BENEDICT

LNApi Insurance Renewal

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.

Submits a reviewed insurance renewal into LFN, replacing manual re-keying of certificate data into the Ins Policies screen.

Pilot phase: writes are currently disabled while the field mapping is finalized with your Benedict Group contact - submissions validate and dry-run-plan but do not change records yet. Every response you receive during the pilot shows exactly what *would* happen; when writes go live, the contract on this page does not change.

Endpoint

```
POST /v1/insurance-renewal (one loan)
POST /v1/insurance-renewal/batch (many loans in one call - see "Batch" below)
X-API-Key: <your key>
X-End-User: <the person or process this call is for - required>
Content-Type: application/json
```

Add `?dryRun=true` to plan without applying - the response shows exactly what would happen (which policies would be updated, which bill numbers would be created, what escrow balance would transfer), with no database changes. Use it to verify a submission before sending it for real.

Request

One request covers one loan's renewal. The header identifies the loan; `items` carries one entry per insurance sub-type (each sub-type may appear at most once per request).

Header fields

Field	Type	Required	Notes
`projectNumber`	string	yes	The project the renewal belongs to; must match the loan on file
`loanNumber`	string (<= 12)	yes	The loan number for the renewal
`insuranceType`	string	yes	`"P"` or `"L"`
`certificateNumber`	string	yes	The certificate this submission came from; echoed back in the response
`items`	array	yes	At least one item; one per insurance sub-type

Item fields

Field	Type	Required	Notes
`instyp`	string (<= 2)	yes	LFN insurance sub-type code; must be on file (e.g. `HZ`, `FL`)
`policyStatus`	string	yes	`"00"` or `"02"` (active statuses only - archiving the prior policy is handled automatically)
`insuranceCompanyName`	string	yes	Exact name of a company on file in LFN
`insuranceAgencyName`	string	yes	Exact name of an agency on file in LFN
`policyCertificateNumber`	string (<= 30)	no	Policy/certificate number to record
`premiumDueDate`	date	no	
`policyExpirationDate`	date	no	
`policyBeginDate`	date	no	
`policyTerm`	integer (1-999)	no	Months
`disbNumberMonths`	integer (1-999)	no	Months
`premiumAmount`	decimal >= 0	no	
`escrowFlag`	string	yes	`"Y"` (escrowed - old bill is archived, a new bill is created, escrow balance transfers) or `"N"` (non-escrowed - policy is updated in place). Turning escrow **off** (`"N"` on a currently escrowed policy) is rejected while the policy still has an escrow balance on file. Sending `"Y"` for a policy that was not escrowed starts escrow on the new bill with a zero opening balance.
`payIntEsc`	string	no	`"Y"` or `"N"`
`aggregate`	decimal >= 0	no	**Currently rejected** - the target column is being confirmed with Benedict; omit it for now
`amtPerOccurrence`	decimal >= 0	no	Must be a **whole dollar amount** (the target column has no cents)
`mortgageClauseNote`	string (<= 120)	no	Recorded as the policy comment
`policyDate`	date	yes	The renewal's own date; must be strictly newer than the policy on file or the item is skipped. (If the policy on file has no comparison date recorded, the check is skipped and the renewal applies.)

Dates are ISO format only - `2026-09-01`. US-format dates (`09/01/2026`) are rejected with a `400` before any item is looked at. If your toolchain produces US-format dates, convert them before sending.

Omitted means unchanged. Every field marked "no" above can be left out, and leaving it out never erases anything: on an in-place update the value on file stays as it is, and on an escrowed renewal the new bill **carries the old bill's value forward**. Send a field only when you have a value for it - never send empty strings or zeros as placeholders.

Bill numbers are **never** supplied by the caller - LNApi computes them server-side.

Response

`200 OK` with the loan, the certificate number echoed back, and one result per item, in request order:

Field	Meaning
`instyp`	Which item this result is for
`status`	`applied` (the change was made), `skipped` (nothing to do - see `reason`), or `error` (this item could not be processed - see `reason`)
`reason`	Plain-English explanation for `skipped`/`error` (and the "dry-run: would ..." plan text); `null` for a live `applied`
`newBillNumber`	The bill number the renewal lives under - the existing bill for an in-place update, the newly created bill for an escrowed renewal

Each item stands alone. A request returns HTTP 200 even when some items error: one sub-type failing does not stop the others, and an `applied` result is committed regardless of what happened to its siblings. Always check every result - never treat "200" as "everything applied".

Example

Request - two items, one escrowed, one not:

```
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",
 "policyCertificateNumber": "POL-12345",
 "premiumDueDate": "2026-09-01",
 "policyExpirationDate": "2027-09-01",
 "policyBeginDate": "2026-09-01",
 "premiumAmount": 1250.00,
 "escrowFlag": "Y",
 "policyDate": "2026-09-01"
 },
 {
 "instyp": "FL",
 "policyStatus": "00",
 "insuranceCompanyName": "Sample Insurance Co",
 "insuranceAgencyName": "Sample Agency LLC",
 "policyCertificateNumber": "POL-67890",
 "policyExpirationDate": "2027-03-01",
 "premiumAmount": 480.00,
 "escrowFlag": "N",
 "payIntEsc": "N",
 "policyDate": "2026-03-01"
 }
 ]
 }'
```

Response (from a live run - one applied, one skipped):

```
{
 "loanNumber": "001922ESC",
 "certificateNumber": "CERT-2026-0912",
 "results": [
 {
 "instyp": "HZ",
 "status": "applied",
 "reason": null,
 "newBillNumber": 4
 },
 {
 "instyp": "FL",
 "status": "skipped",
 "reason": "policy date 03/01/2026 is not newer than the current policy (03/01/2026) -
   nothing to update",
 "newBillNumber": null
 }
 ]
}
```

With `?dryRun=true` (or during the pilot phase), the `applied` result comes back as `skipped` instead, with a plan like `"dry-run: would archive bill number 3 and create bill number 4, transferring escrow balance 1180.50"`.

Malformed requests (missing required fields, bad formats, non-ISO dates) return `400` with a per-field list before any item is looked at. Business rules that depend on what's on file (unknown companies, date-not-newer, and the like) come back as **per-item results on a 200**, not as a 400. Anything else, see the LNApi Overview error handling section.

Batch: many loans in one call

When one action in your system touches many loans - approving forty renewals, or a nightly feed - send them together. A batch keeps the work efficient and avoids unnecessary throttling.

```
POST /v1/insurance-renewal/batch
```

The body is a `renewals` array; **each element is exactly a single-renewal request** (the same header fields and `items[]` as above - copy-paste identical). The response is a `renewals` array of single-renewal responses, in request order; each self-identifies by `loanNumber` and `certificateNumber`.

```
{ "renewals": [
 { "projectNumber": "001922", "loanNumber": "001922ESC", "insuranceType": "P",
   "certificateNumber": "CERT-A", "items": [ ... ] },
 { "projectNumber": "001923", "loanNumber": "001923ESC", "insuranceType": "P",
   "certificateNumber": "CERT-B", "items": [ ... ] }
] }
```

Limit	Value	Why
Renewals per batch	25	Keeps one call's database work bounded
Items per batch, total	100	Keeps the body comfortably under the 256 KB request cap
Items per renewal	50	Same as the single form
Same loan twice in one batch	not allowed	Two renewals for one loan would race each other's bill numbers

Validation is all-or-nothing: if any renewal has a problem, the whole batch returns `400` with every problem listed, each prefixed `renewal N (loanNumber): ...`, and nothing is processed. Fix and resend the batch.

Processing is per item, exactly as the single form: HTTP `200` even if some renewals or items error; each item commits independently; `?dryRun=true` (and the pilot's writes-off state) applies to the whole batch. If a batch fails partway (500 or timeout), a full re-send is harmless - the newer-than check skips anything already applied - but confirm before re-sending, as with the single form.

Reasons you'll see

Per-item `reason` texts are stable and written for humans. The ones you may encounter:

Reason	What to do
`loan {loanNumber} not found`	Check the loan number against LFN.
`project number does not match this loan`	The project and loan belong to different records - verify both against your servicing statements.
`no active {instyp} policy on file for this loan - first-time policies must be added in LFN first`	Renewals only update existing policies; ask your Benedict Group contact to add the first policy manually.
`policy date {date} is not newer than the current policy ({date}) - nothing to update`	LFN already has this renewal (or a newer one). Nothing to fix - this is the normal outcome for a re-send.
`insurance company '{name}' is not on file in LFN - add it via Insurance Processing, then resend`	The name must exactly match a company in LFN. Fix the spelling or have the company added.
`insurance company name '{name}' matches more than one company on file - contact Benedict Group to resolve the duplicate`	Benedict Group staff must merge or rename the duplicates before this item can process.
`insurance agency '{name}' is not on file in LFN - add it via Insurance Processing, then resend`	Same as the company message, for the agency.
`insurance agency name '{name}' matches more than one agency on file - contact Benedict Group to resolve the duplicate`	Same as the company message, for the agency.
`'{instyp}' is not a recognized insurance type`	Use a sub-type code from LFN's insurance-type reference table.
`aggregate is not yet supported - the target column is being confirmed; omit the field and resend`	Drop the `aggregate` field and resend; it will be supported once the mapping is confirmed.
`FHA-escrowed policies are not supported by the API - process this renewal in LFN`	FHA escrow follows a different flow; hand this one to Benedict Group staff.
`escrow detail record not found for the current policy - contact Benedict Group`	The policy's escrow record is missing or irregular - Benedict Group staff need to look at it.
`this policy has an escrow balance on file - resolve it in LFN before turning escrow off`	Escrow can't be switched off while money sits in the escrow record - Benedict Group staff resolve the balance first.
`a pending disbursement dated {date} exists for the current policy - process or remove it in LFN, then resend`	A scheduled payment is still open on the old bill; Benedict Group staff clear it, then resend. (Only on installations with this check enabled.)
`the loan is delinquent more than {n} days - archiving the current policy is not allowed; resolve the delinquency in LFN`	The loan's payment status blocks the archive. (Only on installations with this check enabled.)
`dry-run: would update policy in place (bill number {n})`	The plan for a non-escrowed renewal; nothing was changed.
`dry-run: would archive bill number {old} and create bill number {new}, transferring escrow balance {balance}`	The plan for an escrowed renewal; nothing was changed.

Reason	What to do
`this policy was changed by someone else while processing - resend the item`	An LFN operator touched the record at the same moment. Safe to resend.
`audit collision - resend the item`	A rare timing collision in the audit trail. Safe to resend.
`a concurrent renewal created the same bill number - resend the item`	Two submissions raced. Safe to resend - the retry will compute the next bill number.
`an unexpected error occurred processing this item - Benedict Group has been notified (ref {id})`	Not your request's fault. Quote the `ref` id to your Benedict Group contact if you follow up.

What happens inside LFN

Every applied change writes LFN's standard change audit, attributed to your integration's API user: an in-place update records each changed field's before and after values, and an escrowed renewal records the new policy and escrow records plus the archival of the old ones - exactly the trail an operator's manual entry would leave, so staff can always tell API updates from manual ones.

Good citizenship

- **Use the batch form for many loans.** One call per human action or feed run keeps requests efficient; a loop of single calls is usually slower.
- **Don't auto-retry on 400.** A validation failure won't fix itself - surface it to a person.
- **Check every per-item result.** HTTP 200 means the request was processed, not that every item applied.
- **Use `?dryRun=true` first** when integrating or when a submission is unusual - the plan text tells you exactly what a live run would do.
- **On a 500 or timeout, confirm before re-submitting.** A failure partway through could mean some items were recorded and some were not - each item commits independently. Check with your Benedict Group contact (or the change-audit screens) before sending the same renewal again; the newer-than check makes an accidental duplicate harmless, but confirming is still the right habit.