

# Agicap

## API User Guide - Bank Journal

---

### Introduction

This guide explains how to use the Agicap Treasury Bank Journal API to export your bank accounting entries.

This API allows you to:

- List all the history of exports made on Agicap (GET)
- View the details of a specific export (GET)
- Start a new export of entries (POST)

Access link: <https://api.agicap.com/api-details/treasury-bank-journal-v1>

---

<b>1. Retrieving credentials</b>	<b>2</b>
<b>2. API Authentication</b>	<b>2</b>
<b>3. Retrieving your entities' identifiers</b>	<b>3</b>
<b>4. Using the Treasury Bank Journal API</b>	<b>4</b>
4.1 List previous exports (GET /public/treasury-bank-journal/v1/entities/{entityId}/exports)	5
4.2 Obtain a specific export (GET /public/treasury-bank-journal/v1/entities/{entityId}/exports/{exportId})	6
4.3 Create a new export (POST /public/treasury-bank-journal/v1/entities/{entityId}/exports/{exportId})	7
<b>5. Practical advice</b>	<b>8</b>
Understanding the counters for the first export	8
Pagination management	8

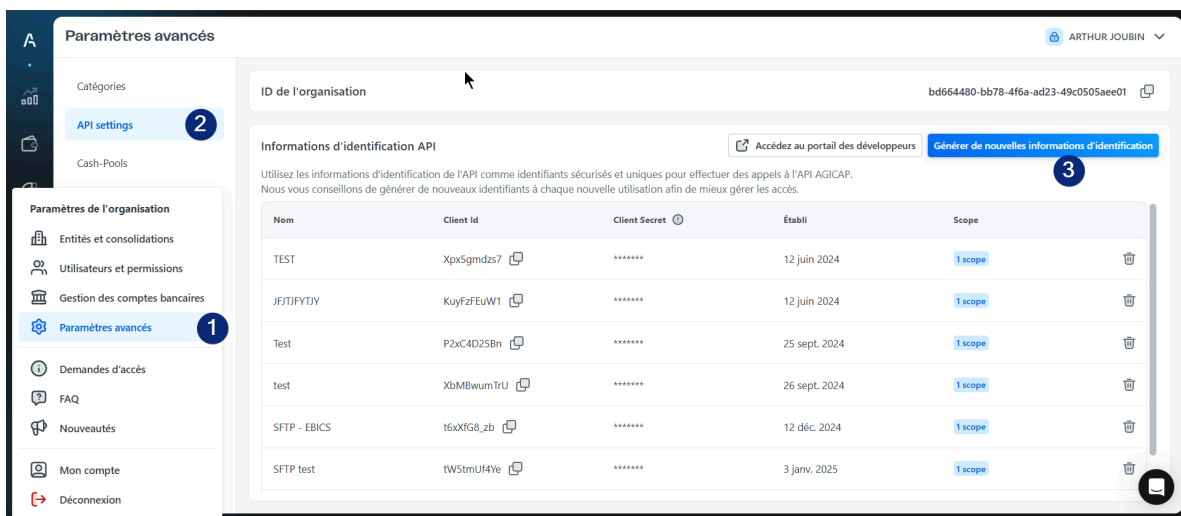
# Agicap

## API User Guide - Bank Journal

### 1. Retrieving credentials

Before you begin, you need to obtain your API credentials from the Agicap interface:

1. Log in to your Agicap account as an administrator at this URL: <https://app.agicap.com/fr/app/organization-advanced-settings/public-api/clients>
2. Create a new API key (if you don't already have one)
3. Carefully note your **Client ID** and your **Client Secret**



### 2. API Authentication

Before making API calls, you must authenticate to obtain an access token.

Once you have obtained the token, you must include it in all your API calls in the Authorization header.

The token has a limited validity period (usually a few hours). Check the expires\_in value in the authentication response and renew the token before it expires.

```
Python
import requests

# Configuration
BASE_URL = "https://api.agicap.com"
AUTH_URL = f"{BASE_URL}/public/auth/v1/token"
```

# Agicap

## API User Guide - Bank Journal

---

```
# Your login details
CLIENT_ID = "your_client_id"
CLIENT_SECRET = "your_secret_client"

# Authentication settings
auth_payload = {
    'client_id': CLIENT_ID,
    'client_secret': CLIENT_SECRET,
    'grant_type': 'client_credentials',
    'scope': 'agicap:public-api'
}
headers = {
    'Content-Type': 'application/x-www-form-urlencoded'
}

# Token request
response = requests.post(AUTH_URL, data=auth_payload, headers=headers)

if response.status_code == 200:
    token_data = response.json()
    access_token = token_data.get('access_token')
    # The token is valid for token_data.get('expires_in') seconds
    print("Token successfully obtained")
else:
    print(f"Authentication error: {response.status_code}")
    print(f"Details: {response.text}")
```

### 3. Retrieving your entities' identifiers

To use the Treasury Bank Journal API, you must use the ID of the entity you want to work on.

1. Go to this page:<https://app.agicap.com/fr/app/organization-advanced-settings/public-api/clients>
2. Locate your **Organization ID** displayed on this page at the top right (UUID format)
3. Use the following API to list all entities:

```
Python
import requests
```

# Agicap

## API User Guide - Bank Journal

---

```
# Configuration
BASE_URL = "https://api.agicap.com"
ORGANIZATION_ID = "your_organization_id"

def get_organization_entities(token, organization_id, page_number=1, page_size=100):
    """
    Retrieves the list of entities associated with an organization
    Settings:
    - token: Previously retrieved authentication token
    - organization_id: ID of your Agicap organization
    - page_number: Page number (starts at 1)
    - page_size: Number of items per page
    """
    url = f"{BASE_URL}/public/organizations/v1/{organization_id}/entities"

    params = {
        'pageNumber': page_number,
        'pageSize': page_size
    }

    headers = {
        'Authorization': f'Bearer {token}',
        'Accept': 'application/json'
    }

    response = requests.get(url, headers=headers, params=params)

    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error: {response.status_code}, Details: {response.text}")
        return None
```

## 4. Using the Treasury Bank Journal API

We have 3 actions available:

1. **List of exports (GET)**- This function allows you to consult the history of all your previous exports made on Agicap.
2. **Detail of an export (GET)**- This function allows you to access the full contents of a specific export to see all the entries it contains. The export IDs required for this call can be viewed via the export list function.

# Agicap

## API User Guide - Bank Journal

---

3. **Creating an export (POST)**- This function allows you to create a new bank journal containing all the entries ready to be exported. In Agicap, all entries with the "Ready to export" status will be exported and will change to the "Exported" status. When calling it, you must provide the unique identifier you wish to assign to this export.

### 4.1 List previous exports (GET)

#### `/public/treasury-bank-journal/v1/entities/{entityId}/exports`

Retrieves the history of previous exports without the detailed entries:

- Export identifiers
- Dates d'export
- Number of writes per export
- Export indices in the year
- Range of writing indices

Useful for viewing export history and obtaining the last generated ID for example.

Python

```
BASE_URL = "https://api.agicap.com/public/treasury-bank-journal/v1/entities"
```

```
def list_exports(token, entity_id, size=100, after=None, before=None):  
    """
```

```
    Retrieves the list of previous exports.
```

```
    Settings:
```

- ```
- token: Previously retrieved authentication token  
- entity_id: ID of your Agicap entity  
- size: Number of elements to return (max 100)  
- after: ISO8601 date to filter exports after this date  
- before: ISO8601 date to filter exports before this date  
    """
```

```
    url = f"{BASE_URL}/{entity_id}/exports"
```

```
    # Parameter configuration
```

```
    params = {'size': size}
```

```
    if after:
```

```
        params['after'] = after
```

```
    if before:
```

```
        params['before'] = before
```

```
    # Headers with token
```

```
    headers = {
```

```
        'Authorization': f'Bearer {token}',
```

```
        'Content-Type': 'application/json',
```

```
        'Accept': 'application/json'
```

# Agicap

## API User Guide - Bank Journal

---

```
}  
  
# API call  
response = requests.get(url, headers=headers, params=params)  
  
if response.status_code == 200:  
    return response.json()  
else:  
    print(f"Error: {response.status_code}")  
    print(f"Details: {response.text}")  
    return None
```

### 4.2 Obtain a specific export (GET)

#### `/public/treasury-bank-journal/v1/entities/{entityId}/exports/{exportId}`

Retrieves the full contents of a specific export with all its entries:

- Export metadata (entity, year, index)
- Detailed list of writings

Retrieve the list of exports and their IDs

with `/public/treasury-bank-journal/v1/entities/{entityId}/exports`

Python

```
BASE_URL = "https://api.agicap.com/public/treasury-bank-journal/v1/entities"  
def get_export(token, entity_id, export_id):  
    """  
    Retrieves the details of a specific export.  
  
    Settings:  
    - token: Authentication token  
    - entity_id: ID of your Agicap entity  
    - export_id: Unique ID of the export to consult  
    """  
    url = f"{BASE_URL}/{entity_id}/exports/{export_id}"  
  
    headers = {  
        'Authorization': f'Bearer {token}',  
        'Content-Type': 'application/json',  
        'Accept': 'application/json'  
    }
```

# Agicap

## API User Guide - Bank Journal

---

```
response = requests.get(url, headers=headers)

if response.status_code == 200:
    return response.json()
else:
    print(f"Error: {response.status_code}")
    print(f"Details: {response.text}")
    return None
```

### 4.3 Create a new export (POST)

#### `/public/treasury-bank-journal/v1/entities/{entityId}/exports/{exportId}`

Export all the writings to the status "**Ready to export**" and upgrades them to status "**Exported**". The operation allows you to export up to 5000 entries per call.

#### Important points:

- **exportId** is a UUID that must be generated by the caller and provided in the URL
- The parameter **currentExportCounts** is optional and is only relevant during the first export of the year, if the target ERP needs to know (in order to ensure numbering continuity if journals have been created outside of Agicap):
  - The number of bank journals already created in the current year
  - The number of bank journal entries already created in the current year

```
Python
import uuid
import requests

BASE_URL = "https://api.agicap.com/public/treasury-bank-journal/v1/entities"

def create_export(token, entity_id, export_id, current_counts=None):
    """
    Creates a new bank journal export.

    Settings:
    - token: Authentication token
    - entity_id: ID of your Agicap entity
      - export_id: ID unique pour l'export
    - current_counts: Current counters (for the first export of the year)
    """
```

# Agicap

## API User Guide - Bank Journal

---

```
url = f"{BASE_URL}/{entity_id}/exports/{export_id}"

headers = {
    'Authorization': f'Bearer {token}',
    'Content-Type': 'application/json',
    'Accept': 'application/json'
}

# Prepare the payload if counters are provided
payload = {}
if current_counts:
    payload['currentExportCounts'] = current_counts

response = requests.post(url, headers=headers, json=payload)

if response.status_code == 200:
    return response.json()
elif response.status_code == 204:
    return {
        "status": "success",
        "message": "Export carried out without exporting entries",
        "exportId": export_id
    }
else:
    print(f"Error: {response.status_code}")
    print(f"Details: {response.text}")
    return None
```

## 5. Practical advice

### Understanding the counters for the first export

During the first export of the year via Agicap, you can define two counters:

1. **currentBankJournalsCountInYear**: Number of bank journals already created this year excluding Agicap
  - Allows the following logs to be numbered correctly
  - Example: If you already have 5 logs, the next one will be #6
2. **currentBankJournalEntriesCountInYear**: Number of entries already created this year outside of Agicap
  - Allows the following entries to be numbered correctly
  - Example: If you already have 100 entries, the next one will be #101

# Agicap

## API User Guide - Bank Journal

---

If you have not created any journals or entries outside of Agicap this year, enter 0 for these values.

### Pagination management

For companies with a lot of exports, use the settings *after* And *before* to paginate your results:

Python

```
# First call to get the most recent exports
exports = list_exports(access_token, 140010, size=50)

# To get the following exports, use the cursor value
if exports["cursor"]["after"]:
    next_exports = list_exports(
        access_token,
        140010,
        size=50,
        after=exports["cursor"]["after"]
    )
```