

LNApi Overview

As of August 26, 2026

Documentation note: LNApi capabilities, fields, validation rules, and examples may change as releases are finalized. Confirm current behavior with Benedict Group before relying on details for production.

LNApi is Benedict Group's API for building integrations with LFN (Loans for .NET). Applications call LNApi over HTTPS with JSON; LNApi applies changes inside LFN with validation, audit trail, and security comparable to an operator working in the application.

These pages are written for the developer building the integration. This PDF set has three documents:

PDF	What it covers
Overview	Core concepts, first call, authentication, users, and error handling
Install	IIS hosting, .NET Hosting Bundle, deployment, upgrades, and troubleshooting
Insurance Renewal	Endpoint reference for the insurance renewal capability

The short version

```
curl https://YOUR-LNAPI-HOST/v1/insurance-renewal \
-H "X-API-Key: bgi_test_3F9KQ2ba_yourSecretHere" \
-H "X-End-User: jsmith" \
-H "Content-Type: application/json" \
-d @renewal.json
```

One API key and one user identifier are sent on every call. There is no login step, no tokens, and no sessions.

What you need

- **Your LNApi base URL.** LNApi runs on your organization's own servers, alongside LFN. Your IT team or Benedict Group will provide the hostname.
- **An API key.** Keys are issued by Benedict Group when your integration is set up. Store the key somewhere safe immediately; it is shown once.

There is no registration flow, OAuth exchange, or token refresh to build. If you have the URL and key, you can make a request.

Test and live

Your site may run two LNApi instances: a **test** instance pointed at a copy of your data, and a **live** instance. Keys are environment-specific. A key that starts with `bgi_test_` works only against test; a key that starts with `bgi_live_` works only against live. Mixing them up fails safely with a `401`.

Build and verify against test first.

Your first call

```
curl "https://YOUR-LNAPI-HOST/v1/insurance-renewal?dryRun=true" \
-H "X-API-Key: bgi_test_3F9KQ2ba_yourSecretHere" \
-H "X-End-User: jsmith" \
-H "Content-Type: application/json" \
-d '{
  "projectNumber": "001922",
  "loanNumber": "001922ESC",
  "insuranceType": "P",
  "certificateNumber": "CERT-2026-0912",
  "items": [
    {
      "instyp": "HZ",
      "policyStatus": "00",
      "insuranceCompanyName": "Sample Insurance Co",
      "insuranceAgencyName": "Sample Agency LLC",
      "premiumAmount": 1250.00,
      "escrowFlag": "N",
      "policyDate": "2026-09-01",
      "policyExpirationDate": "2027-09-01"
    }
  ]
}'
```

A successful call returns HTTP 200 with one result per item. The `?dryRun=true` query plans the request without changing anything.

Request conventions

- **Versioned URLs.** Every endpoint lives under `/v1/`. If a future breaking change is introduced, it will be published under a new version while `/v1/` keeps working.
- **Two headers on every call.** `X-API-Key` identifies your integration. `X-End-User` identifies the person or service process the call is made for.
- **JSON requests.** Send `Content-Type: application/json`; dates are ISO format, such as `2026-09-01`.
- **Status codes first.** Successful responses, validation errors, and platform refusals return JSON. Some errors have an empty or plain-text body, so switch on HTTP status code before parsing.
- **HTTPS only.** Plain HTTP is rejected. Never disable certificate verification in an HTTP client to make a certificate error go away.
- **No server session.** The API key rides every call, and each request stands alone.

Authentication

LNapi uses API keys. One key identifies your integration; send it on every request in the `X-API-Key` header.

```
X-API-Key: bgi_live_3F9KQ2ba_kX4mP9vQ2wR8sT1uV5xY7zA3bC6dE0fG
```

Anatomy of a key

```
bgi_live_3F9KQ2ba_kX4mP9vQ...
| | |
| | | +-- secret (random, never stored by anyone but you)
| | +----- key prefix (safe to show in logs)
| +----- environment: live or test
+----- always "bgi"
```

The prefix is not secret and may appear in logs to identify which key made a call. The secret is the credential. Send the whole string exactly as issued.

How your key is protected

- The key is shown exactly once when issued. Benedict Group stores only a cryptographic hash.
- If the key is lost, it cannot be recovered. A new key is issued and the old one is retired.
- Authentication checks compare hashes in constant time, and failed attempts are logged.

Your key responsibilities

- Store the key in a secret manager or protected server configuration.
- Never store the key in source code, a repository, or a client-side app.
- Ask for separate keys for separate applications so each can be revoked independently.
- Send the key only over HTTPS.
- Contact Benedict immediately if the key may have been exposed.

Users and licensed seats

Every LNApi call is made on behalf of someone: a person using your application, or a service process you run. Add `X-End-User` alongside the API key on every call.

```
X-API-Key: bgi_live_3F9KQ2ba_kX4mP9vQ...  
X-End-User: jsmith
```

Rules for the value:

- Use 1-64 printable ASCII characters.
- Use one stable identifier for each person, such as the user id from your system.
- Use one stable service identity for a process, such as `coi-feed` or `nightly-renewals`.
- Do not send sensitive data. The identifier is visible in logs and support tooling.

Because every distinct `X-End-User` value occupies a licensed user seat, never send a random value, session id, timestamp, or per-request GUID. That mistake can consume seats immediately and make normal users wait.

How seats behave

- An identity's first active call claims a seat, and later calls by the same identity refresh that seat.
- When all licensed seats are occupied, a new identity receives 429 with `"code": "user_limit"`.
- Identities already holding seats are not interrupted.
- Seats are released automatically after activity stops.

Seeing your seats

`GET /v1/seats` shows how many licensed seats are in use and which identities currently hold them. The call is authenticated with your key and does not require `X-End-User`.

```
curl https://YOUR-LNAPI-HOST/v1/seats \
-H "X-API-Key: bgi_test_3F9KQ2ba_yourSecretHere"
```

```
{
  "licensed": 5,
  "inUse": 3,
  "available": 2,
  "activeUsers": [
    { "endUser": "jsmith", "keyPrefix": "3F9KQ2ba", "lastCapability": "insurance-renewal" }
  ]
}
```

Field	Meaning
licensed	Your organization's concurrent API user count
inUse / available	Seats occupied now, and how many new identities could take a seat
activeUsers	Which identities are holding seats and which capability each last used

Common integration shapes

A portal with people in it. Your web app already knows who is logged in. Send that user's id as `X-End-User` on each call.

An unattended feed. Your scheduler can send a fixed identity such as `X-End-User: coi-feed`. A process is a user under the license, and this is how it takes its seat.

A person triggering a batch. If a capability has a batch form, use it so one human action can be processed as one API request. For insurance renewal, use `POST /v1/insurance-renewal/batch`.

Error handling

Every LNapi response uses standard HTTP status codes.

Status	Meaning	What to do
200 OK	The request was accepted and processed	Read the JSON body for the outcome
400 Bad Request	Missing or invalid <code>X-End-User</code> , invalid JSON, validation failure, or plain HTTP	Fix the header or request; switch to HTTPS for HTTP rejection
401 Unauthorized	API key is missing, wrong, revoked, or for the wrong environment	Check <code>X-API-Key</code> , the key value, and the instance
415 Unsupported Media Type	Missing or wrong <code>Content-Type</code>	Send <code>Content-Type: application/json</code>
404 Not Found	Wrong URL path, or a capability is not enabled for your organization	Check the path and version segment; contact Benedict Group if needed
405 Method Not Allowed	Wrong HTTP method	Use the method shown in the capability reference
413 Payload Too Large	Body exceeds the size cap	Reduce the request body
429 Too Many Requests	A platform limit refused the request	Switch on code, honor <code>Retry-After</code> , and surface the message if a person is waiting

Status	Meaning	What to do
500 Internal Server Error	Something failed on the server	Retry later; contact Benedict Group if it persists

Platform refusals use a small JSON shape with a `code`, `message`, and optional supporting fields. Switch on `code`; the `message` is written to be shown to a person.

Code	Meaning	What your code should do
<code>end_user_required</code>	X-End-User is missing	Add a stable user or service-process identifier
<code>end_user_invalid</code>	X-End-User uses an unsupported value	Send 1-64 printable ASCII characters
<code>rate_limit</code>	The identity is being throttled	Honor <code>Retry-After</code> and retry
<code>user_limit</code>	All licensed seats are occupied for a new identity	Honor <code>Retry-After</code> or tell the user that licensed API users are busy

Validation failures return the standard ASP.NET problem-details shape with one entry per field. Business outcomes are not HTTP errors: a request can return `200` while an individual item is skipped or errors. Always check each item result in the capability response.

More capabilities are coming

Insurance renewal is the first LNApi capability. Additional capabilities will be added to the same API with the same base URL, key handling, and conventions.

Getting help

Reach your Benedict Group support contact - the same channel your organization already uses for Benedict Group support - for keys, environment questions, or anything these pages do not answer.