Push user attributes via API

Last Modified on 18.06.2026

About the Attribute Import API

The Attribute Import API lets you send user attributes into Userlane from your own systems, in JSON format. Use it to push attributes such as role, department, region, or licence type from your data warehouse, internal service, or a scheduled job, so Userlane always reflects the user attribute data you already hold.

Why use it

Use the API to:

- keep Userlane user attributes current automatically, on your own schedule, instead of uploading spreadsheets by hand
- segment analytics and target content by attributes that live in your own systems, not just those sent at login

What you can send

In a single request you can send two kinds of rows, on their own or together:

- **Attribute definitions:** create or update the attributes themselves (a name and a type).
- **User profile values:** set values for individual users, keyed by your own public user ID.

You only need a definition for an attribute that does not exist yet. If the attribute already exists, skip the definitions and just send the value under its key in the user's profile. Send a definition only when you are introducing a new attribute, or changing an existing one's name or type. In most ongoing updates you are only sending user values.

Company or property scope

There are two versions of the endpoint. They work identically and differ only in scope:

- **Company endpoint:** the attribute applies across your whole company.
- **Property endpoint:** the attribute belongs to a single application.

An attribute sent to one scope never affects the other.

Company endpoint

Your company ID is a number.

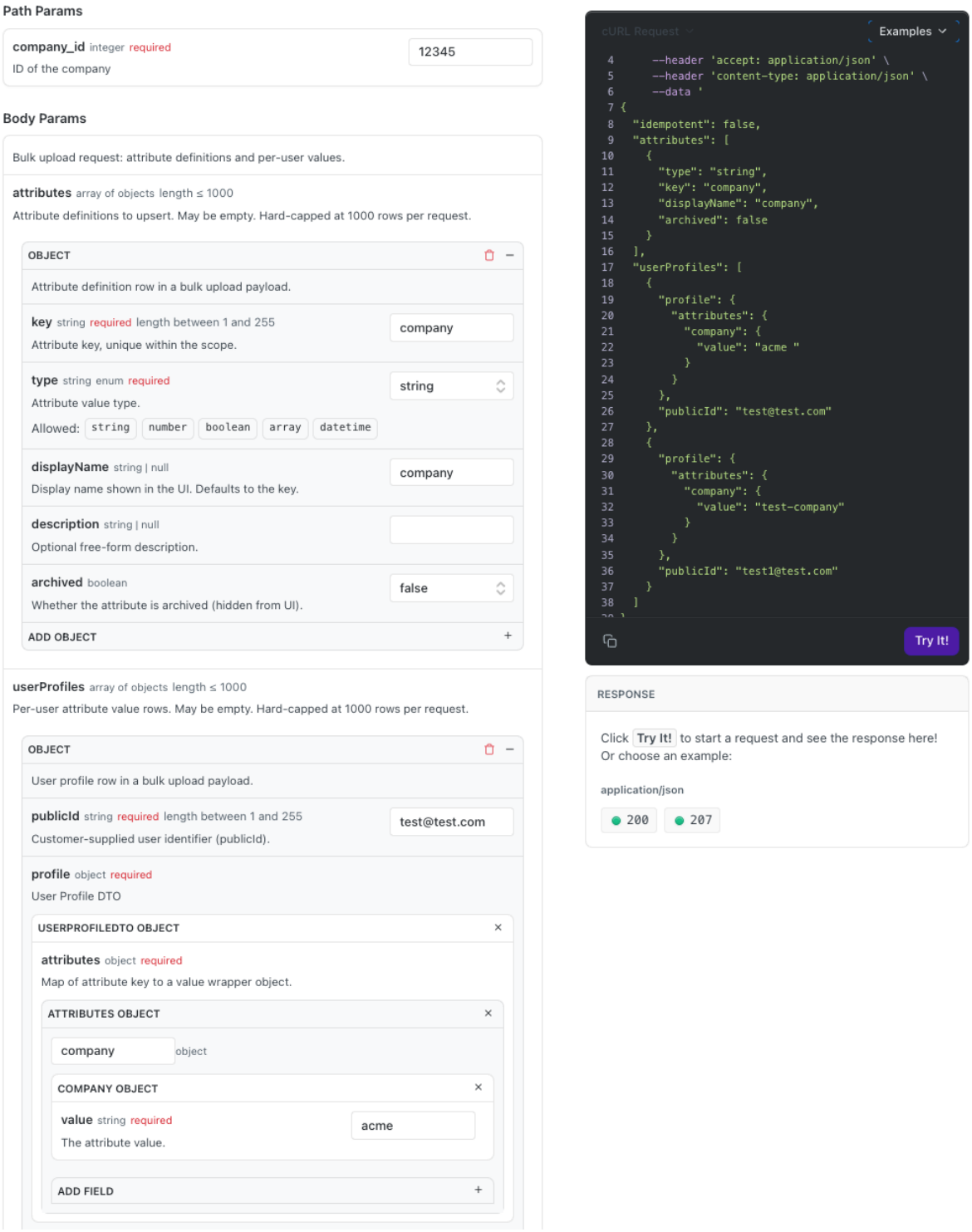
Public ID: test-user-123 USER CREATED

Last seen: - • Date created: **28 May 2026**

Company Attributes | Application Attributes

Attribute ↑	Value
City	Munich
role	product
is-manager	no
test-attribute	Not Set

Company endpoint



Property endpoint

In the API, a "property" is what the Userlane Portal calls an application, your property ID is a short text ID that you can find in your url path in the Portal when you are logged in to your property.

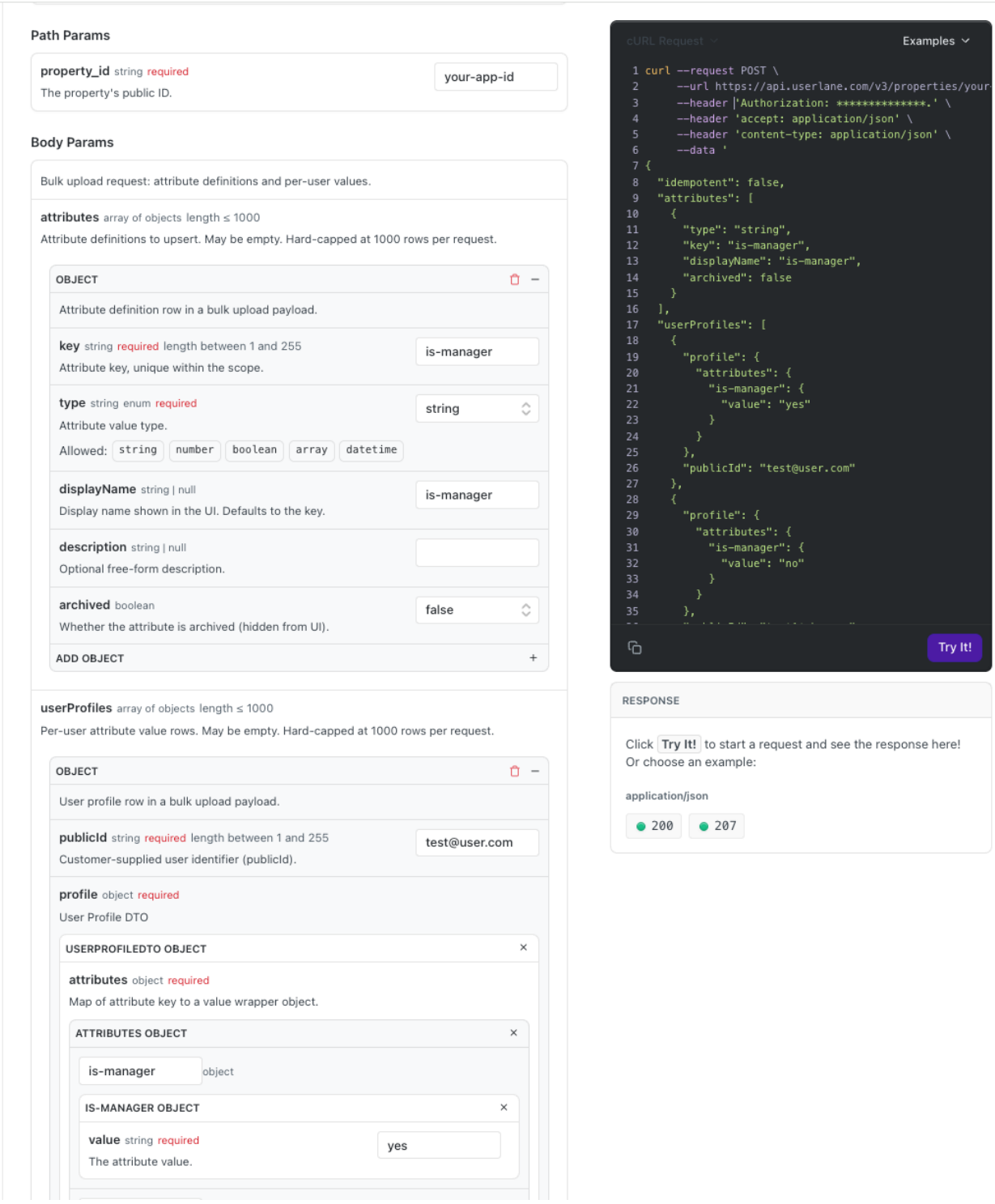
Public ID: test-user-123 USER CREATED

Last seen: - • Date created: **28 May 2026**

Company Attributes | Application Attributes

Attribute	Value
salesforce-admin	yes
permission-x	455
permission-y	Not Set

Property endpoint



How values are sent

Each value is wrapped in a small object with a `value` field. You do not send the value on its own.

Sending the value without the `{ "value": ... }` wrapper is the most common reason a first request is rejected.

To put the two pieces together: to set a value on an **existing** attribute, send only the user profile part with the value under its key. To introduce a **new** attribute, add a definition object for it first, in the same request, then set its value for the user.

Add new values, or replace existing ones

One setting, `idempotent`, controls what happens to a user's existing values:

- **`idempotent: false` (the default) adds.** The values you send are created or updated. Anything the user already has is left untouched. This is the safe, everyday mode.
- **`idempotent: true` replaces.** The values you send are written, and any other values that user already has, within the same scope, are removed.

Leave `idempotent` out when you want to add to what a user has. Only set it to `true` when your request contains the complete, final set of values for the users it names, and you intend to clear anything else. `idempotent: true` only affects the users you include in the request: users you do not mention are never changed.

How to push the data

Generate your token

The API is protected and requires authentication via an Authorization Token. You can generate your token in the Userlane Portal under **Settings > API Token**. The token is generated for each Userlane Manager individually, and the manager must be a **company admin**.

Test a request before you automate

Each endpoint is available in our developer documentation, which includes a "Try It" button you can use to send a request directly from the browser. This is the fastest way to confirm your token, IDs, and request body are correct before you write any code.

This is not a test sandbox. A request sent from "Try It" writes to the real users you name, using your real token. Always test against a test **user whose attribute values do not matter**, so your first attempts never touch real data.

To run your first request:

1. Open the Attribute Import API reference and choose the company or property endpoint.
2. Paste your token into the **Authorization** field.
3. Enter your **company ID** or **property ID** in the path field.
4. In the request body, set one attribute and one value for your**test user's** ID.
5. Leave `idempotent` as `false`
6. Click **Try It** to send the request.
7. Open that test user's profile in the Userlane Portal and confirm the value now appears.

Once a simple request works end to end, build the same call into your own system and run it on your schedule.

What each request needs

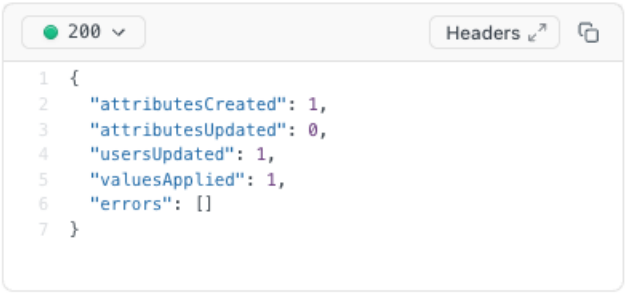
- **Company ID or Property ID** — the scope you are sending to, in the path.
- **Authorization token** — pasted into the Authorization field.
- **Attributes, user values, or both** — in the request body, with each value wrapped as `{ "value": ... }`.
- **`idempotent`** (optional) — `false` to add (the default), `true` to replace.

Limits

You can send up to 1,000 attribute definitions and up to 1,000 user records in a single request. The two limits

are separate. To update more users than that, split them across several requests (for example, 5,000 users as five requests of 1,000).

Reading the result



Every request returns a summary of what happened: how many attributes were created or updated, how many user values were applied, and an `errors` list.

- If `errors` is empty, everything succeeded.
- If some rows failed, each entry names the exact user and attribute that failed and the reason, so you can fix just those and send again.

A response code of `207` means part of your request succeeded and part did not: check the `errors` list. A `422` means the whole request was rejected and nothing was applied: correct the request and resend.

Good to know

- **Attributes are matched by their key.** Sending a key that already exists updates it; sending a new key creates a new attribute. To avoid duplicates, keep your keys consistent (for example, always `job_title`, not sometimes `jobTitle`).

Troubleshooting

If the request fails, you will see an error in the result section. Most issues relate to:

- a token whose manager is not a company admin (`403`)
- an invalid or expired token (`401`)
- a request body that did not validate, usually a value missing the `{ "value": ... }` wrapper (`422`)
- values disappearing, which means the request was sent with `idempotent: true`

Review your parameters against this article. If the issue persists, please reach out to us.

For full request and response schemas, copy-pasteable code samples (curl, Python, Node), and every error code with its meaning, see the Attribute Import API reference.