History SDK

for

Version 4.5

Table of Content

1 Introduction

1.1 Document Purpose

The document is intended to provide description of the History API used to retrieve the history of operations performed in ETAdirect. The data retrieved by the History API can be further used by external applications.

1.2 Scope of the Document

This document describes the History API request used to retrieve the history records and responses received for such request.

This document describes the functionality implemented in ETAdirect version 4.5.12.

1.3 Target Audience

The document is intended for developers and programmers working with the ETAdirect history in order to integrate ETAdirect with external systems.

1.4 Glossary

Term	Explanation
Activity	Entity of the ETAdirect system that represents any time-consuming activity of the resource
API	Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other
Company	1) Legal entity, using ETAdirect 2) Entity that represents a Client in ETAdirect system; company is created by TOA Technologies during the process of implementation
Field	Property present in the system by default
Inventory	Equipment that can be installed or deinstalled during an activity
ISO 8601 format	see http://en.wikipedia.org/wiki/ISO_8601
Linked activities	Two separate activities related so that the completion or start of one is dependent on the completion or start of the other
Property	Field and field value assigned to an entity in ETAdirect (to user, resource, activity or inventory). There are fields and custom properties
Route	List of activities assigned to a resource for a specific date, or a list of non-scheduled activities assigned to a resource
Required Inventory	Inventory necessary for the activity completion
Resource	Element in the resource tree representing a defined company asset
REST	Representational State Transfer, architectural style running over HTTP
Request	Request sent by the Mobile Client to trigger a certain message scenario

2 History API Introduction

ETAdirect supports logging of actions performed to various entities. Such actions are recorded in the history. The History API serves as a means of retrieving such actions and their details for further use by external applications. The history logs the operation performed to an entity, the time of such operation, the user which performed the operation and the actual changes made to the entity.

3 Authentication

3.1 REST API Authentication Methods

REST APIs (including the History API) support the following methods of authentication:

- Query parameters (authentication fields are sent in query parameters)
- Custom HTTP header (authentication fields are sent in a single query parameter and an HTTP header)
- HTTP Basic (authentication fields are sent as part os a standard "Authorization" HTTP header. This method can be used to access APIs directly from WEB browsers.

3.1.1 Authentication Using Query Parameters

REST APIs can authenticate by two parameters added to the URL query string:

- **company** or **i** (parameter containing the instance name)
- **toa-token** (parameter containing a base64-encoded JSON-object with the following fields:
  - now – current date in ISO 8601 format
  - company – instance name
  - login – user's login
  - auth_string – password hash in MD5 form

Example of URL containing authentication data in query parameters:

```
https: //api.etadirect.com/rest/history/v1/?company=test.instance&toa-
token=eyJodXJyIjoiZHVyciIsImhlcnAiOiJkZXJwIn0=
```

3.1.2 Authentication Using Custom HTTP Header

REST APIs can authenticate by one parameter added to the URL query string:

- **company** or **i** (parameter containing the instance name)

Also, a custom HTTP header is sent in the request:

- **toa-token** (header containing a base64-encoded JSON-object with the following fields:
  - now – current date in ISO 8601 format
  - company – instance name

- login – user's login
- auth_string – password hash in MD5 form

Example of HTTP request header containing the toa-token header and company query parameter:

```
GET /rest/history/v1/?company=test.instance HTTP/1.0
User-Agent: curl/7.31.0
Host: api.etadirect.com
Accept: */*
toa-token: eyJodXJyIjoiZHVyciIsImhlcnAiOiJkZXJwIn0=
```

3.1.3 Authentication Using HTTP Basic

REST APIs can authenticate by one parameter added to the URL query string:

- **company** or **i** (parameter containing the instance name)

Also, a standard HTTP header is sent in the request:

- **Authorization** (header containing a base64-encoded string consisting of the login and a plain-text password)

Example of HTTP Basic header containing the Authorization string and company query parameter:

```
GET /rest/history/v1/route/?company=test.instance HTTP/1.0
User-Agent: curl/7.31.0
Host: api.etadirect.com
Accept: */*
Authorization: YXBpLnVzZXI6U0VDUkVU
```

Authenticating using HTTP Basic can be done directly from the WEB browser. For that purpose, the user has to enter the REST API URL in the browser address bar and append the special parameter 'request_auth_basic':

```
https: //api.etadirect.com/rest/history/v1/?
company=test.instance&request_auth_basic
```

The user is then requested to enter their ETAdirect login and password in the dialog window. IF authentication is successful, the user is able to use the API.
More information on HTTP Basic authentication method can be found at http://en.wikipedia.org/wiki/Basic_access_authentication.

Note: HTTP Basic authentication method is not recommended over non-encrypted channels.

4 History API Permission

Access to the History API is controlled by the permission defined in the Manage Application → Permissions → SOAP.

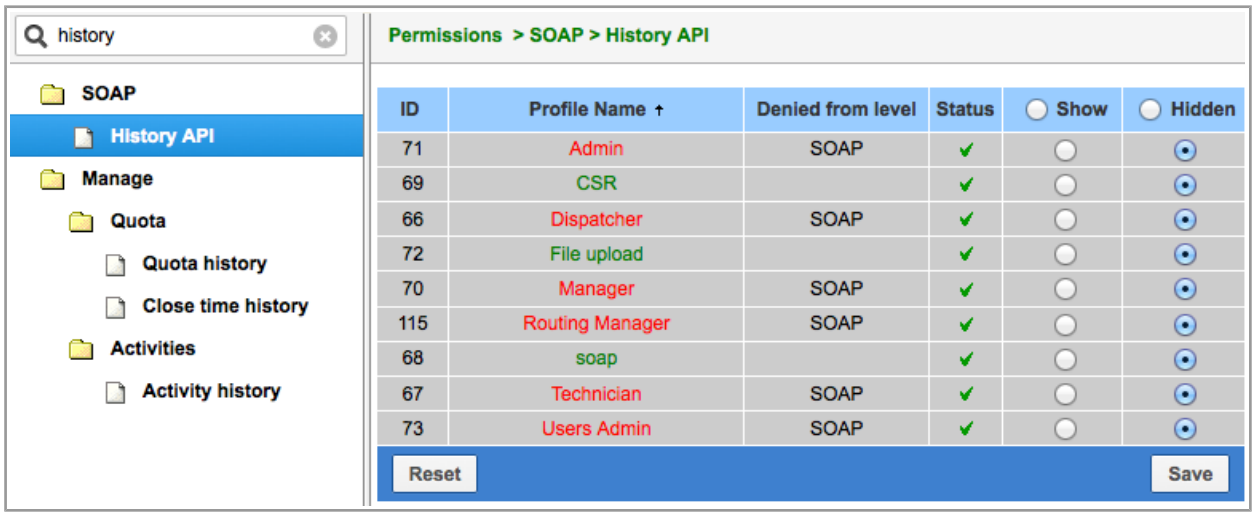

Figure 1: History API permission

Each user is able to retrieve the history records available to the resource assigned to such user.

5 History API Operation Description

5.1 Entities Monitored by History

The History API returns history records of changes to the following entities of ETAdirect:

- route
- activity
- activity link
- resource preference
- required inventory
- inventory
- request

5.2 History API Request

The History API uses a single method, namely, the GET method, to retrieve all history for all entities in ETAdirect. The GET request has the following format:

```
GET /rest/history/v1/route/?company=test.instance&count=150&token=140115-808,27
```

where:

- company – company name or i as used in authentication
- count – the maximum number of records to be returned in the response (the maximum value: 1000, the default value: 100)
- token – string defining the point from which history is to be returned. The token is created at

time of the initial valid request and remains valid for the next 12 hours. Each subsequent response contains a new token. If a request is sent with the same token as the previous request, the response contains the history returned for the previous request plus all changes logged after that. However, if a request is sent with a token for which the 12-hour validity period has expired, the request is not processed. If a request is sent with the new token, the response contains changes logged after such token creation.

- optionally, the request can contain the 'debug' parameter allowing data to be returned in human-friendly format (**Note**: a response in such format will not be processed by the external application.

Note: The history records are available for 12 hours. The maximum number of history records available for retrieval is 20,000. The number of available records is shared between all users simultaneously accessing the history.

5.3 History API Response

Each History API response always contains a package header.

The package header contains the request result ('true' for a valid request and 'false' for an invalid one) and the next token which can be used in subsequent history requests. The package header is followed by the history records.

Note: for the sake of clarity, all examples below are shown in the human-friendly format.

```
{
   "found":true,
   "next_token":"140108-571,0",
   "history":[
      HISTORY RECORDS
   ]
}
```

The history data returned in a response in organized in records, each containing one change to an ETAdirect entity. Each record has a record header consisting of the following fields:

- "operation" – name of the operation logged in the history
- "action_time" – time of the action
- "user" – ETAdirect user performing the action. If the action was performed by the system, the "user" field is omitted
- "realm" – field indicating that the record was updated by a maintenance user, otherwise omitted

```
{
    "operation": OPERATION,
    "action_time": "2014-01-15 12:58:56",
    ["user": "admin",]
    ["realm":"maintenance",]
    ENTITY: {
```

```
        IDENTIFIER,
        "changes": [
            CHANGES
        ]
    }
}
```

The header is followed by the name of the entity, the fields identifying the particular entity and the description of changes performed to the entity.

5.3.1 History API Response Description

Below is the description of history records logged for each ETAdirect entity monitored by history.

Note: the History API logs the same operations which are used in respect of the corresponding entities by the Mobile Client API.

5.3.1.1 Route Updates

The following route operations are logged in the history and returned by the History API:

- create_route
- update_route
- start_route
- end_route
- restart_route

Each route-related record contains the route identifier fields:

- date
- resource_id

The records may optionally contain the "changes" field containing the fields and their values changed for the route. The "changes" field logs updates of the following fields:

- activated
- deactivated
- calendar_time_from
- calendar_time_to
- calendar_points
- time_zone

5.3.1.2 Route Update Response Example

The example below shows the history record of an activated route creation:

```
{
    "operation": "create_route",
```

```
    "action_time": "2014-01-15 13:36:54",
    "user": "admin",
    "route": {
        "date": "2014-01-15",
        "resource_id": "33036",
        "changes": {
            "activated": "2014-01-15 13:36:00",
            "time_zone": "Eastern",
            "calendar_time_from": "2014-01-15 12:00:00",
            "calendar_time_to": "2014-01-15 21:00:00",
            "calendar_points": "100"
        }
    }
}
```


5.3.1.3 Activity Updates

The following activity operations are logged in the history and returned by the History API:

- create_activity
- update_activity
- start_activity
- suspend_activity
- complete_activity
- notdone_activity
- cancel_activity
- delete_activity
- delay_activity
- reopen_activity
- prework_activity

For the following operations the identifier fields contain old values and the "changes" field contains new values:

- move_activity
- reschedule_activity

Each activity-related record contains the activity identifier fields:

- date (the date of the route to which the activity is assigned)
- resource_id (external ID of the resource to which the activity is assigned)

- activity_id (internal activity ID)
- appt_number
- customer_number

The records may optionally contain the "changes" field containing the fields and their values changed for the activity. The "changes" field logs updates of the following fields:

- activity fields and custom properties updated in the operation

5.3.1.4 Activity Update Response Example

The example below shows the history record of a started activity creation:


```
{
    "operation": "create_activity",
    "action_time": "2014-01-15 16:34:29",
    "user": "admin",
    "activity": {
        "date": "2014-01-15",
        "resource_id": "33011",
        "activity_id": 3998009,
        "changes": {
            "ACTIVITY_NOTES": "just lunch",
            "type": "regular",
            "status": "started",
            "worktype": "LU",
            "duration": "60",
            "start_time": "2014-01-15 16:34:00",
            "time_slot": "lunch",
            "service_window_start": "2014-01-15 20:00:00",
            "service_window_end": "2014-01-15 20:30:00",
            "language": "en",
            "time_zone": "Pacific",
            "position_in_route": "1",
            "time_of_booking": "2014-01-15 16:34:29",
            "time_of_assignment": "2014-01-15 16:34:29"
        }
    }
}
```


The example below shows the history record of an activity move between resources. In this case, the identifier field "resource_id" contains the old value, while the "resource_id" field in "changes" contains the new one:


```
{
    "operation": "move_activity",
    "action_time": "2014-01-15 17:34:28",
    "user_id": "admin",
```

```
    "activity": {
        "date": "2014-01-15",
        "resource_id": "33015",  // old resource id
        "activity_id": 3956550,
        "appt_number": "#137163458",
        "customer_number": "019942164",
        "changes": {
            "resource_id": "33011"  // new resource id
        }
    }
},
{
    "operation": "update_activity",
    "action_time": "2014-01-15 17:34:28",
    "user": "admin",
    "activity": {
        "date": "2014-01-15",
        "resource_id": "33011",  // new resource id
        "activity_id": 3956550,
        "appt_number": "#137163458",
        "customer_number": "019942164",
        "changes": {
            "duration": "30",
            "start_time": "2014-01-16 00:13:52",
            "traveling_time": "11",
            "delivery_window_start": "2014-01-15 23:30:00",
            "delivery_window_end": "2014-01-16 00:45:00",
            "position_in_route": "12",
            "time_of_assignment": "2014-01-15 17:34:28"
        }
    }
}
```

5.3.1.5 Activity Link Updates

The following activity link operations are logged in the history and returned by the History API:

- link_activities
- unlink_activities

Each activity link-related record contains the link identifier fields:

For the first activity:

- link_type
- from_activity
- to_activity_id

- to_appt_number

For the second activity:

- link_type
- to_activity
- from_activity_id
- from_appt_number

5.3.1.6 Activity Link Update Response Example

The example below shows the history record of a link creation. One activity link is created, but the request returns two responses – for the first activity in the link and for the second activity:


```
{
    "operation": "link_activities",
    "action_time": "2014-01-15 16:54:48",
    "user": "admin",
    "activity_link": {
        "link_type": "start-before",
        "from_activity": {
            "date": "2014-01-15",
            "resource_id": "33003",
            "activity_id": 3956464,
            "appt_number": "#137163544",
            "customer_number": "019922286"
        },
        "to_activity_id": 3954821,
        "to_appt_number": "#137165187"
    }
},
{
    "operation": "link_activities",
    "action_time": "2014-01-15 16:54:48",
    "user": "admin",
    "activity_link": {
        "link_type": "start-after",
        "to_activity": {
            "date": "2014-01-15",
            "resource_id": "routing",
            "activity_id": 3954821,
            "appt_number": "#137165187",
            "customer_number": "019911355"
        },
        "from_activity_id": 3956464,
        "from_appt_number": "#137163544"
    }
}
```

5.3.1.7 Resource Preference Updates

The following resource preference operations are logged in the history and returned by the History API:

- set_resource_preference
- del_resource_preference

Each preference-related record contains the resource preference identifier fields:

- activity
- type (preferred, forbidden, required)
- resource_id

5.3.1.8 Resource Preference Update Response Example

The example below shows the history record of a preferred resource adding:

```
{
    "operation": "set_resource_preferences",
    "action_time": "2014-01-15 17:05:16",
    "user": "admin",
    "preference": {
        "activity": {
            "date": "2014-01-15",
            "resource_id": "33003",
            "activity_id": 3956464,
            "appt_number": "#137163544",
            "customer_number": "019922286"
        },
        "resource_id": "11129",
        "type": "preferred"
    }
}
```

5.3.1.9 Required Inventory Updates

The following required inventory operations are logged in the history and returned by the History API:

- create_required_inventory
- update_required_inventory
- delete_required_inventory

Each required inventory-related record contains the required inventory identifier fields:

- model

- type

The records may optionally contain the "changes" field containing the "quantity" field, if the inventory quantity was changed in the operation (for non-serialized inventory).

5.3.1.10 Required Inventory Update Response Example

The example below shows the history record of required inventory adding to an activity:

```
{
    "operation": "create_required_inventory",
    "action_time": "2014-01-28 12:14:02",
    "user": "admin",
    "required_inventory": {
        "type": "Wire",
        "model": "RG-45",
        "activity": {
            "date": "2014-01-28",
            "resource_id": "33035",
            "activity_id": 3954885,
            "appt_number": "137165123",
            "customer_number": "019921925"
        },
        "changes": {
            "model": "RG-45",
            "quantity": "5"
        }
    }
}
```

5.3.1.11 Inventory Updates

The following inventory operations are logged in the history and returned by the History API:

- install_inventory
- deinstall_inventory
- create_customer_inventory
- delete_inventory
- undo_install_inventory
- undo_deinstall_inventory
- exchange_inventory_install
- exchange_inventory_deinstall

Records of the following operations contain no activity identifier:

- update_customer_inventory

- create_resource_inventory
- update_resource_inventory
- delete_resource_inventory

Each inventory-related record contains the inventory identifier fields:

- inventory_id
- serial_number
- type

The records may optionally contain the "changes" field containing the fields and their values changed for the inventory. The "changes" field logs updates of the following fields:

- status
- custom inventory properties updated in the operation

5.3.1.12 Inventory Update Response Example

The example below shows the history record of inventory installation:

```
{
    "operation": "install_inventory",
    "action_time": "2014-01-15 17:11:57",
    "user": "admin",
    "inventory": {
        "activity": {
            "date": "2014-01-15",
            "resource_id": "33003",
            "activity_id": 3956439,
            "appt_number": "#137163569",
            "customer_number": "019911470"
        },
        "inventory_id": 20998758,
        "type": "NT",
        "status": "install",
        "serial_number": "0001757132",
        "changes": {
            "status": "install",
            "EQUIPMENT_ROOM_CODE": "2354235488"
        }
    }
}
```

The following example shows the history record of customer inventory creation:

```
{
    "operation": "create_customer_inventory",
    "action_time": "2014-01-28 11:24:32",
    "user": "admin",
```

```
    "inventory": {
        "serial_number": "HRSC636029",
        "type": "NT",
        "status": "customer",
        "inventory_id": 21034416,
        "activity": {
            "date": "2014-01-28",
            "resource_id": "33003",
            "activity_id": 3954809,
            "appt_number": "#137165199",
            "customer_number": "019901104"
        },
        "changes": {
            "status": "customer",
            "type": "NT",
            "serial_number": "HRSC636029",
            "quantity": "1",
            "EQUIPMENT_ROOM_CODE": "test"
        }
    }
}
```


5.3.1.13 Request Creation

The following request operations are logged in the history and returned by the History API:

- create_customer_request
- create_inventory_request
- create_resource_request

Each request-related record contains the request identifier fields:

- type
- date
- resource_id

The records may optionally contain the "changes" field containing the custom request properties and their values changed for the request.

5.3.1.14 Request Creation Response Example

The example below shows the history record of a customer request creation:


```
{
    "operation": "create_customer_request",
    "action_time": "2014-02-05 12:04:24",
    "user": "admin",
    "request": {
        "type": "SR",
```

```
        "date": "2014-02-06",
        "resource_id": "routing",
        "activity": {
            "activity_id": 3952162,
            "appt_number": "#137167846",
            "customer_number": "019892755"
        },
        "changes": {
            "sr_body": "asfd"
        }
    }
}
```

The following example shows the history record of an inventory request creation:

```
{
    "operation": "create_inventory_request",
    "action_time": "2014-02-05 13:42:57",
    "user": "admin",
    "request": {
        "type": "SR",
        "date": "2014-02-05",
        "resource_id": "33015",
        "activity": {
            "activity_id": 3954828,
            "appt_number": "137165180",
            "customer_number": "019946338"
        },
        "inventory": {
            "serial_number": "7213125210",
            "type": "TV",
            "inventory_id": 20994113
        },
        "changes": {
            "sr_subject": "asdfd",
            "sr_body": "asdf"
        }
    }
}
```