

Meet the API

VeloxFactory ships with a fully capable REST API — and it is not an afterthought. Every action you can perform in the frontend can also be performed via the API, because **the frontend and the API share the same controller methods**. There is no separate implementation, no feature gap, no second-class citizen.

This makes the API a genuine alternative to the UI, not just an integration bolt-on. Automate report rendering in your CI pipeline, trigger prints from your ERP, manage report configurations programmatically — all with the same logic that powers the frontend you already know.

i For the full endpoint reference including all parameters and response schemas, explore the live demo API documentation at demo.veloxfactory.kiwi-software.dev.

Login: · **Password:**

Authentication

The API uses **token-based authentication** via Laravel Sanctum. Every request must include a Bearer token in the `Authorization` header.

```
Authorization: Bearer your-api-token
```

Tokens are created and managed per user — either from the user management section in the frontend, or via the API itself (`POST /api/v1/user/{id}/create-token`). Each token can be revoked at any time. A revoked token is rejected immediately — there is no grace period.

```
{
  "errors": ["This API token has been"]
}
```

The audit trail records which token was used for every create and update operation across all models, so every API action is fully traceable.

Query Parameters

Most `GET` endpoints and the render endpoint share a common set of query parameters that control what is included in the response. They are additive — combine them freely.

Parameter	Type	Default	Description
<code>limit</code>	integer	<code>25</code>	Maximum records to return. Set to <code>0</code> for all.
<code>withRelations</code>	boolean	<code>false</code>	Include related models (e.g. parameters, fields, resources on a ReportConfig).
<code>recursiveRelations</code>	boolean	<code>false</code>	Include nested relations within relations.
<code>withMedia</code>	boolean	<code>false</code>	Include Base64-encoded file content, previews, and thumbnails.
<code>withAudit</code>	boolean	<code>false</code>	Include the audit segment on each record (timestamps, users, token, method).

Example — retrieve a single report config with all relations, media, and audit data:

```
GET /api/v1/report-config/1?withRelations=true&withMedia=true&withAudit=true
```

Audit Segment

When `withAudit=true` is set, every record in the response carries an `audit` block:

```
"audit": {  
    "createdAt":  
  
    "updatedAt":
```

```
}
```

`creationMethod` and `updateMethod` are derived automatically — `"API"` when a token was used, `"Frontend"` when the action came through the UI.

Render-Specific Request Body

The render endpoint (`POST /api/v1/report-config/{id}/render`) accepts the following fields in the **request body** (JSON).

Field	Type	Required	Description
<code>outputType</code>	string	<input type="checkbox"/>	How to return the PDF: <code>base64</code> , <code>url</code> , <code>preview</code> , or <code>none</code>
<code>createHistoryRecord</code>	boolean	<input type="checkbox"/>	Whether to persist a <code>ReportHistoryRecord</code> for this render
<code>createPrintTask</code>	boolean	<input type="checkbox"/>	Whether to create a <code>ReportPrintTask</code> after rendering
<code>printerName</code>	string	if <code>createPrintTask</code>	Name of the target printer
<code>useExampleValues</code>	boolean		Use stored example values instead of supplying parameters manually
<code>parameters</code>	object		Key-value map of parameter names to values
<code>data</code>	array		Array of data rows — for reports without a live DB connection
<code>numberOfCopies</code>	integer		Number of print copies (default: 1)
<code>broadcastId</code>	string		WebSocket channel ID — triggers real-time status updates for the print task
<code>traceId</code>	string		Custom trace ID; auto-generated as UUID if omitted

Field	Type	Required	Description
<code>laconicResponse</code>	boolean		Strip the response to just the PDF output (see below)

⚠ **Output type** `none` **requires** `createPrintTask: true`. Rendering without any output and without a print task is rejected.

Laconic Responses

By default, a render response is fully populated — it includes the echoed input, the PDF output, the linked `ReportConfig`, the created `ReportHistoryRecord`, and the `ReportPrintTask` if applicable. In high-frequency or bandwidth-sensitive scenarios, add `laconicResponse=true` to strip everything down to just the PDF output.

Rate Limiting

The API enforces a configurable rate limit per token. The default is **10 requests per minute**. When the limit is exceeded:

```
{
  "message": "Too many requests! The current rate limiting is 10 requests per minute."
}
```

The limit is adjusted via the `API_RATE_LIMIT_PER_MINUTE` environment variable — see [Configuration and Data Models](#).

Endpoints Overview

Resource	Available operations
----------	----------------------


```
        "repo",
        "reportHistoryf",
    },
},
}
```

Add `laconicResponse=true` and the response collapses to just `output` — no echoed input, no embedded related records. Ideal for automated pipelines that only need the PDF.

Revision #11
Created 2026-05-10 10:43:08 UTC by Benjamin Fischer
Updated 2026-05-27 18:40:41 UTC by Benjamin Fischer