



www.remitcloud.de

Webware RRM Service
API Documentation
Version 1.0



Webware internet solutions GmbH

Teichstraße 14-16
34130 Kassel
Germany

Contents

1. Overview	2
2. Authentication	2
3. Base URL, Versioning and Formats	2
4. Error Handling	3
5. Status Values	3
6. Common Endpoints	4
6.1 Upload XML File	4
7. ACER Endpoints	5
7.1 List Files	5
7.2 Get File by load_id	5
7.3 Get File History	6
7.4 Download XML by load_id	7
7.5 Get RRM Response	7
7.6 Get ACER Response	8
8. ELCOM Endpoints	8
8.1 List Files	8
8.2 Get File by elcom_load_id	9
8.3 Get File History	10
8.4 Download XML by elcom_load_id	10
8.5 Get ELCOM Response	11
9. Additional Endpoints	11
9.1 Health / Status Check	11
9.2 Webhooks for Status Updates	12
10. Rate Limiting and Usage Limits	12
11. Example Usage	13
12. Support	14

1. Overview

The RemitCloud API allows your systems to integrate directly with our reporting platform.

Main capabilities:

- **Upload XML files** to RemitCloud
- **Track processing status** per file and load_id
- **Retrieve file history and processing results** (RRM, ACER, and ELCOM responses)
- **Download stored XML files** by load_id

Your existing web interface remains unchanged; the API is an additional technical interface for automation.

2. Authentication

Authentication method: Bearer token

Where to get the token:

In the RemitCloud **web interface**, under **Office Settings** → **API Tokens**: - Create new API tokens (give each token a name, e.g. "ERP Production") - See when a token was created and last used - Revoke tokens at any time

How to use the token:

Include the token in every API request:

Authorization: Bearer <your_api_token>

Security recommendations: - Treat the token like a password - Store it in a secure secret store - Rotate/revoke tokens regularly or when an integration changes

3. Base URL, Versioning and Formats

Base URL:

- Production: <https://rrmprod.remitcloud.de/platform/api/v1/>
- Test: <https://rrmtest.remitcloud.de/platform/api/v1/>

Versioning:

All endpoints are prefixed with the version, e.g. /platform/api/v1/....

Breaking changes will be introduced under a new version (e.g. /platform/api/v2/...).

Request / Response format: - **Request body:** JSON unless explicitly stated (e.g. file upload uses multipart/form-data) - **Response:** JSON - **Character set:** UTF-8

Standard response envelope:

```
{
  "data": { ... },
  "meta": { ... }
}
```

Field naming: All JSON fields use **snake_case** (e.g. load_id, file_name, rrm_status, created_at).

4. Error Handling

HTTP status codes: - 200 / 201 – Successful operation - 400 – Validation or input error - 401 – Authentication failed (missing/invalid token) - 403 – Not allowed for this token - 404 – Resource not found (load_id does not exist) - 409 – Conflict (e.g. duplicate load_id when not allowed) - 500 – Internal server error

Error body format:

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid input data.",
    "details": {
      "field_name": ["Error description"]
    }
  }
}
```

5. Status Values

The status field can have one of the following values:

- **RRM statuses:** RRMrejected, RRMaccepted
- **ACER statuses:** ACERrejected, ACERaccepted, ACERpartialAccepted
- **System statuses:** SYSerror, SYSok
- **Customer delivery:** SFTPtoCustomer

- **ELCOM statuses:** ELCOMrejected, ELCOMaccepted

6. Common Endpoints

6.1 Upload XML File

Purpose: Upload a new XML file for processing in RemitCloud. The file will be processed through ACER, ELCOM, or both, depending on your office configuration.

Method: POST

URL: /platform/api/v1/files

Authentication: Required (Authorization: Bearer <token>)

Content-Type: multipart/form-data

Parameters: - file (required, file): XML file to upload - Example filename: 20251126_REMITTable1_V3_file06.xml - type (required, string): Business type of the file (e.g. REMITTable1, REMITTable2) - period (optional, string): Reporting period (e.g. 2025-11 or ISO date) - external_reference (optional, string): Your internal reference or document number

Response:

```
{
  "data": {
    "load_id": 1345,
    "elcom_load_id": 789,
    "file_name": "20251126_REMITTable1_V3_file06.xml",
    "status": "RRMaccepted",
    "created_at": "2025-11-26T10:00:00Z"
  }
}
```

Notes: - load_id is an integer that uniquely identifies the file for ACER processing (present if ACER is enabled for your office) - elcom_load_id is an integer that uniquely identifies the file for ELCOM processing (present if ELCOM is enabled for your office) - If your office has only ACER enabled, only load_id will be returned - If your office has only ELCOM enabled, only elcom_load_id will be returned - If your office has both ACER and ELCOM enabled, both load_id and elcom_load_id will be returned - Use the appropriate ID (load_id for ACER endpoints, elcom_load_id for ELCOM endpoints) in subsequent API calls - The status field will be either RRMrejected or RRMaccepted after initial processing

7. ACER Endpoints

All ACER-related endpoints are under the `/platform/api/v1/acer/` prefix and use `load_id` to identify files.

7.1 List Files

Purpose: Retrieve a list of uploaded ACER files (for monitoring, reconciliation, etc.).

Method: GET

URL: `/platform/api/v1/acer/files`

Authentication: Required

Query parameters (all optional): - `load_id` (integer): Filter by exact `load_id` - `status` (string): Filter by status (e.g. `RRMaccepted`, `ACERaccepted`) - `from` / `to` (ISO datetime): Filter by creation date range - `page` (integer): Page number for pagination - `per_page` (integer): Items per page (default: 50, max: 100)

Response:

```
{
  "data": [
    {
      "load_id": 1345,
      "file_name": "20251126_REMITTable1_V3_file06.xml",
      "status": "ACERaccepted",
      "created_at": "2025-11-26T10:00:00Z",
      "last_updated_at": "2025-11-26T10:10:00Z"
    }
  ],
  "meta": {
    "page": 1,
    "per_page": 50,
    "total": 1
  }
}
```

7.2 Get File by `load_id`

Purpose: Get the current status and key information for a specific ACER file.

Method: GET

URL: /platform/api/v1/acer/files/{load_id}

Authentication: Required

Path parameter: - {load_id} (integer): The load_id returned from the upload endpoint

Response:

```
{
  "data": {
    "load_id": 1345,
    "file_name": "20251126_REMITTable1_V3_file06.xml",
    "rrm_status": "RRMaccepted",
    "acer_status": "ACERaccepted",
    "created_at": "2025-11-26T10:00:00Z",
    "last_updated_at": "2025-11-26T10:10:00Z",
    "raw_xml_download_url": "/platform/api/v1/acer/files/1345/download"
  }
}
```

7.3 Get File History

Purpose: Retrieve the processing history/events for an ACER file, including submissions and responses.

Method: GET

URL: /platform/api/v1/acer/files/{load_id}/history

Authentication: Required

Response:

```
{
  "data": [
    {
      "timestamp": "2025-11-26T10:00:00Z",
      "event": "UPLOADED",
      "status": "RRMaccepted",
      "details": {
        "rrm_receipt_file": "ReceiptRRM_20251126_REMITTable1_V3_file06.xml"
      }
    },
    {
      "timestamp": "2025-11-26T10:02:00Z",
      "event": "SENT_TO_ACER",
      "status": "SYSok",
      "details": {

```

```

    "sent_file": "20251126_REMITTable1_V3_B0001064H.DE_47.xml"
  },
  {
    "timestamp": "2025-11-26T10:10:00Z",
    "event": "ACER_RESPONSE_RECEIVED",
    "status": "ACERaccepted",
    "details": {
      "acer_receipt_file": "ReceiptACER_20251126_REMITTable1_V3_file06.xml"
    }
  }
]
}

```

Notes: - The history includes all processing steps and status changes - Receipt filenames (RRM and ACER) are included in the details field when available - Events may include: UPLOADED, SENT_TO_ACER, ACER_RESPONSE_RECEIVED, etc.

7.4 Download XML by load_id

Purpose: Download the originally uploaded XML file for a given load_id.

Method: GET

URL: /platform/api/v1/acer/files/{load_id}/download

Authentication: Required

Response: - **Content-Type:** application/xml - **Content-Disposition:** attachment; filename="<file_name>.xml"

The response body contains the XML content.

7.5 Get RRM Response

Purpose: Retrieve the RRM response for a given load_id.

Method: GET

URL: /platform/api/v1/acer/files/{load_id}/rrm-response

Authentication: Required

Path parameter: - {load_id} (integer): The load_id of the file

Response:

```

{
  "data": {

```

```

    "load_id": 1345,
    "status": "RRMAccepted",
    "rrm_receipt_file": "ReceiptRRM_20251126_REMITTable1_V3_file06.xml",
    "raw_response": "<xml>...</xml>",
    "received_at": "2025-11-26T10:05:00Z"
  }
}

```

7.6 Get ACER Response

Purpose: Retrieve the ACER response for a given load_id.

Method: GET

URL: /platform/api/v1/acer/files/{load_id}/response

Authentication: Required

Path parameter: - {load_id} (integer): The load_id of the file

Response:

```

{
  "data": {
    "load_id": 1345,
    "status": "ACERaccepted",
    "acer_receipt_file": "ReceiptACER_20251126_REMITTable1_V3_file06.xml",
    "raw_response": "<xml>...</xml>",
    "received_at": "2025-11-26T10:10:00Z"
  }
}

```

8. ELCOM Endpoints

All ELCOM-related endpoints are under the /platform/api/v1/elcom/ prefix and use elcom_load_id to identify files.

8.1 List Files

Purpose: Retrieve a list of uploaded ELCOM files (for monitoring, reconciliation, etc.).

Method: GET

URL: /platform/api/v1/elcom/files

Authentication: Required

Query parameters (all optional): - elcom_load_id (integer): Filter by exact elcom_load_id - status (string): Filter by status (e.g. ELCOMaccepted,

ELCOMrejected) - from / to (ISO datetime): Filter by creation date range -
 page (integer): Page number for pagination - per_page (integer): Items per
 page (default: 50, max: 100)

Response:

```
{
  "data": [
    {
      "elcom_load_id": 789,
      "file_name": "20251126_REMITTable1_V3_file06.xml",
      "status": "ELCOMaccepted",
      "created_at": "2025-11-26T10:00:00Z",
      "last_updated_at": "2025-11-26T10:15:00Z"
    }
  ],
  "meta": {
    "page": 1,
    "per_page": 50,
    "total": 1
  }
}
```

8.2 Get File by elcom_load_id

Purpose: Get the current status and key information for a specific ELCOM file.

Method: GET

URL: /platform/api/v1/elcom/files/{elcom_load_id}

Authentication: Required

Path parameter: - {elcom_load_id} (integer): The elcom_load_id returned from the upload endpoint

Response:

```
{
  "data": {
    "elcom_load_id": 789,
    "file_name": "20251126_REMITTable1_V3_file06.xml",
    "elcom_status": "ELCOMaccepted",
    "created_at": "2025-11-26T10:00:00Z",
    "last_updated_at": "2025-11-26T10:15:00Z",
    "raw_xml_download_url": "/platform/api/v1/elcom/files/789/download"
  }
}
```

8.3 Get File History

Purpose: Retrieve the processing history/events for an ELCOM file, including submissions and responses.

Method: GET

URL: /platform/api/v1/elcom/files/{elcom_load_id}/history

Authentication: Required

Response:

```
{
  "data": [
    {
      "timestamp": "2025-11-26T10:00:00Z",
      "event": "UPLOADED",
      "status": "SYSok",
      "details": null
    },
    {
      "timestamp": "2025-11-26T10:15:00Z",
      "event": "ELCOM_RESPONSE_RECEIVED",
      "status": "ELCOMaccepted",
      "details": {
        "elcom_receipt_file": "ReceiptELCOM_20251126_REMITTable1_V3_file06.xml"
      }
    }
  ]
}
```

Notes: - The history includes all processing steps and status changes - Receipt filenames (ELCOM) are included in the details field when available - Events may include: UPLOADED, ELCOM_RESPONSE_RECEIVED, etc.

8.4 Download XML by elcom_load_id

Purpose: Download the originally uploaded XML file for a given elcom_load_id.

Method: GET

URL: /platform/api/v1/elcom/files/{elcom_load_id}/download

Authentication: Required

Response: - **Content-Type:** application/xml - **Content-Disposition:** attachment; filename="<file_name>.xml"

The response body contains the XML content.

8.5 Get ELCOM Response

Purpose: Retrieve the ELCOM response for a given elcom_load_id.

Method: GET

URL: /platform/api/v1/elcom/files/{elcom_load_id}/response

Authentication: Required

Path parameter: - {elcom_load_id} (integer): The elcom_load_id of the file

Response:

```
{
  "data": {
    "elcom_load_id": 789,
    "status": "ELCOMaccepted",
    "elcom_receipt_file": "ReceiptELCOM_20251126_REMITTable1_V3_file06.xml",
    "raw_response": "<xml>...</xml>",
    "received_at": "2025-11-26T10:15:00Z"
  }
}
```

9. Additional Endpoints

9.1 Health / Status Check

Purpose: Verify that the API is reachable and operational. Useful for monitoring and health checks.

Method: GET

URL: /platform/api/v1/status

Authentication: Not required

Response:

```
{
  "data": {
    "status": "OK",
    "time": "2025-11-26T10:00:00Z"
  }
}
```

9.2 Webhooks for Status Updates

Purpose: Register callback URLs to receive automatic status updates when file processing events occur (e.g. when ACER or ELCOM responses are received), eliminating the need for polling.

Endpoints:

- **Register a webhook:**

POST /platform/api/v1/webhooks

Request body:

```
{
  "url": "https://your-system.com/api/callbacks",
  "events": ["ACER_RESPONSE_RECEIVED", "ELCOM_RESPONSE_RECEIVED"]
}
```

- **List webhooks:**

GET /platform/api/v1/webhooks

- **Remove a webhook:**

DELETE /platform/api/v1/webhooks/{id}

Note: Webhook configuration and available events will be provided during API setup.

10. Rate Limiting and Usage Limits

To ensure optimal performance and fair usage, the following limits apply to all API requests:

Rate Limits: - API requests are subject to rate limiting per authentication token - Rate limits are configured based on your subscription plan and usage requirements - When rate limits are exceeded, you will receive an HTTP 429 Too Many Requests response - Rate limit headers are included in all responses:
- X-RateLimit-Limit: Maximum number of requests allowed - X-RateLimit-Remaining: Number of requests remaining in the current window - X-RateLimit-Reset: Time when the rate limit resets (Unix timestamp)

File Upload Limits: - Maximum file size per upload: Configured per account (typically 50-100 MB) - Supported file formats: XML files only - File validation occurs immediately upon upload

Data Retention: - Uploaded files and processing responses are retained according to your account's retention policy - Historical data access is available for the configured retention period - Contact your account manager for specific retention details

Best Practices: - Implement exponential backoff when receiving 429 responses - Cache responses when appropriate to reduce API calls - Use webhooks (when available) instead of polling for status updates - Monitor rate limit headers to optimize your request patterns

Specific limits for your account will be provided during API setup and onboarding.

11. Example Usage

Upload a file (common endpoint - returns load_id and/or elcom_load_id)

```
curl -X POST https://rrmtest.remitcloud.de/platform/api/v1/files \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-F "file=@/path/to/20251126_REMITTable1_V3_file06.xml" \
-F "type=REMITTable1" \
-F "period=2025-11"
```

Get ACER file status (using load_id)

```
curl -X GET https://rrmtest.remitcloud.de/platform/api/v1/acer/files/1345 \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Get RRM response (using load_id)

```
curl -X GET https://rrmtest.remitcloud.de/platform/api/v1/acer/files/1345/rrm-
response \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Get ACER response (using load_id)

```
curl -X GET https://rrmtest.remitcloud.de/platform/api/v1/acer/files/1345/response
\
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Get ELCOM file status (using elcom_load_id)

```
curl -X GET https://rrmtest.remitcloud.de/platform/api/v1/elcom/files/789 \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

Get ELCOM response (using elcom_load_id)

```
curl -X GET
https://rrmtest.remitcloud.de/platform/api/v1/elcom/files/789/response \
-H "Authorization: Bearer YOUR_API_TOKEN"
```

12. Support

For API support, please contact your RemitCloud account manager or technical support team.

Contact Information: - Website: remitcloud.de - Email: support@remitcloud.de