# TRACKER REST API

v1.0 - 01/07/2016

# Table of contents

# Goal of the application

Guichet Entrepises uses numerous ids to refer to the same business creator. For example, Guichet Entreprises can attach an id beginning with “H” and another id beginning with “U” to the file of the business creator. Then, the file is automatically sent to a partner, who creates yet another id, and so on. Tracker allows several client applications, like Guichet Entreprises and its partners, to create links between all those ids. A partner just has to use this API to link its id to the provided one. It allows Guichet Entreprises to follow the workflow of how the file is being processed, and eventually make statistics on the overall files processing.

# Manipulated data

This service interface manipulates the **UIDs**, **messages** and **clients** data types.

An **UID** is either a random generated alphanumeric character string or a character string provided by a client. The format of generated **UIDs** is “YYYY-MM-XXX-XXX-CC”, where :

- **YYYY** is the current year
- **MM** is the current month
- **XXX-XXX** are random letters except ‘I’, ‘O’ and ‘U’
- **CC** is a checksum

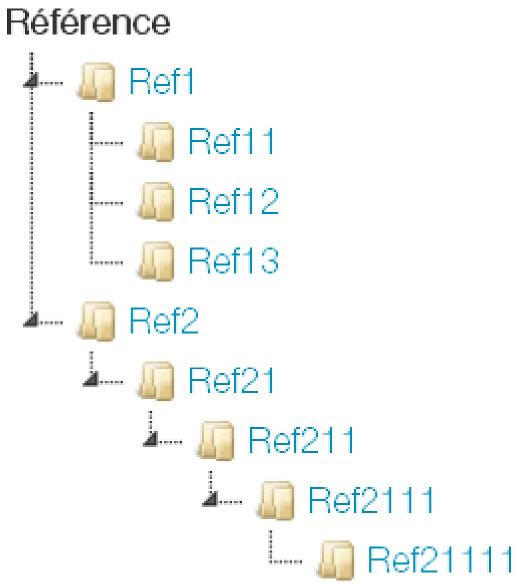
Checksum is obtained from *UID* first part and computed as `s[0]*31^(n-1) + s[1]*31^(n-2) + ... + s[n-1]` where s[i] is the ith character of the whole *UID* until the last ‘X’, n is the number of characters until the last ‘X’ in the *UID* pattern. Resulting value is reduced to 2 digits by combining digits of string representation of the previous formula.

A **message** is a short text describing a processing that just happened for a given **UID**.

A **client** is an application using Tracker. It contains the author name and may be used for security purposes. Each **UID** and **message** has an author.

# How it works

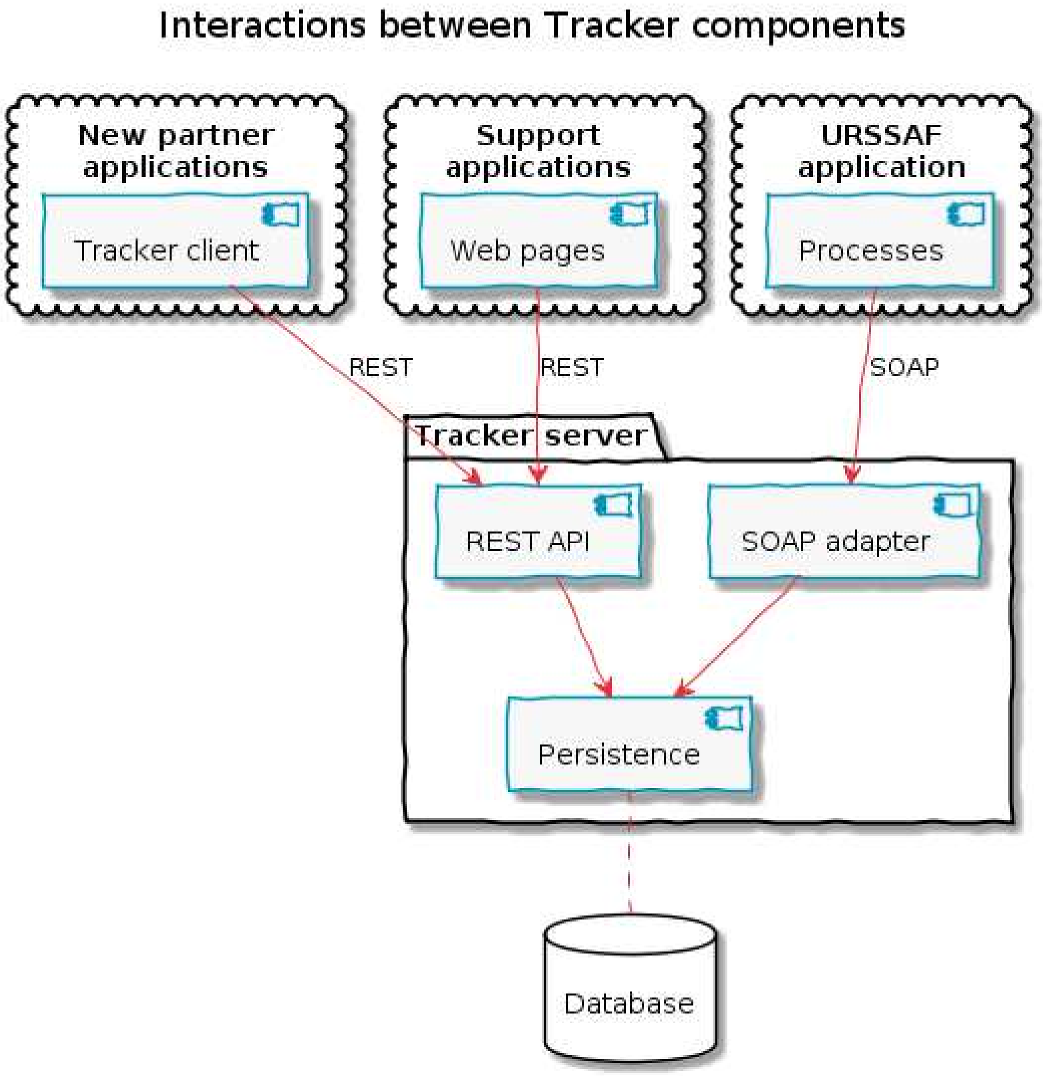
When an **UID** is linked to another **UID**, it is a **reference**. Therefore, a **reference** is an **UID** which has a parent. The terms **UID** and **reference** refer to the same concept (same data structure). A possible hierarchy of references would be :



A **message** can be posted on any **UID** or **reference**. Some technical **messages** are also automatically posted. It is then possible to find all **messages** posted on a hierarchy of **references**.

Whenever an **UID** or a **message** is created, the **client** name must be provided for traceability.

Each operation is performed calling the Tracker server. Java applications should use the Tracker client, which will call the Tracker server automatically :



# Errors handling

Errors are handled silently. It means that services always answer with the following HTTP codes :

- 2** when the service exists
- 4** when the service does not exist

Services may include functional error codes in their answers, as explained in each service description.

# Available methods

## HTTP protocol

All the following API methods must be invoked using HTTP methods : PUT, POST, GET, … To sum up :

- PUT is used to create data
- POST is used to update data
- GET is used to retrieve data

Further description : HTTP protocol (https://www.w3.org/Protocols/)

## Create a new UID

**Request** : `PUT /api/v1/uid?author={author}`

| Parameter | type | Description |
|---|---|---|
| author | String (query) | the identity of the entity sending this request |

**Response** :

- **Status 200** - application/json - UID has bean created

**Example**

```
Request : PUT http://tracker-server-url/api/v1/uid?author=Guichet Entreprises
Answer : 2016-06-NFK-NZE-30
```

**Behaviour** :

- If no **author** is provided :
  - **Status 400** is returned with message “Missing author”

# Retrieve an UID or create it

**Request** : `PUT /api/v1/uid/{reference}?author={author}`

| Parameter | type | Description |
|---|---|---|
| reference | String (path) | the UID or reference to retrieve or create |
| author | String (query) | the identity of the entity sending this request |

**Response** :

- **Status 200** - application/json - UID has bean created

**Example**

```
Request : PUT http://tracker-server-url/api/v1/uid/john_doe?author=Guichet Entreprises
Answer : 2016-06-NFK-NZE-30
```

**Behaviour** :

- If no **author** is provided :
  - **Status 400** is returned with message “Missing author”

# Link between references

**Request** : `POST /api/v1/uid/{previous ref}/ref/{new ref}?author={author}`

| Parameter | type | Description |
|---|---|---|
| previous ref | String (path) | the UID or reference to link to |
| new ref | String (path) | the UID or reference to link |
| author | String (query) | the identity of the entity sending this request |

**Response** :

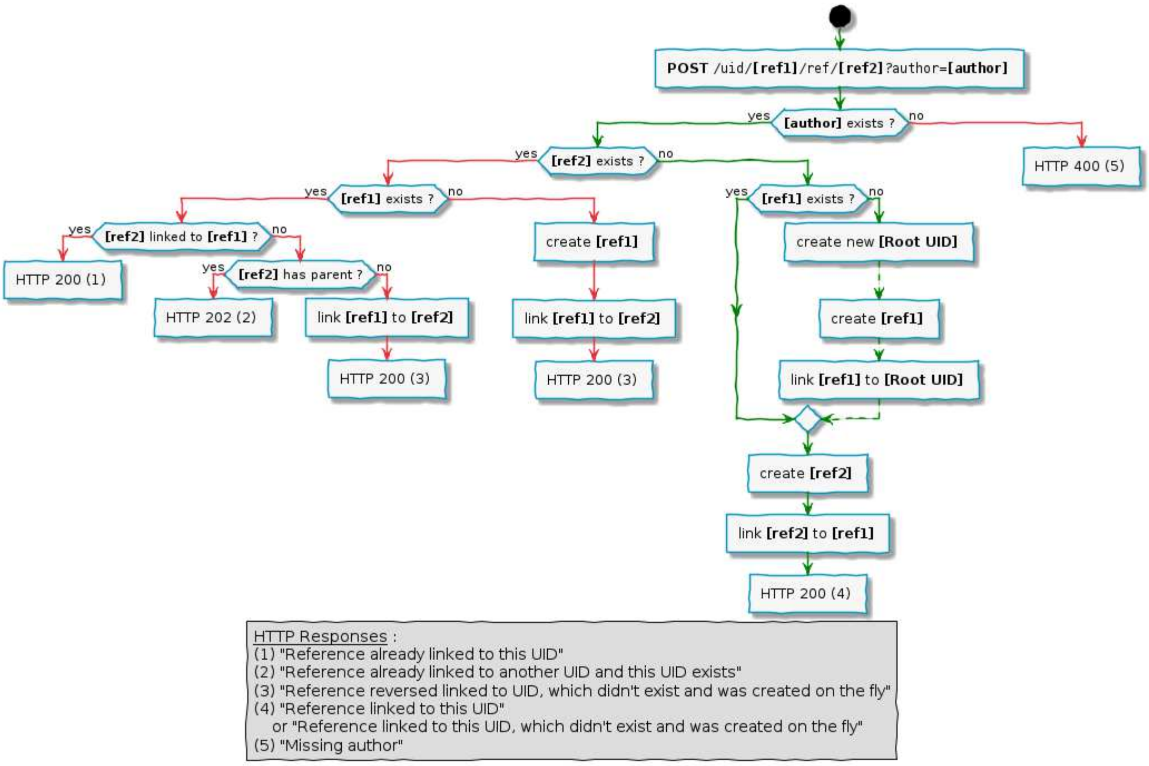
- **Status 204** - application/json - References are linked

**Examples**

```
Request : POST http://tracker-server-url/api/v1/uid/2016-06-NFK-NZE-30/ref/U78010019435?author=Guichet Entreprises
Answer : Reference linked to this UID

Request : POST http://tracker-server-url/api/v1/uid/U78010019435/ref/H12345678900?author=Guichet Entreprises
Answer : Reference linked to this UID

Request : POST http://tracker-server-url/api/v1/uid/U78010019435/ref/H12345678900?author=Guichet Entreprises
Answer : Reference already linked to this UID

Request : POST http://tracker-server-url/api/v1/uid/2016-06-NFK-NZE-30/ref/H12345678900?author=Guichet Entreprises
Answer : Reference already linked to another UID and this UID exists

Request : POST http://tracker-server-url/api/v1/uid/not existing UID/ref/H12345678900?author=Guichet Entreprises
Answer : Reference reversed linked to UID, which didn't exist and was created on the fly

Request : POST http://tracker-server-url/api/v1/uid/not existing UID bis/ref/new ref?author=Guichet Entreprises
Answer : Reference linked to this UID, which didn't exist and was created on the fly
```

**Behaviour** :

- If no **author** is provided :
  - **Status 400** is returned with message “Missing author” and the algorithm stops here
- If **ref** does not exist and **uid** does not exist :
  - An UID **newUid** is created
  - The reference **uid** is created and linked to the UID **newUid**
  - The reference **ref** is created and linked to the reference **uid**
  - **Status 200** is returned with message “Reference linked to this UID, which didn’t exist and was created on the fly”
- If **ref** does not exist and **uid** exists :
  - The reference **ref** is created and linked to the UID **uid**
  - **Status 200** is returned with message “Reference linked to this UID”
- If **ref** exists and **uid** does not exist :
  - Reverse link between **ref** and **uid**
  - **Status 200** is returned with message “Reference reversed linked to UID, which didn’t exist and was created on the fly”
- If **ref** exists and **uid** exists but **ref** is not linked to **uid** :
  - **Status 202** is returned with message “Reference already linked to another UID and this UID exists”
- If **ref** exists and **uid** exists and **ref** is linked to **uid** :
  - **Status 200** is returned with message “Reference already linked to this UID”



HTTP Responses :
(1) "Reference already linked to this UID"
(2) "Reference already linked to another UID and this UID exists"
(3) "Reference reversed linked to UID, which didn't exist and was created on the fly"
(4) "Reference linked to this UID"
or "Reference linked to this UID, which didn't exist and was created on the fly"
(5) "Missing author"

# Post a new message

**Request** : `POST /api/v1/uid/{ref}/msg?content={content}&author={author}`

| Parameter | type | Description |
|---|---|---|
| ref | String (path) | the UID or reference to link the message to |
| content | String (query) | content of the message |
| author | String (query) | the identity of the entity sending this request |

**Response** :

- **Status 204** - application/json - Message created and linked with specific UID reference.

**Example**

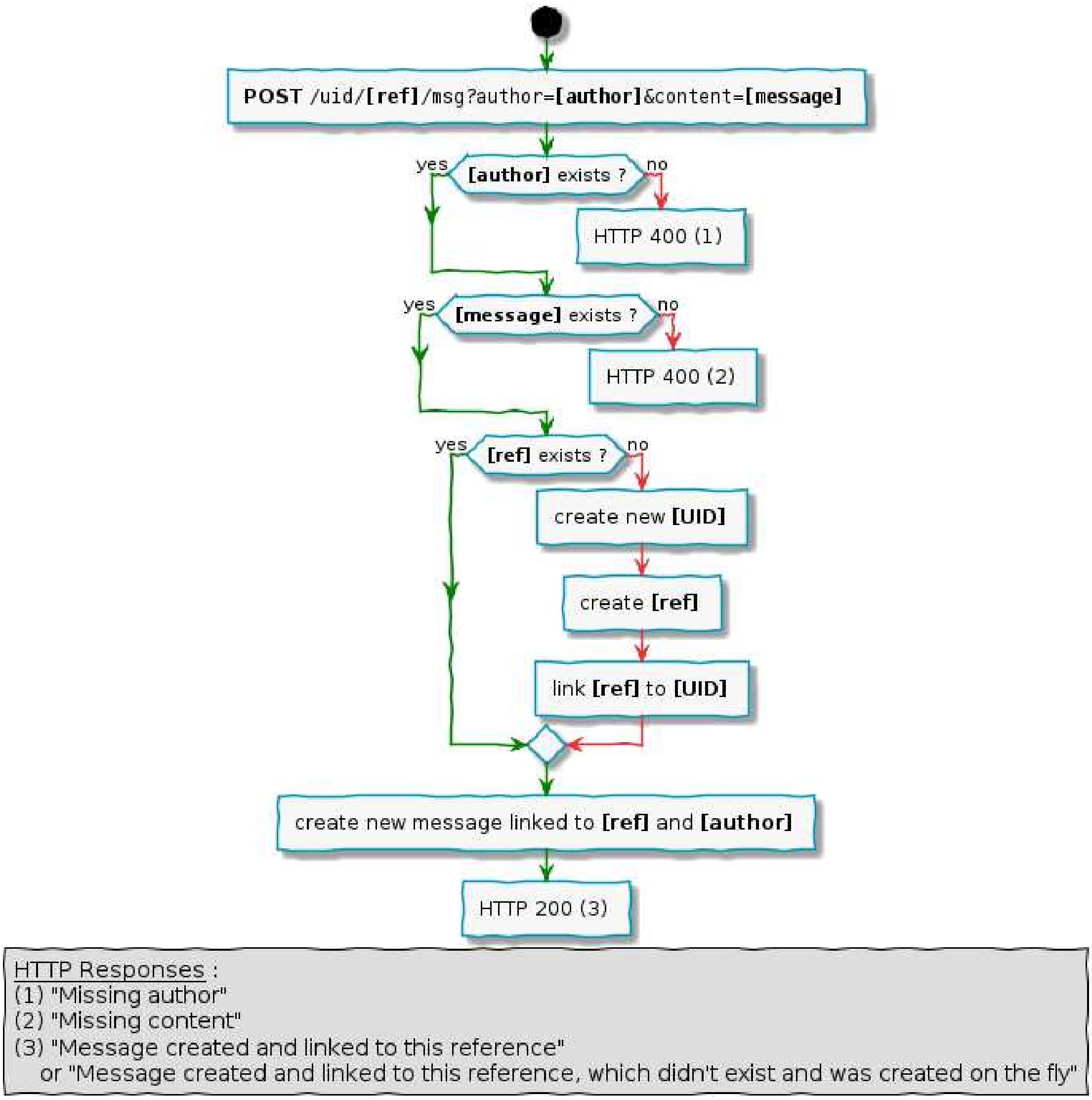
```
Request : POST http://tracker-server-url/api/v1/uid/U78010019435/msg?content=File sent to partner&author=Guichet Entreprises
Answer : Message created and linked to this reference

Request : POST http://tracker-server-url/api/v1/uid/not existing UID/msg?content=File sent to partner&author=Guichet Entreprises
Answer : Message created and linked to this reference, which didn't exist and was created on the fly
```

**Behaviour** :

- If no **content** is provided :
  - **Status 400** is returned with message “Missing content” and the algorithm stops here
- If no **author** is provided :
  - **Status 400** is returned with message “Missing author” and the algorithm stops here
- If **ref** does not exist :

- An UID **uid** is created
- The reference **ref** is created and linked to the UID **uid**
- A message is created and linked to the reference **ref**
- **Status 200** is returned with message “Message created and linked to this reference, which didn’t exist and was created on the fly”

- If **ref** exists :
  - A message is created and linked to the reference **ref**
  - **Status 200** is returned with message “Message created and linked to this reference”



HTTP Responses :
(1) "Missing author"
(2) "Missing content"
(3) "Message created and linked to this reference"
or "Message created and linked to this reference, which didn't exist and was created on the fly"

# Information for a specific reference

**Request** : `GET /api/v1/uid/{ref}`

| Parameter | type | Description |
|---|---|---|
| ref | String (path) | the UID or reference to retrieve information from |

**Response** :

- **Status 200** - application/json - Information about a reference.

**Behaviour** :

- If no content is found :
  - **Status 202** is returned with message “No matching data found”

### Example

```
Request : GET http://tracker-server-url/api/v1/uid/U78010019435
Answer :
{
    "value": "U78010019435",
    "created": 1463738071235,
    "client": {
        "created": 1463673051028,
        "name": "Guichet entreprises"
    },
    "parent": null,
    "references": [
        {
            "value": "2016-04-AAA-AAA-69",
            "created": 1463739523361,
            "client": {
                "created": 1463673051028,
                "name": "Guichet entreprises"
            },
            "parent": "2016-04-AAA-AAA-69",
            "references": [],
            "messages": []
        },
        {
            "value": "2016-04-BBB-BBB-42",
            "created": 1463739523362,
            "client": {
                "created": 1463673051028,
                "name": "Guichet entreprises"
            },
            "parent": "2016-04-AAA-AAA-69",
            "references": [],
            "messages": []
        }
    ],
    "messages": [
        {
            "created": "1463739545898",
            "content": "UID has been created"
            "client": {
                "created": 1463673051028,
                "name": "tracker"
            }
        },
        {
            "created": "1463739545899",
            "content": "Record step 1 has been validated",
            "client": {
                "created": 1463673051029,
                "name": "form-manager"
            }
        }
    ]
}
```

# Search existing UIDs

**Request** : `GET /api/v1/uid/search/data`

| Parameter | type | Description |
|---|---|---|
| columns[{i}][data] | String (query) | the column name to search on ({i} is an incremented number starting with 0) |
| columns[{i}][search][value] | String (query) | the column value to search for ({i} is an incremented number starting with 0) |
| draw | String (query) | the number of times the search has been called for the current page |
| length | String (query) | the length of the result set page for the pager |
| order[{i}][column] | String (query) | the index of the column to sort on (see above how column names are declared) |
| order[{i}][dir] | String (query) | the sort order (asc for ascending, desc for descending) |
| start | String (query) | the index of the result set page to display (starting with 0) for the pager |

**Response** :

- **Status 200** - application/json - The searched UID are returned.

**Behaviour** :

- Searches and returns the matching UIDs. The search criteria are cumulative.
- If search value is “empty-mode:true”
  - The search will be performed to check if the column which name was provided is empty.
- If search value is like “date-range-mode:DD/MM/YYYY:DD/MM/YYYY”
  - The search will be performed to check if the column which name was provided is a date between the 2 provided dates
- In other cases
  - The search will be performed to check if the column which name was provided contains the provided string

**Example**

```
Request : GET http://tracker-server-url/api/v1/uid/search/data
&columns[0][data]=value   &columns[0][search][value]=48
&columns[1][data]=parent  &columns[1][search][value]=empty-mode:true
&columns[2][data]=created &columns[2][search][value]=date-range-mode:10/06/2016:06
/07/2016
&draw=6                   &length=10
&order[0][column]=2       &order[0][dir]=desc
&start=0

Answer :
{
    "draw": 6,
    "recordsTotal": 2,
    "recordsFiltered": 2,
    "data": [
        {
            "value": "2016-07-KCH-ZNY-48",
            "client": "Guichet Entreprises",
            "created": 1467619509662,
            "references": [],
            "parent": "",
            "messages": null
        },
        {
            "value": "2016-06-BKP-CCX-48",
            "client": "Guichet Entreprises",
            "created": 1467619509762,
            "references": [],
            "parent": "",
            "messages": null
        }
    ]
}
```