh battery (6.9 Wh)"
has_cdma	If phone has CDMA (*	Optional; true if the model has CDMA, false if not (property does not exisis if not known)
has_gsm	If phone has GSM (*	Optional; true if the model has GSM, false if not (property does not exisis if not known)
has_euicc	If phone has eSIM (*	Optional; true if the model has eSIM, false if not (property does not exisis if not known)

NOTE: properties marked with (* can be present only by using function `get_device_details`, e.g. `get_transaction_details(_ex)` does never have those.

2.8 get_devices

This command returns a list of device models known to Piceasoft reporting database.
Note that this command requires special access rights, typically also a separate API key.

2.8.1 Request

```
{
}
```

2.8.2 Response

```
{
   "status": 0,
   "devices": [
       {
         "manufacturer": "Apple",
         "model_name": "iPhone 6s Plus",
         "variant_id": "apple@iphone6splus@mkvq2@16",
         "model_code": "MKV72",
         "alternative_model_codes": [
           "FKV72",
           "NKV72",
           "PKV72",
           "3KV72"
         ],
         "color": "Rose Gold",
         "memory_internal": 16,
         "brand_name": "",
         "aca": null,
         "has_cdma": true,
         "has_euicc": null,
         "has_gsm": true
       },
       {
         "manufacturer": "Samsung",
         "model_name": "Galaxy Tab A 7.0 (2016)",
         "variant_id": "samsung@galaxytaba7.0(2016)@sm-t285@",
         "model_code": "SM-T285M",
         "alternative_model_codes": null,
         "color": null,
         "memory_internal": 8,
         "brand_name": "",
         "aca": [
           "Galaxy Tab A 7.0"
         ],
         "has_cdma": false,
         "has_euicc": null,
         "has_gsm": true
       },
...
   ]
}
```


Value descriptions (optional means that the value may be null or ""):

Tag	Description	Possible values / notes
manufacturer	Device manufacturer	
model_name	Device model name	
variant_id	Unique model ID	Can be used as a key to a database
model_code	Device model	
alternative_model_codes	Alternative model codes	Array of alternative model codes, optional
memory_internal	Internal memory	Internal memory size in GBs, optional
brand_name	Device brand name	Optional
aca	“Also called as”	Array of "also-called-as" model names, optional
color	Device color(s)	Optional
has_cdma	If phone has CDMA	Optional; true if the model has CDMA, false if not, null if not known
has_gsm	If phone has GSM	Optional; true if the model has GSM, false if not, null if not known
has_euicc	If phone has eSIM	Optional; true if the model has eSIM, false if not, null if not known

2.9 get_pdf_report

This command returns the PDF report of the selected transaction.

2.9.1 Request

```
{
   "uid": "002ebf12-a125-5ddf-a739-67c3c5d20177"
}
```

Parameter	Value	Mandatory	Description
uid	Operation unique id	Yes	Unique identifier for the operation.

2.9.2 Response

The response content type is `application/pdf` in case the PDF retrieval was success, and `application/json` in case of failure (the nonzero status value in the JSON)

2.10 get_file

This command returns the report attachment.

2.10.1 Request

```
{
   "uid": "002ebf12-a125-5ddf-a739-67c3c5d20177",
   "file_id": "5DBEBD3C-1880-4615-B03A-33B0D04D4D4B"
}
```

Parameter	Value	Mandatory	Description
uid	Report unique id	Yes	Unique identifier for the report.
file_id	File ID	Yes	Unique identifier for the file (report attachment).

2.10.2 Response

The response content type is the attachment mime type.

3 Definitions

3.1 Device formats

Fid value	Description
b7463bb4-fffe-cf49-be41-a77e9cda2bac	iOS (iPhone)
3044cd8e-bd26-2884-17ea-326e9acb5d95	iOS (iPad)
64486dd7-af3f-16f6-0052-604a3d30e196	Android
5e4abe5d-f76d-df45-20e9-9c80a16dcd9c	Windows Phone
88449182-2246-cd48-1ba9-21da53519f8f	Nokia Series 40
b449d9a7-637c-5de0-d411-96ff0db3ed8c	Nokia S60 (Symbian)
36413fb0-d1c2-5b26-6857-06a710a65790	Blackberry
dd4e1de3-baff-1bc2-bbc5-8b758658d2af	USB mass storage
064bc831-59ce-bc9b-6d53-5b66f90f9784	Unknown USB device
49cffeff-b43b-46b7-ac2b-da9c7ea741be	iOS (iPhone)
842826bd-8ecd-4430-955d-cb9a6e32ea17	iOS (iPad)
f6163faf-d76d-4864-96e1-303d4a605200	Android
45df6df7-5dbe-4a5e-9ccd-6da1809ce920	Windows Phone
48cd4622-8291-4488-8f9f-5153da21a91b	Nokia Series 40
e05d7c63-a7d9-49b4-8ced-b30dff9611d4	Nokia S60 (Symbian)
265bc2d1-b03f-4136-9057-a610a7065768	Blackberry
c21bffba-e31d-4edd-afd2-5886758bc5bb	USB mass storage
9bbcce59-31c8-4b06-8497-0ff9665b536d	Unknown USB device

3.2 Operation types

Type value	Description
1	Switch
2	Erasure
3	Diagnostics
4	Verify
5	Stolen check
6	Workflow
7	Software update
8	Action (internal)
9	Trade-in
10	Trade-in workflow

3.3 Content types

Value	Description
0	Undefined
1	Contacts
2	Calendar
3	Messages
4	Bookmarks

5	Music
6	Images
7	Videos
8	Documents
9	Applications
10	Call logs
11	Optional apps
12	Preload apps
20	Restore
21	Rollback

3.4 Transaction status values

Value	Description
0	Undefined
1	Succeeded
2	Failed
3	Empty run (no succeeded content)

3.5 Verify result values

Status code	Description
0	Passed
1	Unknown
32769	Not supported
32770	Security lock enabled
32771	Memory card inserted
32772	SIM card inserted
32773	Device protection enabled (Android)
32774	Find My iPhone enabled (iOS)
32775	Device is marked as stolen
32776	Accounts found
32777	Device not stolen, but something unclear.
32778	Device is rooted
32779	Device is protected with Xiaomi account
32780	Device is protected with Samsung account
32781	Device carrier locked
32782	Device might not be genuine
32783	eSIM activated
32784	Device battery may not be genuine
32785	Stolen check failure
32786	MDM profile install needed
< 0	See failure codes in chapter Result codes

3.6 Diagnostics test cases

0	General
1	Stolen check
2	Rooted check
3	General auto
4	General manual
5	Warranty bit
6	Device status
7	Find my iPhone
9	User accounts
10	Device genuiness
16	Speaker
17	Microphone
18	Call speaker
19	Record and play
32	Front camera
33	Rear camera
34	Front camera auto focus
35	Rear camera auto focus
36	Flash
37	Front camera video
38	Rear camera video
48	Multitouch
49	Touch screen
50	Screen colors
51	Bad pixels
52	Force touch
53	Screen burns
54	Pinch
64	Back button
65	Volume up button
66	Volume down button
67	Mute button
68	Menu button
69	Power button
70	Home button
71	Camera button
80	Network
81	Bluetooth
82	GPS

83	NFC
84	SIM
85	Wi-Fi
86	Telephony
87	Mobile data
96	Light sensor
97	Magnetic sensor
98	Accelerometer
99	Gyroscope
100	Proximity sensor
101	Pressure sensor
102	Vibration
103	Finger print
105	Face ID
112	USB connector
113	Headset connector
114	SD card connector
115	Charger connector
128	Software updates
129	Security update
130	Device uptime
131	Device sleeptime
132	RAM usage
133	Storage usage
134	Applications
134	Malware
136	Software crashes
137	Device settings
138	Device status
139	Find My iPhone
144	Battery health *1)
145	Battery performance *5)
146	Battery optimization
160	Back cover condition grade *2)
161	Screen grade *2)
162	Body grade *2)
163	Water Damage Indicator
164	Device grade *2)
165	Sales box content *3)
166	Description (free text) *4)
176	USB cable
177	Headset

178	SD card
179	Charger
180	SPen (Samsung pen)
181	Apple pencil
182	SIM tool
183	Manual
184	Wireless charger
192	Network speed
193	Memory speed
194	Internal storage speed
195	External storage speed
196	CPU speed
208	Front camera chipset
209	Back camera chipset
210	Flash chipset
211	Charger chipset
212	Microphone chipset
213	Fingerprint chipset
214	NFC chipset

*1) Battery health `case_data` is as follows:

```
        "case_data": {
             "type": "4",
             "value": {
                   "batterygrade": <GRADE>,
                   "health": <HEALTH>,
                   "cyclecount": <CYCLECOUNT>
               }
          }
   <GRADE> (for iOS devices only) is a number between 0 (worst) .. 5 (best)
   <HEALTH> is as follows:
             1 = Unknown
             2 = Good
             3 = Overheat (Android only)
             4 = Dead (Android only)
             5 = Over voltage (Android only)
             6 = Unspecified failure (Android only)
             7 = Cold (Android only)
             8 = Poor (iOS only)
             9 = Bad (iOS only)
            10 = Excellent (iOS only)
   <CYCLECOUNT> is the number of charging cycles
```


*2) Visual condition `case_data` is as follows:

```
        "case_data": {
             "type": "8",
             "value": {
                   "grade": <GRADE>
               }
          }
   <GRADE> is a number between 0 (not available) .. 6 (A+),
   255 means not graded at all (canceled)
```

*3) Sales-box content `case_data` is as follows:

```
        "case_data": {
           "type": "8",
            "value": {
               "charger": <VALUE>,
               "guides": <VALUE>,
               "headset": <VALUE>,
               "receipt": <VALUE>,
               "simtool": <VALUE>,
               "usbcable": <VALUE>
          }
        }
   <VALUE> is 0 when item is not present, 1 if it is.
```

*4) Description `case_data` is as follows:

```
         "case_data": {
           "type": "8",
            "value": {
                  "text":"entered freeform text"
               }
         }
```

*5) Battery performance `case_data` is as follows:

```
        "case_data": {
             "type": "4",
             "value": {
                   "battery_grade_offline": <GRADE>
               }
          }
   <GRADE> (for Android only) is a number between 0 (worst) .. 5 (best)
```

A case can also contain attachment data. The `case_data` is as follows:

```
        "case_data": {
             "type": "7",
             "value": {
                   "file_mime_type": <MIME_TYPE>
                   "file_id": <FILE_ID>
```



```
            }
        }
<FILE_ID> is a string
<MIME_TYPE> is the Mime type, for example image/jpeg
```

The attachment can be fetched with function `get_file`.

3.7 Result values

3.7.1 Success codes

Result code	Description
0 and 1	Succeeded
2	Succeeded, but no succeeded items
3	Succeeded, but reboot is required
4	Succeeded, but exceptions to be notified

3.7.2 Failure codes

In general all negative result values indicate error.

The following table contains generic error codes.

Failure code	Description
-1	Undefined error
-2	Generic failure
-3	Diagnostics test case failed
-4	Diagnostics test case skipped
-5	Diagnostics test case canceled
-6	Diagnostics test case not executed
-7	Diagnostics test case resulted to warning
-8	Diagnostics test case not implemented
-9	Diagnostics test case not supported by device

For error codes from PC application refer to document PICEASOFT REPORTING FORMATS, chapter “Error codes”.

4 Sample Code

Sample code can be downloaded as a tgz-tarball from
https://api.piceasoft.com/reporting/v1/get_samples

Please refer to file README.txt.

