# Augmented Cash API interface guide



**Version: 1.1**

**Creation Date: 23rd August 2023 - Updated the 14th April 2024**

# Version Control

| Date | Version | Author | Description |
|---|---|---|---|
| 23/08/2023 | 1.0 | Product - Elodie Ravachol | Initial Version |
| 14/08/2023 | 1.1 | Product - Elodie Ravachol | 'Make REST API Requests / Get item details' section added |

# Contents

# Preamble

Welcome to the Sidetrade APIs, your gateway to seamless integration with our cutting-edge Order-to-Cash platform.

Sidetrade empowers client applications to make authenticated requests to our APIs on their own behalf, enabling seamless integration without the need for individual end-user authentication.

Our authorization server adheres to the Client Credentials Grant flow of the OAuth 2.0 specification, ensuring a secure and consistent authentication mechanism.

English serves as the official language for all Sidetrade APIs, including this comprehensive documentation and all object and property descriptions. This ensures clarity and consistency across the entire platform.

# Overview

The Sidetrade Augmented Cash API is a set of endpoints that allow you to programmatically update the status of invoices from your own applications. The API is designed to be used by companies that use Sidetrade Augmented Cash to manage their collections.

The API provides four main endpoints:

- **Bulk Update status**: This endpoint allows you to update the status of a list of invoices at the same time. The API will return an ID for the created task, which you can use to track the status of the update.
- **Get task status**: This endpoint allows you to get the status of a task created by the Bulk Update status endpoint. The status can be one of the following:
    - Queued: The task is waiting to be processed.
    - In progress: The task is being processed.
    - Partially successful: The task has been completed, but some rows have been rejected.
    - Failed: The task has failed.
    - Successful: The task has been completed successfully.
- **Get task status with details**: This endpoint allows you to get more detailed information about a task created by the Bulk Update status endpoint. This information includes the status of each invoice in the task, as well as any errors that occurred.
- **Get items details**: This endpoint that can be integrated in your application to pull the information of invoices. It allows users to access the updated information of invoices, including their qualification status and any changes that have been made to their status. This API can be used to keep track of the status of invoices and to ensure that they are being processed correctly.

The Sidetrade Augmented Cash API is a powerful tool that can help you automate your collections process. By using the API, you can save time and improve the accuracy of your collections efforts.

Here are some of the benefits of using the Sidetrade Augmented Cash API:

- **Increased efficiency**: You can automate the process of updating invoice statuses, which can save you time and resources.
- **Improved accuracy**: The API can help you to ensure that invoice statuses are updated correctly, which can help you to avoid errors.
- **Enhanced visibility**: You can track the status of your invoice updates in real time, which can give you greater visibility into your collections process.

If you are a company that uses Sidetrade Augmented Cash to manage your collections, I recommend that you consider using the Sidetrade Augmented Cash API to automate your collections process. The API is a powerful tool that can help you save time, improve accuracy, and enhance visibility.

**How it works**

Follow the described process to make authenticated API calls.

- Authentication: Get Client ID and Client Secret and use them to generate an Access Token
- Make REST API Requests

# Authentication

## Step 1: Get Client ID and Client Secret

Each application is assigned a unique Client ID (also known as Consumer key or API key) and Client Secret. Make note of these values as they have to be integrated into the configuration files or the actual code of your application.

Please contact Sidetrade Customer Care to get your Client ID and Client Secret.

⚠ Your *Client Secret* protects your application's security so be sure to keep it secure! Do not share your *Client Secret* value with anyone, including posting it in support forums for help with your application.

## Step 2: Generate an Access Token

To generate an access token, issue a HTTP POST against **accessToken** with both your Client ID and Client Secret values.

### Resource URL

```
https://login.sidetrade.net/connect/token
```

### Resource Information

| | |
|---|---|
| Response formats | JSON |
| Requires authentication? | No |
| Rate limited? | No |

### Parameters

| Parameter Name | Description | Required |
|---|---|---|
| grant_type | The value of this field should always be client_credentials. | Yes |
| client_id | The Client ID value generated by the Customer Care. | Yes |
| client_secret | The Client Secret value generated by the Customer Care. | Yes |

## Sample Request

```
POST https://login.sidetrade.net/connect/token HTTP/1.1
Host: www.sidetrade.net
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=client_credentials&client_id={your_client_id}&clie
nt_secret={your_client_secret}
```

## Sample Response

```
{
    "access_token": "eyDjejsgdbGciOiJSUzI1XdXXdXddXXXXfXXdXX",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope": "ALL"
}
```

`expires_in` value in seconds

### Possible error messages

`"error": "invalid_client"`: Check if your client id and client secret are filled correctly.

This message will also show up when the client_secret contains invalid URL characters such as the + (plus) sign. For example,

"CzbYUTS/AGiysc59cpIjoiMTk2RUJf2Ba23+bTv6knV2tgz=" will trigger the error.

```
C:\Users\kkiiza>curl -X POST "https://cloud-ptf4-web-oauth.sidetrade.com/connect/token" -H "Content-Type: application/x-
www-form-urlencoded" -d "grant_type=client_credentials&client_id=84b6efb1-2ad1-4e2e-9750-dfb672e304b6&client_secret=nupC
zbYUTS/AGiysc59cpIjoiMTk2RUJf2Ba23+bTv6knV2tgz="
{"error":"invalid_client"}
C:\Users\kkiiza>
```

`"error": "unsupported_grant_type"`: Check if the grant_type is client_credentials

# Make REST API Requests

Once you've received an access token, you can make API requests by including an Authorization header with your token in the HTTP call to Sidetrade's API.

Augmented cash collection API provide 3 endpoints:

1. **Bulk Update status**

Qualify a list of invoices and return the ID of the created qualification task.

⚠ The functional rules for this update are strictly identical as from Sidetrade Augmented Cash platform. So only items that can be qualified from the platform are eligible to this update. It means they have to respect the below conditions to not be rejected:

- Item is still open (Remainder != 0)
- Item type is Credit Note (AVO) or Invoice (FAC)
- Item is set as "collectable" in the daily import

2. **Get task status**

Get the invoice qualification task status.

3. **Get task status with details**

Get the invoice qualification task details.

4. Get item details

# 1. Bulk Update status

### Resource URL

```
https://api.sidetrade.net/collection
```

### Resource Information

| | |
|---|---|
| Response formats | JSON |
| Requires authentication | YES |
| Rate limited | YES : One call per second |

### Sample Request

```
POST v1/invoices/status/bulk HTTP/1.1
Host: api.sidetrade.net
Connection: Keep-Alive
Cache-Control: no-cache
Content-Type: application/json
Authorization: Bearer {access_token}

Body:
[
  {
          "InvoiceNumber": "18222222222",
          "CompanyCode": "123333",
          "Substatus": "DLPP",
          "PaymentDate": "2022-03-15",
          "Comment": "paid on time",
          "ResolverCode": ""
  },
  {
          "InvoiceNumber": "43434343443",
          "CompanyCode": "8787878",
```

```
            "Substatus": "DLPP",
            "PaymentDate": "2022-03-15",
            "Comment": "paid on time",
            "ResolverCode": ""
    }
]
```

## Sample Response

You can use this value to get the task status or task details in the next GET end points.

```
653
```

Having this value returned in the response only means that the task to qualify the invoices has been created. You need to make the get task status or get task details request to confirm whether the qualification is successful or not.

## Body Properties

| Parameter Name | Description | Type | Required |
|---|---|---|---|
| invoiceNumber | Functional invoice number of the invoice. | string | YES |
| substatus | sub status code* | string | YES |
| comment | Comment about the qualification | string | NO |
| paymentDate | Promise of Payment Date | string<br><br>ISO 8601 | Yes if the sub status is a type of Promise of payment |
| companyCode | Functional code of the company | string | YES |
| resolverCode | Resolver code in the case of the Litigation** | string | YES if the sub status is a type of Litigation |

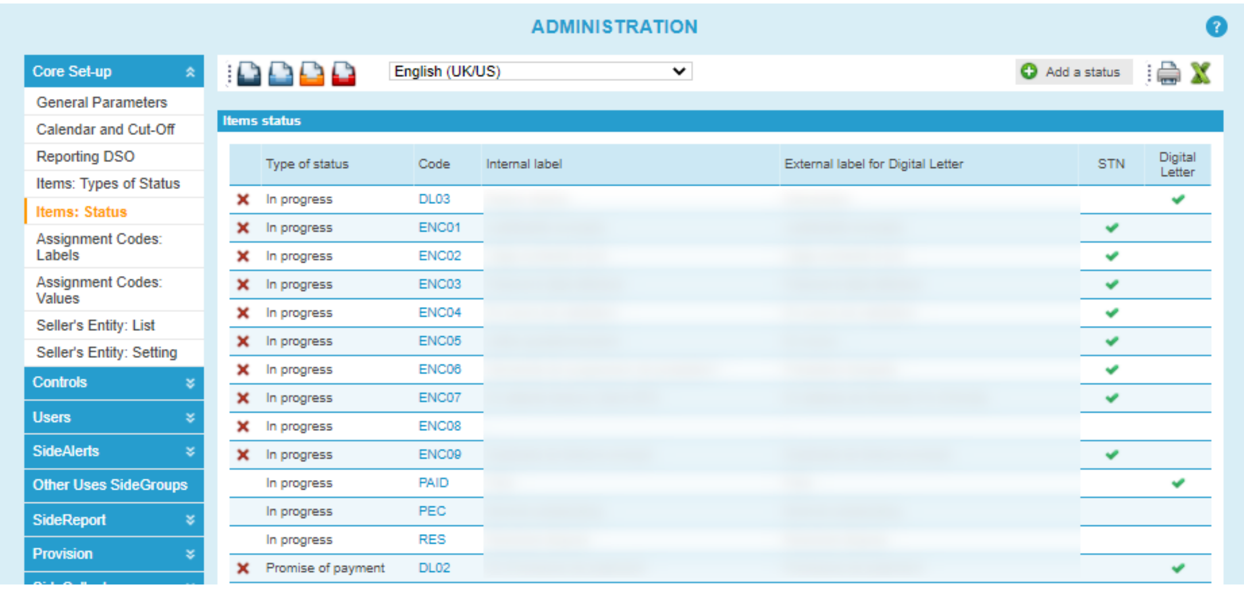
> ⚠️ If some items (invoices and/or credit note) are updated with the exact same information : same sub-status code, same comment, same payment date, same company code, same resolver code, they are managed together.
>
> It means that if an automated actions is set to be triggered for this sub-status, only one action will be triggered for the group of items.
>
> If at least one information is different between 2 items, 2 separate actions will be triggered and visible in the collector to-do-list.

### Where to find the sub-status and the resolver code?

- **Sub-status code**

To get the list of sub status code please connect to the platform with an administrator profile and access to the **Settings/Administration/Items: Status** section.

From there you can export the sub-status list and get the mapping between the sub-status code and the functional label in any user language:

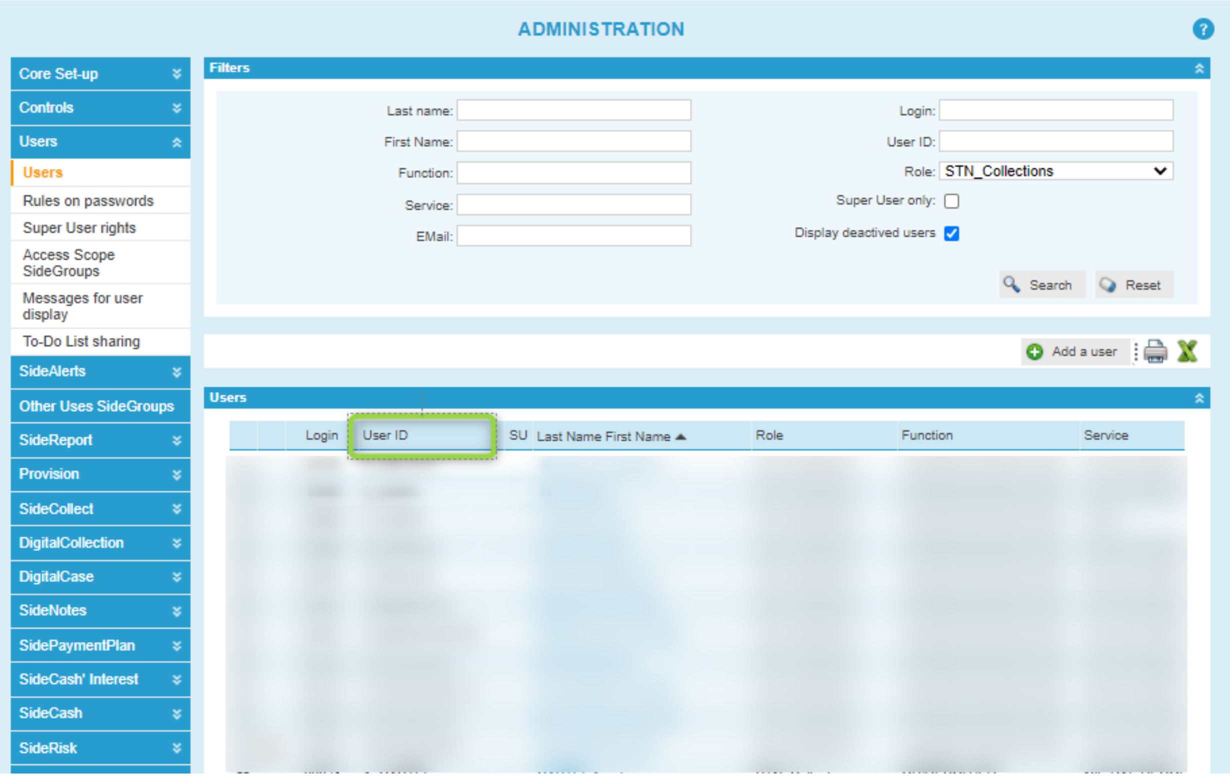



- **Resolver code**

The resolver code is usually the employee ID used in your company.

In order to get your Resolver code, please connect to the platform with an administrator profile and access to the **Settings/Users/Users** section.

From there you can export the list of users and get User ID set for each user in the platform:



# 2. Get task status

### Resource URL

```
https://api.sidetrade.net/collection
```

### Resource Information

| | |
|---|---|
| Response formats | JSON |
| Requires authentication? | YES |
| Rate limited? | YES : One call per second |

### Parameters

| Parameter Name | Description | Required |
|---|---|---|
| Task id | The id of the task created by bulk update endpoint | Yes |

### Sample Request

```
GET v1/invoices/task/{taskid}  HTTP/1.1
Host: api.sidetrade.net
Connection: Keep-Alive
Cache-Control: no-cache
Content-Type: application/json
Authorization: Bearer {access_token}
```

### Sample Response

```
Succeeded
```

| Status | Description |
|---|---|
| Queued | The task is queued, waiting for the previous one to be completed |
| In progress | The task is still in progress |

| | |
|---|---|
| Partially successful | The task has been completed, but errors have been found and at least one row has been rejected. Note that even if some errors have been found for all the rows, it will be partially successful, once the API was able to read the data that were sent and analyze what the issue is for each of them.<br><br>The reason of the error can be obtain through the *get task status with details* end point<br><br>For a quick correction, it is also possible to access to the platform, to the Status import section, download the csv file that contains the row in errors and the error description. It is then possible to correct it and manually re-import the file. |
| Failed | The update is totally failed due to a technical error, the API was not able to read the data.<br><br>The reason of the error can be obtain through the *get task status with details* end point |
| Successful | The update is totally successful. All items have been updated as expected |

# 3. Get task status with details

## Resource URL

```
https://api.sidetrade.net/collection
```

## Resource Information

| | |
|---|---|
| Response formats | JSON |
| Requires authentication? | YES |
| Rate limited | YES : One call per second |

## Parameters

| Parameter Name | Description | Required |
|---|---|---|
| Task id | The id of the task created by bulk update endpoint | Yes |

## Sample Request

```
GET v1/invoices/task/{taskid}/details  HTTP/1.1
Host: api.sidetrade.net
Connection: Keep-Alive
Cache-Control: no-cache
Content-Type: application/json
Authorization: Bearer {access_token}
```

## Sample Response


```
{
    "taskId": 653,
    "status": "Succeeded",
    "startDate": "2021-02-11T10:37:48.2685050",
    "endDate": "2021-02-11T10:44:53.1633030",
    "message": "",
    "totalInvoices": 2,
    "processedInvoices": 2,
    "invoices": [
        {
            "taskId": 613,
            "invoiceNumber": "R82CCC85XXX",
            "substatus": "DLPP",
            "comment": "paid on time",
            "paymentDate": "2022-03-15",
            "companyCode": "1232567",
            "companyName": "MyComapny",
            "status": "Succeeded",
            "errors": []
        },
        {
            "taskId": 653,
            "invoiceNumber": "H82BBB224YYY",
            "substatus": "DLPP",
            "comment": "promised to pay",
            "paymentDate": "2022-03-15",
            "companyCode": "1232567",
            "companyName": "MyCompany",
            "status": "Succeeded",
            "errors": []
        }
    ]
}
```

## Response Properties

### Task

Task is an element created when you call the Bulk update end point.

| Parameter Name | Description |
|---|---|
| taskId | Id of the task |
| status | Partially successful<br>Queued<br>Failed<br>Successful |
| startDate | Start date and time of the task ISO 8601 |
| endDate | End date and time of the task ISO 8601 |
| message*(listed below) | Filled if the status is Failedor PartiallySucceeded |
| totalInvoices | Total number of invoices in the task |
| processedInvoices | Number of successfully updated invoices |
| invoices | List of invoices in the task ( Detailed below ) |

### Invoice

| Parameter Name | Description |
|---|---|
| taskId | Task Id |
| invoiceNumber | Functional item number |
| substatus | sub status code |
| comment | Comment about the qualification |
| paymentDate | Promise of Payment Date ISO 8601 |
| companyCode | Functional code of the company |
| companyName | Company Name |

| | |
|---|---|
| status | Update result status of the invoice : Failed or Succeeded |
| errors*(listed below) | List of error messages |

*Possible Functional Errors or messages :

| |
|---|
| The item has not been found on this company or its update is not allowed. The stat us can only be updatedfor open invoices and credit notes that have been set as coll ectable on the daily import. |
| Company code, invoice number and Sub status are mandatory fields |
| The sub status code does not exist. |
| The payment date is mandatory for a promise of payment qualification. |
| The payment date format is incorrect. Be sure to choose a format that matches your file format when importing the data. |
| The payment date can not be in the past. |
| The dispute resolver internal code does not exist. |
| Several open items with the same number have been found on the company. Only manual qualification is allowed in this situation. |
| The invoice status has not been updated due to a technical error. Please update the item manually and contact the Sidetrade Support Team. |

# 4. Get item details

## Resource URL

```
https://api.sidetrade.net/items
```

## Resource Information

| | |
|---|---|
| Response formats | JSON |
| Requires authentication? | YES |
| Rate limited | YES : 100 calls per second |

## Request Body Properties

| Parameter Name | Description | Type | Required |
|---|---|---|---|
| ItemNumbers | List of Functional item numbers | list (limited to 1000) | YES |
| AccountNumber | Functional code of the account (buyer) | string | No |

## Sample Request

```
POST v1/Item/list  HTTP/1.1
Host: api.sidetrade.net
Connection: Keep-Alive
Cache-Control: no-cache
Content-Type: application/json
Authorization: Bearer {access_token}
X-CULTURE : {culture code}*
Body :
{
    "ItemRequest": {
        "AccountNumber": "string",
```

```
        "ItemNumbers": [
            "string"
        ]
    }
}
```

Ex:

```
curl --location 'https://cloud-ptf4-api.sidetrade.com/gateway_api/items/v1/list' \
--header 'Content-Type: application/json' \
--header 'X-CULTURE: fr-FR' \
--header 'Authorization: Bearer Ha65VET5YoQzfpeRSi3ZlXQDL8agZsrmo912R805A' \
--data '[
"9826276"
]'
```

Culture code is optional, if empty, by default the en-GB will be used.

## Response Properties

| Parameter Name | Description | Type |
|---|---|---|
| itemNumber | Functional invoice number of the invoice. | string |
| accountNumber | Functional code of the company | string |
| accountName | Company Name | string |
| itemTypeDescription<br>*( displayed in UK English if no X-Culture provided)* | Credit Note (AVO)<br>Installment (ECH)<br>Invoice (FAC)<br>Miscellaneous operation (OD)<br>Payment (RGL)<br>Draft (TRT) | string |

| | | |
|---|---|---|
| itemSubTypeDescription<br>*( displayed in UK English if no X-Culture provided)* | Ordinary invoices (ORD)<br>Deposits (ACO)<br>Bad debt invoices (DTX)<br>Unpaid invoices (IMP)<br>Late Payment Interests (INR) | string |
| itemStatusDescription*<br><br>*( displayed in UK English if noX-Culture provided)* | Status of the item<br><br>Ex:<br>Account Update<br>Below Threshold<br>Broken Promise to Pay<br>Call Back | string |
| itemStatusTypeDescription**<br><br>*( displayed in UK English if no X-Culture provided)* | Category/type of item status<br>In progress (EC)<br>Dispute (LI)<br>Promise of Payment (PV)<br>Litigation (CO) | string |
| promiseOfPaymentDate | Promise of payment Date | string<br><br>ISO 8601 |
| resolver*** | Resolver user ID for Dispute and Litigation | string |
| statusUpdatedDateTime | Status Modification Date | string<br><br>ISO 8601 |
| statusUpdatedComment | Comment about the qualification | string |

## Sample Response

```
[
  {
        "itemNumber": "9826276",
        "accountNumber": "968178",
        "accountName": "Marechal Bourgeois S.A.S.",
        "itemTypeDescription": "Facture (FAC)",
        "itemSubTypeDescription": "Ordinaires",
        "itemStatusDescription": "promesse de règlement",
        "itemStatusTypeDescription": "Promesse de règlement",
```

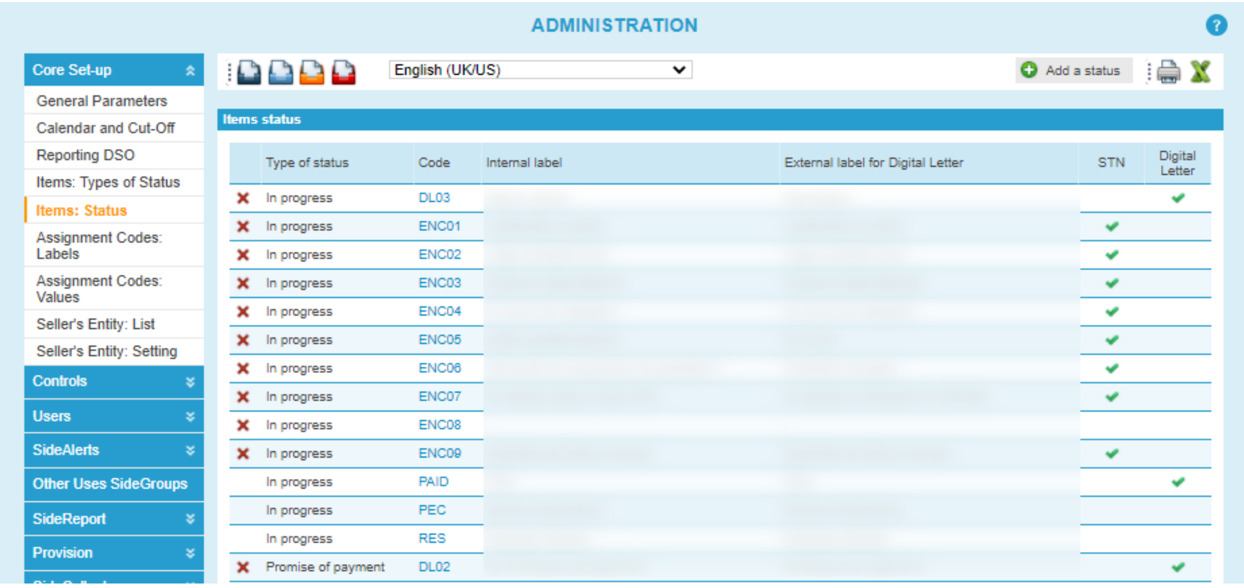
```
        "promiseOfPaymentDate": "2007-03-03",
        "resolver": " Gregoire",
        "statusUpdatedDateTime": "2007-02-13"
    }
]
```

**Where to find the code correspondance?**

- **itemStatusDescription : Sub-status code**

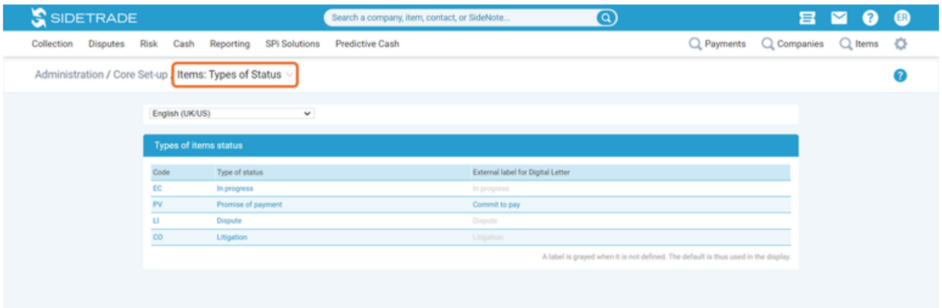
To get the list of sub status code please connect to the platform with an administrator profile and access to the **Settings/Administration/Items: Status** section.

From there you can export the sub-status list and get the mapping between the sub-status code and the functional label in any user language:




- **Item type**

To get the information about available types, please connect to the platform with an administrator profile and access to the **Settings/Administration/Items: Type of Status** section.

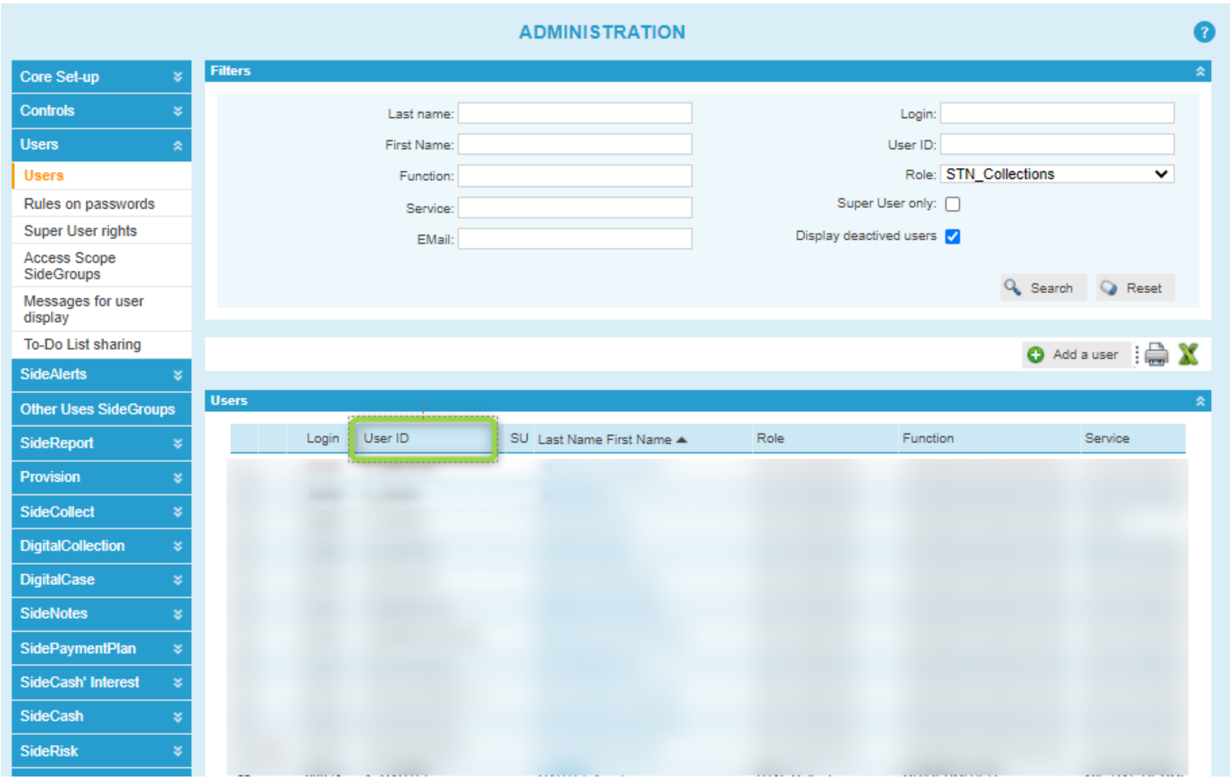



- **Resolver : User ID**

The resolver code is usually the employee ID used in your company.

In order to get your Resolver code, please connect to the platform with an administrator profile and access to the **Settings/Users/Users** section.

From there you can export the list of users and get User ID set for each user in the platform:



# Responses Codes

Sidetrade API uses standard HTTP response codes.

## Successful execution

200 : Successful request and response.

## Error Handling

There are two status code ranges you can expect:

- 4xx error status codes indicate an error due to a problem on the client's part because of the information provided (e.g., invalid input)
- 5xx error status codes indicate an error due to a problem in the server side.

Ex: 404 Not Found for the end point */collection/v1/invoices/task/{taskID}/details* if the data are not found for the given task ID

### Invalid Tokens

If you make an API call using an invalid token, you'll receive a 401 Unauthorized response from the server. In this case, the token may need to be regenerated because it expired or was revoked.

These are not the only reasons for an invalid token. Make sure your applications are coded to properly handle a 401 Unauthorized error.

# Integration into Sidetrade

⚠ This section only applies to 'Bulk update Status', 'Get task Status', and 'Get task status with details'

All updates done via the API are visible in the Status Import screen with their current status. If your data is partially successful a csv file that contains the items that have been rejected can be downloaded in order to be manually corrected, then re-imported.

It is possible from the platform to set a notification email address in order to receive a confirmation email each time an update via the API is done, with the corresponding status (Successful, Partially Successful or Failed)

An important point in terms of integration is to make sure these updates via API fit with your business process. So we recommend to verify with the platform administrator that all automated actions are well set as expected.

## Status Import screen

By default, this feature can be access only by administrator and Credit Manager. It allow you to test it and decide how you want to organize the deployment with your teams.

It is then possible to request its deactivation for a role, or its activation for collectors and/or sales managers profiles, by contacting your Sidetrade main contact.

From that screen that you can access via the **Collections/Actions/Status import** menu, you will find, in addition of the import manually handled by users, the updates done via API.

Each line contains :

- The import date
- The source (it can be an external source using the API or a csv file for manual import)
- The Author (Sidetrade Engine for import done using the API)
- The number of rows imported on the total number of rows that were expected to be imported. This information is only displayed for **successful** or **partially successful** imports
- The status (queued, in progress, successful, partially successful or failed)
- The link to the csv that contains the rows in errors and the reason of the error. This file is available only when the import is **partially successful**

## How to correct a file in a partially successful status from the status import screen?

In case you have at least one row in error, it is possible to manually handle it from this page. You will find a link to download a csv when you import has been partially successful.

This csv contains only the rows that have been rejected with the reason of the error. From this file you can:

1. Correct the errors
2. Delete the "reason of the error" column
3. Manually import the file from the import wizard

# Administrator - How to prepare the activation of the feature

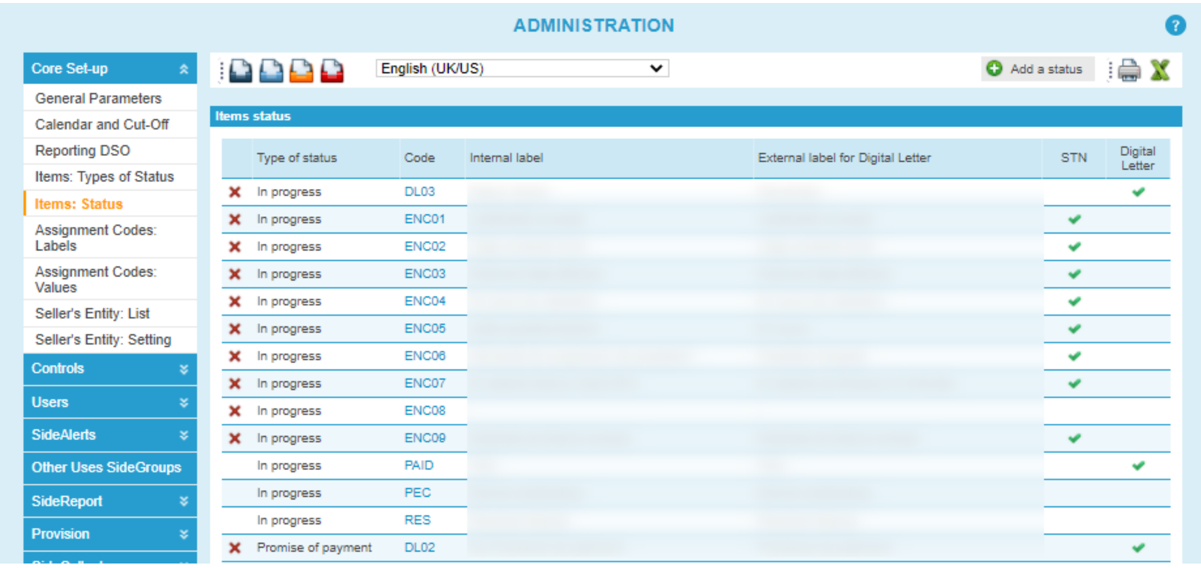
> ⚠ This section only applies to 'Bulk update Status', 'Get task Status', and 'Get task status with details'

## Step 1 - Define the list of sub-status to be used

The administrator can define the list of sub-status he wants to use for the import from the **Settings/Core set-up/Items: Status** section.

From this section, the administrator has two options :

1. Identify the list of sub-status code he wants to use and make sure they are the ones that will be sent to the API
2. Create new sub-status specific to the import via API, with a code that matches the one sent to the API and a label explicit for the end-user.




> ⚠ When creating the sub-status, you have to indicate if it can be used to qualify items from the platform and/or from a Digital letter. If none of these 2 options are selected, the sub-status will only be available for the items status import.

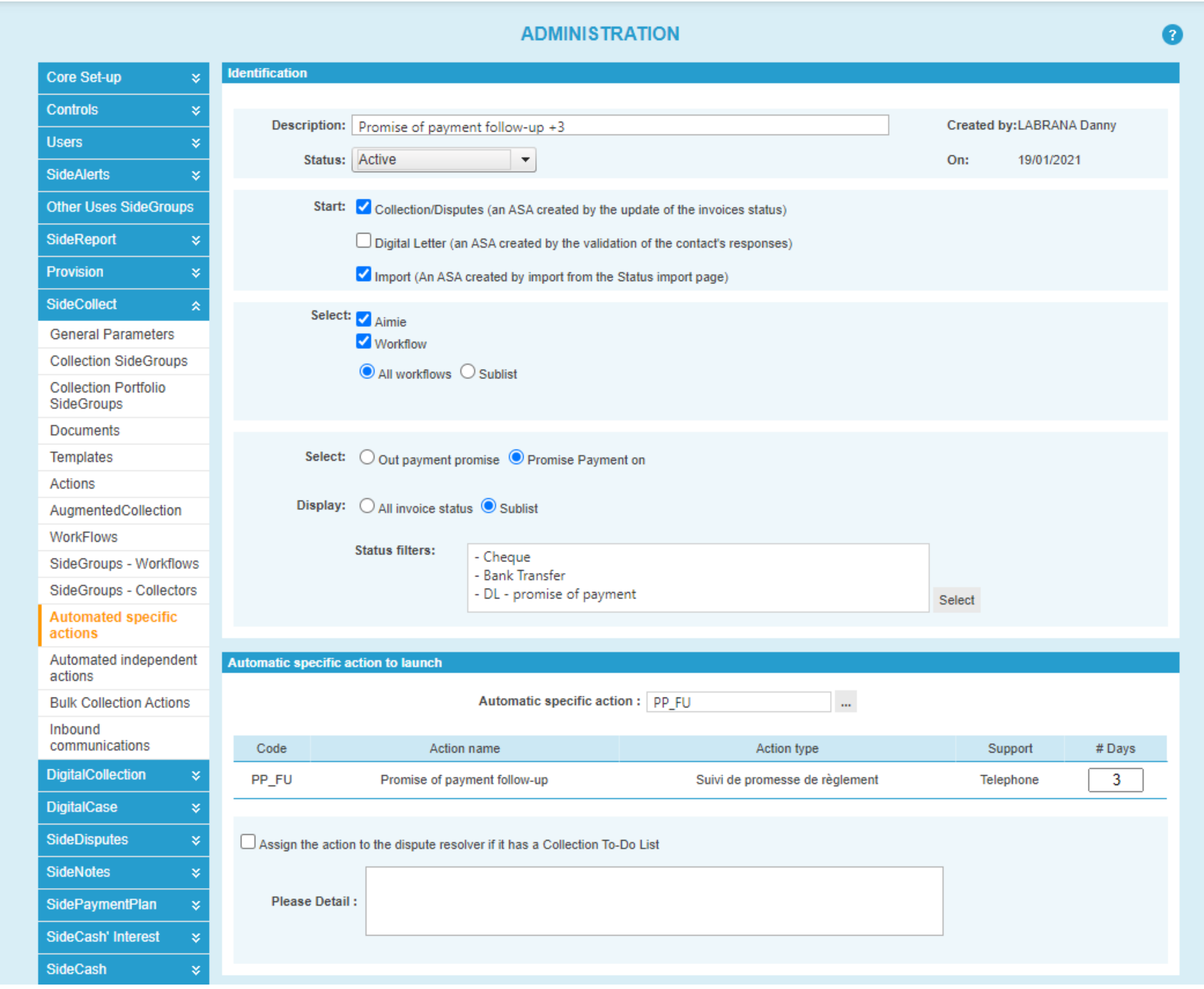
## Step 2 - Set the automated actions

To fit with your business processes, you will probably have to set some specific actions to be triggered on the collector or on the dispute resolver to-do-list depending on the imported sub-status.

You have the option to set :

- **Automated specific Actions** that will exclude the invoice(s) from the workflow. It means the invoice(s) can't be the leading invoice anymore and won't trigger any workflow or Aimie actions as the action is not completed or cancelled. They are set from the **Settings/Sidecollect/ Automated Specific Actions** section.

- **Automated independent actions** that will not exclude the invoice(s) from the workflow. It means the invoice(s) can still be the leading invoice and trigger some workflow or Aimie actions even if the action is not completed or cancelled. They are set from the **Settings/Sidecollect/ Automated Independant Actions** section.




From these 2 sections, it's possible to:

- Use an already existing automated action set for update via the company page or via the Digital Letter, to follow the same process if the status is updated via import

- Create a new automated action, specific to import, that will start only if the qualification is done via the import

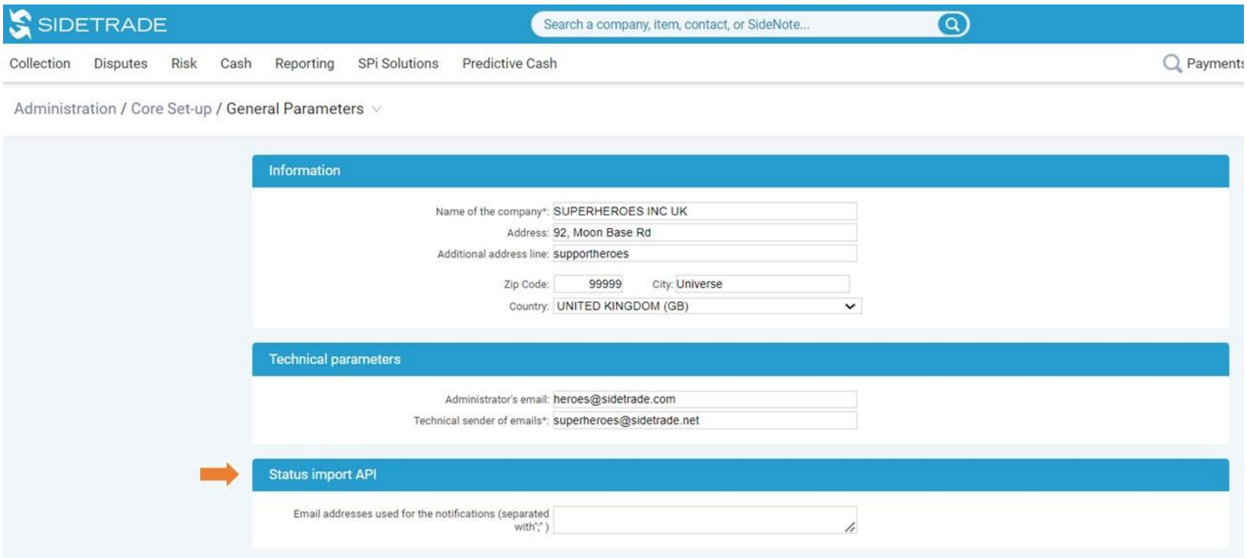
Actions will be triggered by default in the company's collector to-do-list, except if you indicate for and automated specific actions, that it has to be assigned to the dispute resolver, if a dispute resolver has been defined in the import and he has a collection to-do-list.

## Step 3 - Set the notification emails

From this section, you can define who should receive a notification email for a status update via API, and for which status you want to receive this notification (Success, Failed and/or Partially success)

This can be modified at any time.

To do this setting, please connect to the platform with an administrator profile and access to the **Settings/Core set-up/General Parameters** section.




Only one email address can be used for the notification, so if several persons need to be notified, we invite you to use a group email address.

XXXXXXXXXXXXXXXXXXXXX END OF DOCUMENT XXXXXXXXXXXXXXXXXXX