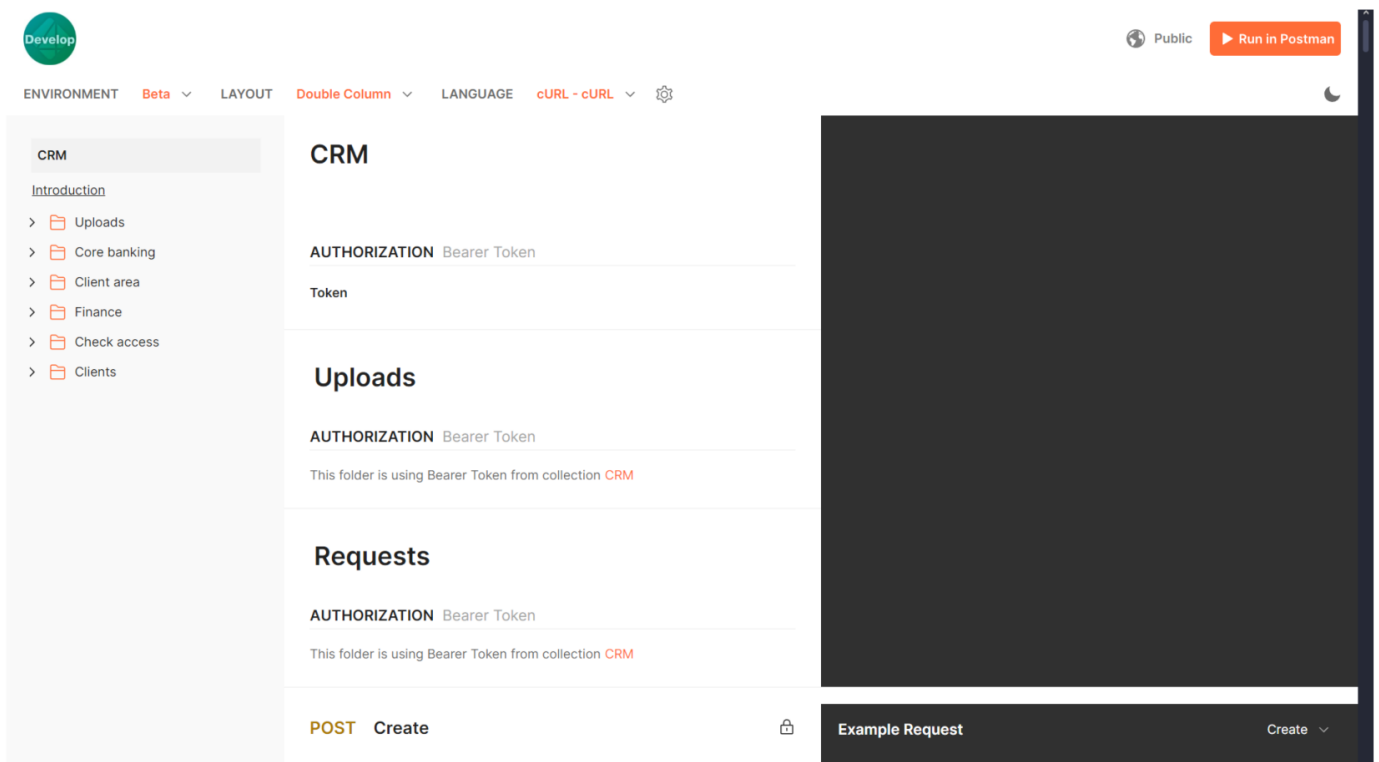


# For Developers

- [1. How to Connect to the WBCS API](#)
- [Buckets Leads Status API](#)
- [Traders Room — Sign Up API](#)

# 1. How to Connect to the WBCS API

We provide documentation with examples of API requests, which can be found [\[here\]](#). The requests are categorized by the modules they are used for, including working with **Core Banking**, **Client Area**, **Clients**, and others.



To authenticate most requests, you will need an identification token (an API authorization key for Wi CRM).

## To generate an identification token:

1. Navigate to **Security > Identification Tokens**.
2. In the upper right corner, click the **+Add** button.
3. Enter a **Name** for the token.
4. From the drop-down list, select **Role**. The selected role will determine which CRM modules will be accessible when using this token. You can read more about setting up roles [\[here\]](#).
5. If necessary, set an expiration date for the token. If no expiration date is specified, the token will remain valid indefinitely.
6. Optionally, specify a **Whitelist**: a list of IP addresses for which the token will be valid. If no whitelist is provided, the token will be valid for any IP address that has the token.

7. Click on the **Generate token** button.

**Important:** The identification token is displayed **only once**—immediately after clicking the **Generate token** button. Be sure to copy and store it securely, as it cannot be viewed again. Do not share this token with anyone, and make sure you don't lose it.

Created tokens can be modified by adding or removing IP addresses from the whitelist. Other parameters of the token cannot be edited.

Tokens can also be deleted, but once deleted, they cannot be restored.

# Buckets Leads Status API

The Buckets Leads Status API allows authorized systems to retrieve lead data for a given **bucket** and **affiliate**, using **emails and/or client IDs**, with optional filtering, pagination, and flexible response structure.

These endpoints are **read-only** and implemented as **POST** to support bulk lookups.

---

## 📄 Base path

```
/api/public
```

---

## 📄 Authentication

**Method:** Token-based via `Authorization` header

```
Authorization: <API_TOKEN>
```

- No Bearer prefix
- Token must be passed as-is

### **Required permission:**

📄 Buckets: View Own

If authentication fails:

```
401 Unauthorized
{
  "message": "Access with the provided credentials is incorrect. CODE: X-0005",
  "statusCode": 401
}
```

---

## ⚙️ Common behavior

- Method: POST
- Format: JSON
- Client type: Server-to-server
- Supports pagination via `limit` and `offset`

---

# Endpoint: Leads by bucket & affiliate

POST /buckets/{bucketId}/{affiliateID}/leads

---

## Description

Returns lead data for clients identified by:

- emails
- client IDs
- or both

Includes validation:

- Bucket existence
  - Bucket ownership by affiliate
- 

## Path parameters

- `bucketId` — bucket identifier
  - `affiliateID` — affiliate identifier
- 

## Request headers

- Content-Type: application/json
  - Authorization: `<API_TOKEN>`
- 

## Request body

```
{
  "emails": ["user1@example.com"],
  "ids": ["69ce524d741e8f2238329fff"],
  "keyBy": "email",
  "filters": {
    "createdDate": ["1676172330450", "1776172330450"]
  },
  "limit": 100,
  "offset": 0
}
```

```
}
```

## Fields

- `emails` — array of emails (optional)
- `ids` — array of client IDs (optional)
- `keyBy` — "email" (default) or "id"
- `filters.createdDate` — optional created date range (epoch ms strings)
- `limit` — number of records to return (default: 100)
- `offset` — number of records to skip (default: 0)

## Notes

- If both `emails` and `ids` are empty → returns data based on filters only
- `keyBy = "id"`:
  - uses client `_id` as key
  - falls back to email if `_id` is missing
- Pagination is applied **after filtering**

## Successful response

```
200 OK
```

Response is an object keyed by `email` or `id` (based on `keyBy`).

```
{
  "user1@example.com": {
    "id": "69ce524d741e8f2238329fff",
    "email": "user1@example.com",
    "affiliateID": "1234567890",
    "status": "active",
    "createdDate": "2021-01-01T00:00:00.000Z",
    "ftd": "2021-01-01T00:00:00.000Z"
  }
}
```

## Errors

### 400 Bad Request

- Bucket with corresponding affiliateID doesn't exist
- Provided bucketId does not belong to the provided affiliateID

## 401 Unauthorized

- Invalid or missing token
- Missing required permissions

# Endpoint: Leads by bucket, affiliate & source

```
POST /buckets/{bucketId}/{affiliateID}/leads/{sourceId}
```

## Description

Same as previous endpoint, with an additional **source validation step**.

Used for **source-level access control**.

## Path parameters

- `bucketId` — bucket identifier
- `affiliateID` — affiliate identifier
- `sourceId` — source identifier

## Request body

Same as previous endpoint.

## ⚠ Source validation behavior

- If `sourceId` is provided → system checks that **bucket has a source assigned**
- If validation fails → request is rejected

## Successful response

Same structure as previous endpoint.

## ❏ Errors

### 400 Bad Request

All previous errors, plus:

- Provided sourceId does not belong to bucket

### 401 Unauthorized

Same as previous endpoint.

---

## 📄 Example requests

### Without sourceId

```
curl -X POST "(CRM origin)/api/public/buckets/<bucketId>/<affiliateID>/leads" \  
-H "Content-Type: application/json" \  
-H "Authorization: <API_TOKEN>" \  
-d '{  
  "emails": ["user1@example.com"],  
  "ids": ["69ce524d741e8f2238329fff"],  
  "limit": 50,  
  "offset": 0  
'
```

---

### With sourceId

```
curl -X POST "(CRM origin)/api/public/buckets/<bucketId>/<affiliateID>/leads/<sourceId>" \  
-H "Content-Type: application/json" \  
-H "Authorization: <API_TOKEN>" \  
-d '{  
  "filters": {  
    "createdDate": ["1676172330450", "1776172330450"]  
  }  
'
```

# Traders Room — Sign Up API

The **Sign Up API** allows external systems to **register new clients** in the platform.

It supports:

- Creating a new user
  - Passing marketing / affiliate data
  - Passing business / routing configuration
  - Optional email notification
  - Optional instant login via autologin link
- 

## ☐☐ Use Cases

- Partner platforms
  - Affiliate systems
  - Landing pages / funnels
  - External onboarding flows
- 

## ☐☐ Endpoint

```
POST /tradersroom/api/auth/signup
```

---

## ☐☐ Authentication

All external requests **must include an API key**.

### Header

```
x-api-key: <your-api-key>
```

---

# Request Format

## Headers

Header	Required	Value
Content-Type	<input type="checkbox"/>	application/json
x-api-key	<input type="checkbox"/>	Provided by platform
Origin	<input type="checkbox"/>	Your domain

---

## Request Body

### Required Fields

```
{
  "email": "user@example.com",
  "phone": "+1234567890",
  "firstName": "Jane"
}
```

## Full Field Reference

### Basic Information

Field	Type	Required	Description
email	string	<input type="checkbox"/>	User email
phone	string	<input type="checkbox"/>	Phone number
firstName	string	<input type="checkbox"/>	First name

Field	Type	Required	Description
middleName	string	<input type="checkbox"/>	Middle name
lastName	string	<input type="checkbox"/>	Last name
gender	string	<input type="checkbox"/>	Gender
dateOfBirth	string	<input type="checkbox"/>	Date of birth (ISO format)
nationality	string	<input type="checkbox"/>	Nationality
passport	string	<input type="checkbox"/>	Passport

## Contact Information

Field	Type	Required	Description
additionalPhone	string	<input type="checkbox"/>	Secondary phone

## Affiliate & Marketing

Field	Type	Required	Description
affiliateID	string	<input type="checkbox"/>	Affiliate identifier
subID	string	<input type="checkbox"/>	Affiliate sub-id
campaignId	string	<input type="checkbox"/>	Campaign identifier
sourceId	string	<input type="checkbox"/>	Traffic source
meta	string	<input type="checkbox"/>	Additional metadata
externalId	string	<input type="checkbox"/>	External system user ID

## Business / Routing Configuration

Field	Type	Required	Description
project	string	<input type="checkbox"/>	Project identifier
desk	string	<input type="checkbox"/>	Desk identifier
manager	string	<input type="checkbox"/>	Manager (id / email / name)
status	string	<input type="checkbox"/>	Client status

Field	Type	Required	Description
state	string	<input type="checkbox"/>	Client state ( <code>active</code> , <code>live</code> , etc.)
type	string	<input type="checkbox"/>	Client type
companyFeeGroup	string	<input type="checkbox"/>	Fee group
processed	boolean	<input type="checkbox"/>	Used in custom flows

## Verification & Permissions

Field	Type	Required	Description
verificationLevel	string	<input type="checkbox"/>	Verification level
verificationStatus	string	<input type="checkbox"/>	Verification status
allowToCreateAsset	boolean	<input type="checkbox"/>	Allow asset creation
agreements	array	<input type="checkbox"/>	Signed agreements

## Traders Room Options

Field	Type	Required	Description
sendEmail	boolean	<input type="checkbox"/>	Send registration email

## Nested Objects

### billing

```
{
  "billing": {
    "country": "US",
    "region": "NY",
    "city": "New York",
    "postcode": "10001",
    "address": "Wall Street 1"
  }
}
```

```
}  
}
```

Field	Type
country	string
region	string
city	string
postcode	string
address	string

## agreements

```
{  
  "agreements": [  
    {  
      "label": "Terms and Conditions",  
      "ip": "127.0.0.1",  
      "signedAt": "2026-01-01T12:00:00Z"  
    }  
  ]  
}
```

Field	Type	Description
label	string	Agreement name
ip	string	Signing IP
signedAt	string	ISO date

## ☐ Responses

### ☐ Success — With Autologin

```
{  
  "id": "user-uuid",
```

```
"url": "https://your-domain/autologin/uuid/"
}
```

- URL allows **instant login**
- Valid for a **short time only**

---

## □ Success — Without Autologin

```
{
  "id": "user-uuid"
}
```

---

## □ Error Responses

### 400 — Invalid Request

```
{
  "message": "Email, phone and first name are required",
  "providerStatus": 400
}
```

---

### 401 — Unauthorized

```
{
  "message": "Invalid secret key",
  "providerStatus": 401
}
```

---

### 500 — Server Error

```
{  
  "message": "Internal server error",  
  "providerStatus": 500  
}
```

# Example Requests

## Basic Registration

```
curl -X POST "https://your-domain/tradersroom/api/auth/signup" \  
  -H "Content-Type: application/json" \  
  -H "x-api-key: YOUR_API_KEY" \  
  -d '{  
    "email": "user@example.com",  
    "phone": "+1234567890",  
    "firstName": "Jane"  
  }'
```

## Full Example

```
curl -X POST "https://your-domain/tradersroom/api/auth/signup" \  
  -H "Content-Type: application/json" \  
  -H "x-api-key: YOUR_API_KEY" \  
  -d '{  
    "email": "user@example.com",  
    "phone": "+1234567890",  
    "firstName": "Jane",  
    "lastName": "Doe",  
    "affiliateID": "AFF-123",  
    "subID": "SUB-1",  
    "campaignId": "CMP-456",  
    "sourceId": "facebook_ads",  
    "project": "default",  
    "desk": "sales",  
  }'
```

```
"sendEmail": true
```

```
}'
```

---

## Important Notes

- Each request creates a **new user**
- Email should be **unique**
- Autologin link is **temporary**
- Store returned `id` in your system
- API key must be kept **secure**
- Some system fields are **automatically managed** and cannot be overridden

---

## Recommended Integration Flow

1. Collect user data
2. Call Sign Up API
3. Handle response:
  - If `url` exists → redirect user
  - Otherwise → show success message
4. Store user ID