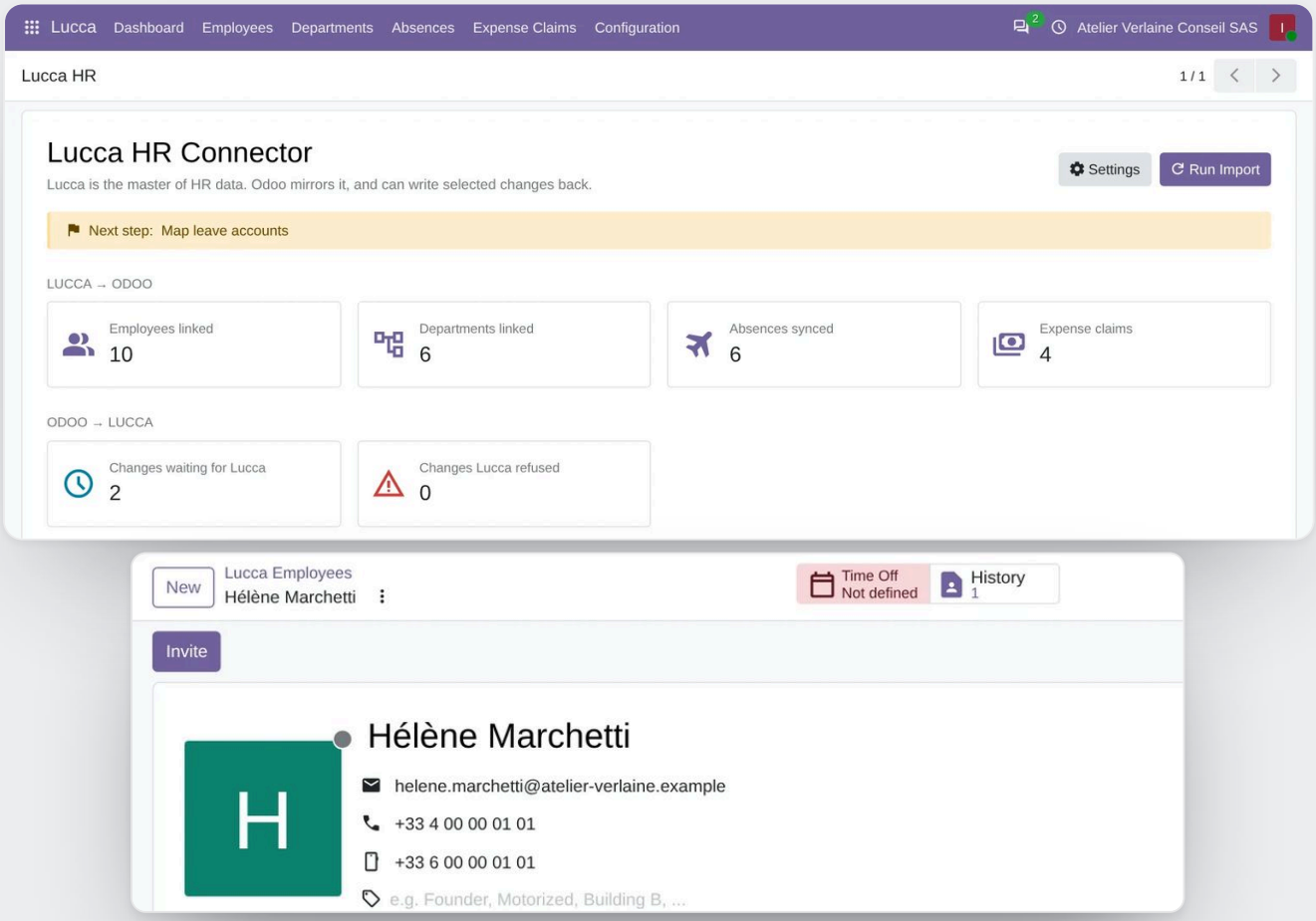
CONNECTOR GUIDE · PAYROLL AND HR

Lucca connector for Odoo

Bring your Lucca employees, departments, absences and expense claims into Odoo.

Odoo 17 · 18 · 19 · 20 | 14-day trial, no card | Edition of 19-11-2026



Real Odoo 20 screenshot. The **Lucca** dashboard with its counters, and the record of a Lucca employee inside Odoo.

Contents

On the right, the page where each chapter starts. Menu and button names are those of Odoo in English.

1	Who it's for and what it solves	3
2	The connector in figures	4
3	Where you see it in your Odoo	5
4	What you need before you start	6
5	The FlexigoTech Cloud trial	6
6	Connecting your Lucca instance step by step	8
7	Lucca employees in Odoo	10
8	Lucca departments in Odoo	13
9	From Lucca absence to Odoo absence	15
10	Lucca expense claims in Odoo	18
11	Analytic accounts to Lucca	21
12	Changes sent, webhooks and log	23
13	Webhooks: Lucca changes in real time	25
14	Initial import and dashboard	27
15	The data that goes and comes back	30
16	What happens automatically	31
17	The Lucca options and how they appear	32
18	Security and personal data	34
19	Several Lucca accounts and several companies	34
20	If something fails	35
21	Other ways to do it	36
22	Plans and pricing	37
23	Frequently asked questions	38
24	Glossary	40
25	Checklist and contact	42

This guide describes version 20.0.2.3.5 of the module (Odoo branch 20.0, commit 882d925bc). Everything in it comes from its published code, the platform's official documentation or real screenshots.

The screenshots come from a test Odoo 20 database, with a made-up company and data and no real connection to Lucca. The module also has a version for Odoo 17, 18 and 19.

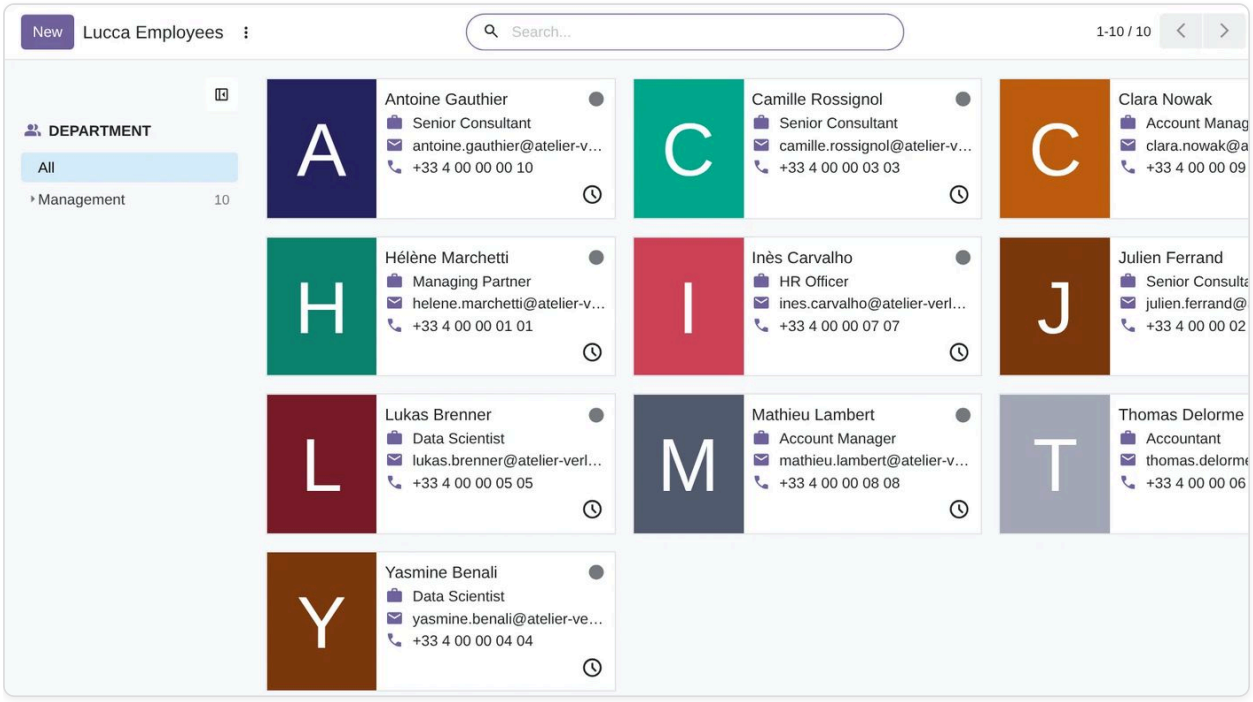
1 Who it's for and what it solves

> The connector needs a FlexigoTech Cloud subscription, with a 14-day trial and no card required. Without the trial or a subscription, it does not talk to Lucca.

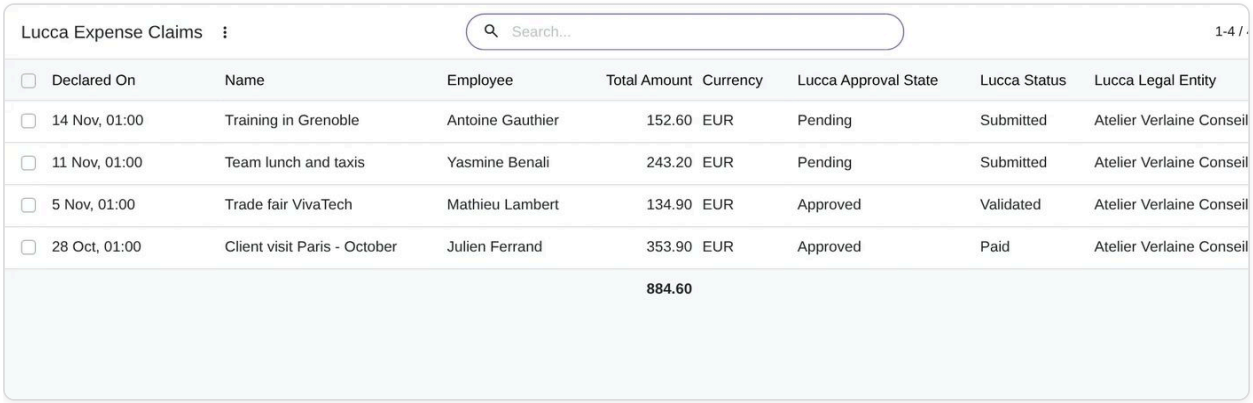
You need your own Lucca account and an API key created in it. Lucca's fees are not included. In Odoo it uses the Employees, Leaves, Expenses and Analytic apps.

Without the connector, someone re-enters in Odoo what is already in Lucca: they create employees and departments, record leave, and move expense reports into Odoo expenses. Every change has to be copied twice, and the two systems drift apart.

With the connector, Lucca remains the master system. Odoo replicates its data and can send the changes you choose back to Lucca.



- **Real Odoo 20 screenshot.** The **Lucca Employees** list with the employees pulled from Lucca.



- **Real Odoo 20 screenshot.** The **Lucca Expense Claims** list showing the status of each claim.

What changes day to day

- Employees come in from Lucca and, if you enable it, the job changes you make in Odoo go back to Lucca.

- Departments come in from Lucca.
- Lucca leave appears in Odoo as leave.
- Lucca expense claims are visible in Odoo, and from each one you can create Odoo expenses.
- Every call and notice is logged in the **Sync Log**.

What it does not do

- It does not do payroll: it keeps employees, leave and expenses up to date between Lucca and Odoo.
- It does not replace your Lucca account: you need your own and an API key.
- The published version was tested on Odoo 17, 18, 19 and 20 with simulated Lucca API responses, built from its official documentation. It has not yet been tested against a live Lucca account.

CONNECTOR GUIDE · LUCCA

2 The connector in figures

4
Odoo versions: 17, 18, 19 and 20

7
scheduled tasks, all active on installation

3 and 10
Lucca accounts on the Standard and Pro plans (each connected Odoo company is one)

14
days of trial, no card required

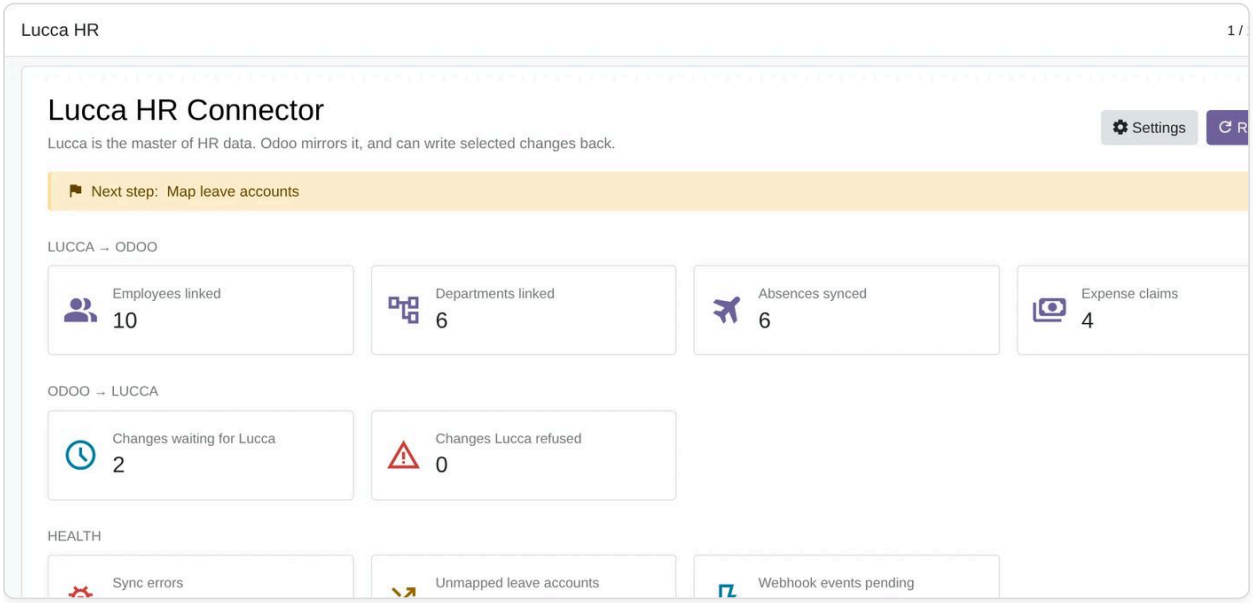
373
automated tests

13
own menus

Works on Odoo 17, 18, 19 and 20, Community and Enterprise: it only requires Employees, Leaves, Expenses and Analytic, which are Community apps, and FlexigoTech Cloud.

> How it was tested. The 373 automated tests run on Odoo 17, 18, 19 and 20 against simulated Lucca API responses, built from its official documentation. In September 2026 we tested an earlier version of the connector on Odoo 19 against a Lucca test environment provided to us: employees, departments, leave, expense claims and changes sent from Odoo all worked. In October we tested the connection through FlexigoTech Cloud there on Odoo 17, 18 and 19. The published version has not yet been tested against a live Lucca account. Two things have never actually reached Lucca: the axis sections that come from analytic accounts, because the test environment's key did not have permission to create them, and webhook notices, because that environment did not allow registering endpoints. Both are covered by simulated responses and deliveries.

3 Where you see it in your Odoo



- **Real Odoo 20 screenshot.** The **Lucca** dashboard with counters for what arrives from Lucca and what Odoo sends.

Dashboard	Opens the **Lucca Dashboard**, the app's home screen.
Employees	Lists the **Lucca Employees**.
Departments	Lists the **Lucca Departments**.
Absences	Lists the **Lucca Absences**.
Expense Claims	Lists the **Lucca Expense Claims**.
Configuration	Groups **Run Import**, **Leave Account Mapping**, **Push Analytic Accounts**, **Changes Sent to Lucca**, **Webhook Events** and **Sync Log**.

Outside its own app, the connector extends Odoo employees and departments, the **Absences** leave, Odoo expenses and analytic accounts. It also adds a **Lucca** section under **Settings**, where the account is connected.

4 What you need before you start

- Your own Lucca account and the URL of its instance, for example **https://mycompany.ilucca.net**. Each Lucca customer has its own subdomain.
- A Lucca API key generated by your Lucca administrator for this connector. Ask them to also make sure the key can see former employees, because full loads request them too.
- Odoo 17, 18, 19 or 20, on Odoo.sh or on your own server. Odoo Online will not work, because it does not allow third-party modules to be installed.
- An Odoo administrator, because only they can enter the key and manage the Lucca connection.
- The Odoo modules the connector depends on: Employees, Leaves, Expenses and Analytic Accounting, plus FlexigoTech Cloud.

> The connection to Lucca goes through FlexigoTech Cloud and requires its free 14-day trial or a subscription. Without one of the two, the connector does not talk to Lucca.

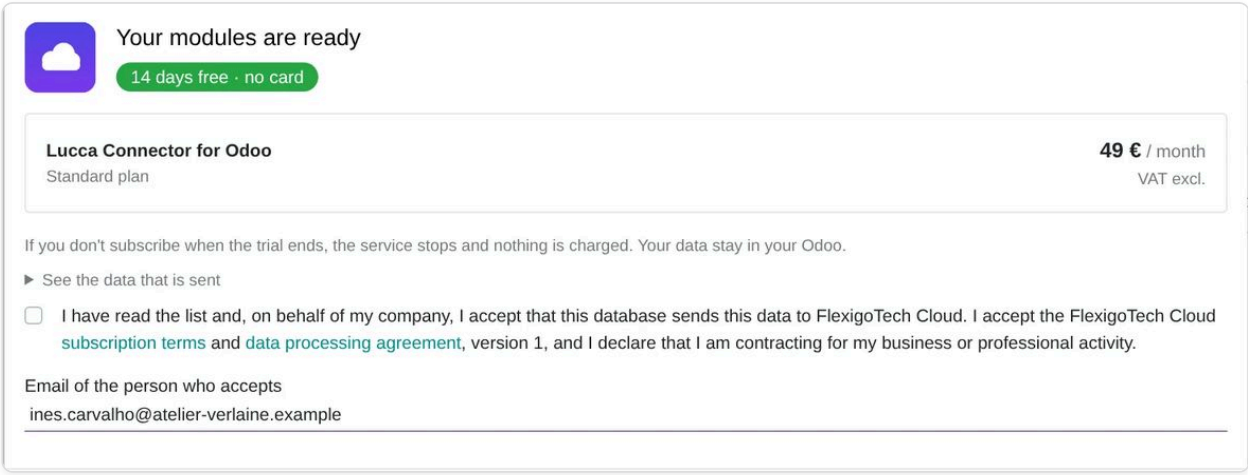
> The connector does not create the API key in your Lucca account: your Lucca administrator generates it inside their account and you paste it into Odoo. It also does not include Lucca's fees.

It is installed like any third-party Odoo module, on Odoo.sh or on your server. Odoo installs `flexigotech_cloud`, the FlexigoTech Cloud connection, together with the apps it depends on: Employees, Leaves and Expenses.

5 The FlexigoTech Cloud trial

1 **Click Activate in the top bar**

It appears as soon as you install the module. It opens the **Activate FlexigoTech Cloud** window, with the Standard plan and its 14 free days, no card required.



- **Real Odoo 20 screenshot.** The activation window: read the list of data, tick the box and enter your email.

2 **Accept and enter your email**

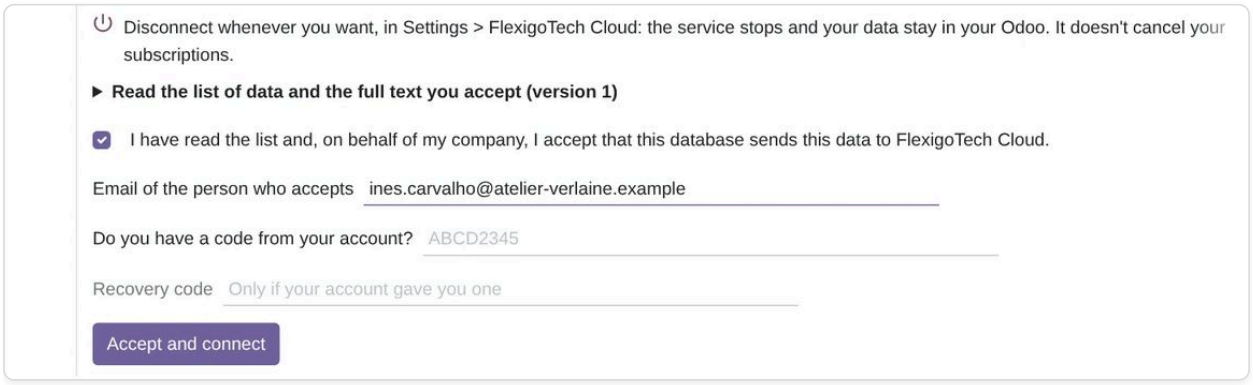
Tick «I have read the list and, on behalf of my company...» (it covers the subscription terms and the data processing agreement) and enter the **Email of the person who accepts**.

3 **Click Connect and start the trial**

Odoo registers the database with FlexigoTech Cloud and starts the module's trial. If you want to do it later, **Later** closes the window.

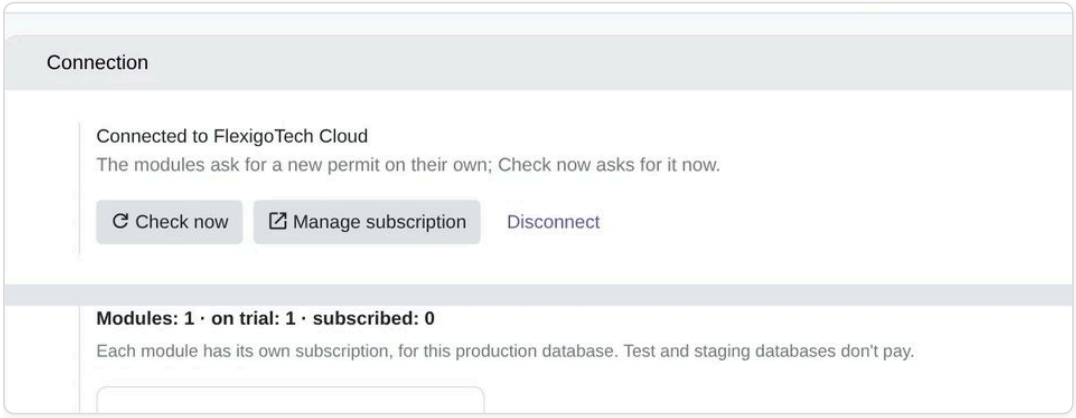
The same, from Settings

Under **Settings › FlexigoTech Cloud** is the full list of what the database sends. Tick the box, enter the email and click **Accept and connect**. If you already have an account at app.flexigotech.com, the field **Do you have a code from your account?** links the database to it (the code is valid for 10 minutes).



- **Real Odoo 20 screenshot.** Settings › FlexigoTech Cloud, before connecting: the ticked box, the email and **Accept and connect**.

With the database connected, the **Modules** block shows each module with its status: **On trial** and the date it ends, **Subscribed**, and so on. From there you choose the plan (Standard or Pro, monthly or yearly) and click **Subscribe**: payment is made on the Stripe page, outside Odoo.



- **Real Odoo 20 screenshot.** Settings › FlexigoTech Cloud with the database connected and the connector on trial.

Prices exclude VAT. If you don't subscribe when the trial ends, the service stops and nothing is charged; your data stays in your Odoo. **Check now** requests the permission immediately (otherwise the module renews it by itself every 6 hours).

> You cannot start a trial on a neutralised copy (staging) or on a database with demo data: Settings says "This database is a neutralised copy or has demo data, so trials can't be started here."

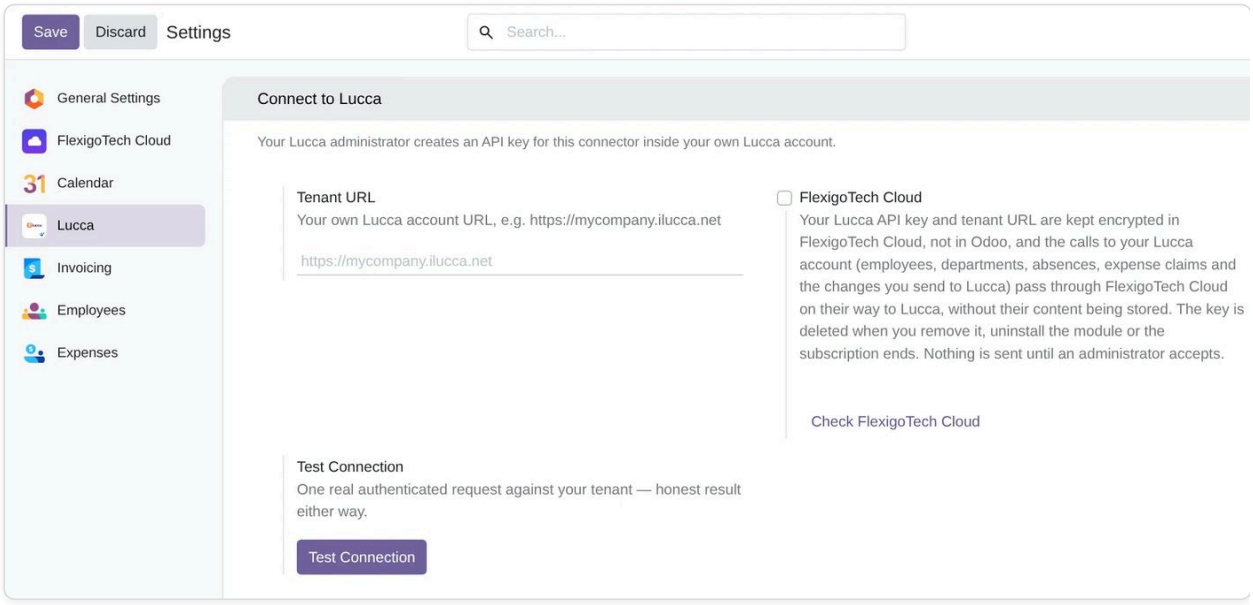
6 Connecting your Lucca instance step by step

1 **Open the Lucca settings**

Go to **Settings › Lucca**. You can also get there from the Lucca dashboard with the **Settings** button.

2 **Enter the instance URL**

In **Tenant URL** enter the root URL of your Lucca account, with https:// and no path, for example **https://mycompany.ilucca.net**. If you enter anything else, Odoo warns that the URL must be complete and without a path.



- **Real Odoo 20 screenshot.** **Settings › Lucca** freshly installed, with the **Connect to Lucca** block and the **Tenant URL**.

3 **Accept the route through FlexigoTech Cloud**

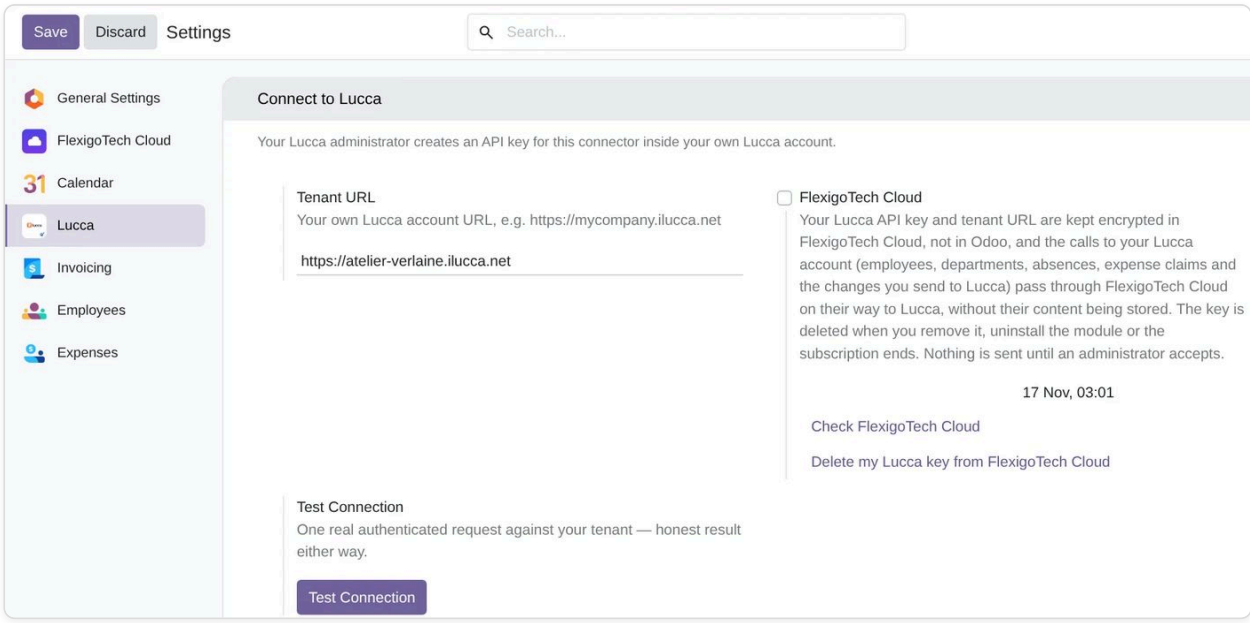
An administrator ticks **I accept that my Lucca API key and the calls to Lucca go through FlexigoTech Cloud**. Without this box, Odoo does not send the key and tells you so.

4 **Paste the API key**

In **Lucca API Key for FlexigoTech Cloud** paste the key generated by your Lucca administrator. When you save, Odoo sends it to FlexigoTech Cloud, which stores it encrypted together with the instance URL. Odoo does not store it. Without the trial or a subscription, the key is not sent.

5 **Save**

Click **Save**. **Saved in FlexigoTech Cloud** shows when FlexigoTech Cloud saved it.



- **Real Odoo 20 screenshot.** Settings › Lucca with the key saved in FlexigoTech Cloud.

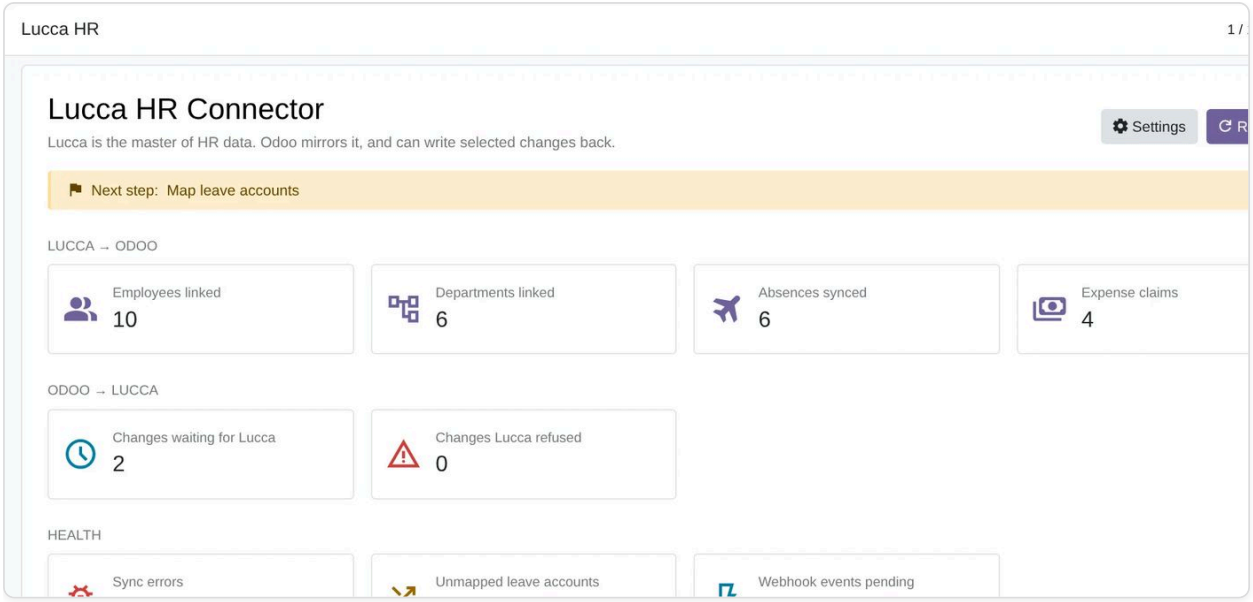
6 Test the connection

Click **Test Connection**, which makes a real call to your Lucca account and tells you the result. If it fails, check the URL and the key. **Check FlexigoTech Cloud** tells you since when the key has been stored there, without showing it.

> Only an administrator can accept the consent and enter the credentials. If you are not one, Odoo replies "Only an administrator can manage the Lucca connection."

> If you change the URL after saving the key, you must paste the key again. FlexigoTech Cloud stores your account's host together with the key, and with a different URL the calls would go to the previous account.

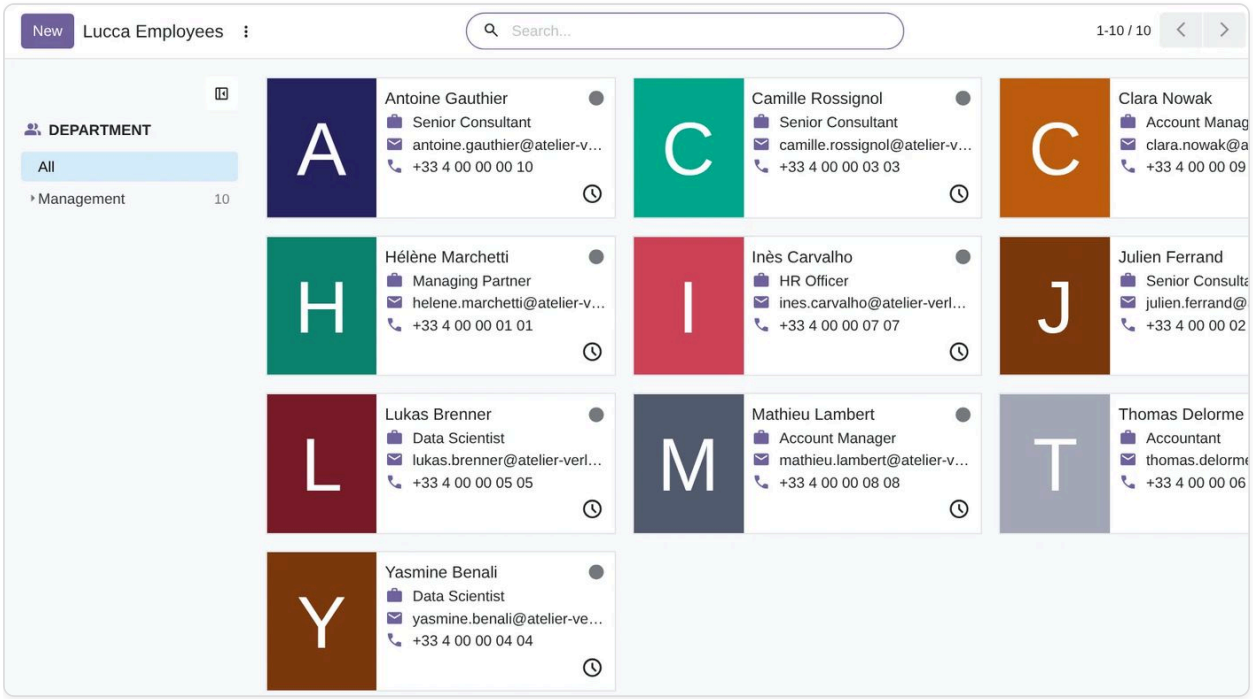
Once connected, data arrives automatically, according to what you have switched on in **What to sync**. The scheduled tasks bring in employees every 30 minutes, departments every hour, leave every 15 minutes (with a full read once a day) and expense claims every hour. Changes going to Lucca are sent as soon as you save, and webhook events are processed as soon as they arrive; their tasks, every 5 and every 15 minutes, pick up whatever is left. With **Run Import** you start the load by hand.



- **Real Odoo 20 screenshot.** The **Lucca** dashboard with the linked employees, departments and expense claims.

CONNECTOR GUIDE · LUCCA

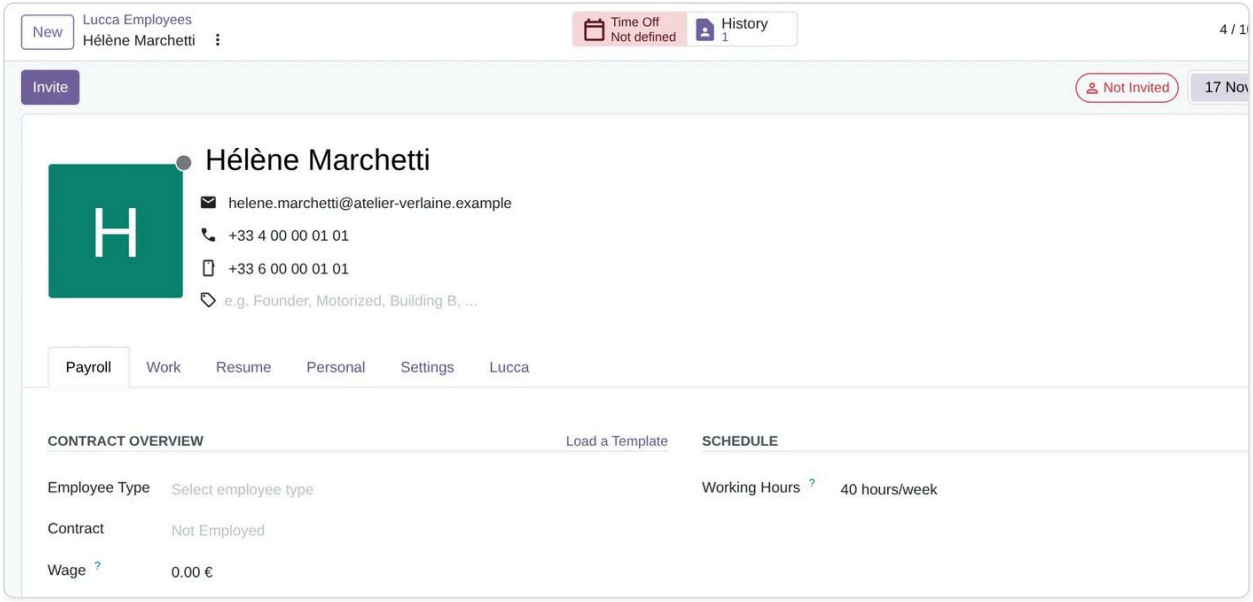
7 Lucca employees in Odoo



- **Real Odoo 20 screenshot.** The **Lucca Employees** list with the ten imported employees.

Employees are read from Lucca by the scheduled task **Lucca: sync employees**, which runs every 30 minutes and is active by default. They can also be imported with **Run Import**, from **Lucca › Configuration › Run Import**. Employees are in **Lucca › Employees**, in the **Lucca Employees** list, with the filter **Linked to Lucca**.

With **Employee changes** switched on in the settings, changes also flow the other way: the work email and phone, mobile, job title, department and manager you edit in Odoo are sent to Lucca as soon as you save, through the task **Lucca: send changes to Lucca**.



- **Real Odoo 20 screenshot.** Hélène Marchetti's record with the **Lucca** tab.

Lucca	Employee record tab with the data that comes from Lucca.
Lucca ID	The number by which Odoo recognises the person in Lucca.
Lucca Employee Number	Read only.
Lucca Legal Entity	Read only.
Lucca: Contract Start and **Lucca: Contract End**	Read-only dates.
Lucca: Last Synced	Read only.
Changes Sent to Lucca	Button on the record that opens the changes for this employee that have been sent to Lucca, or are pending sending. Only the **Manager** group of the connector can see it.

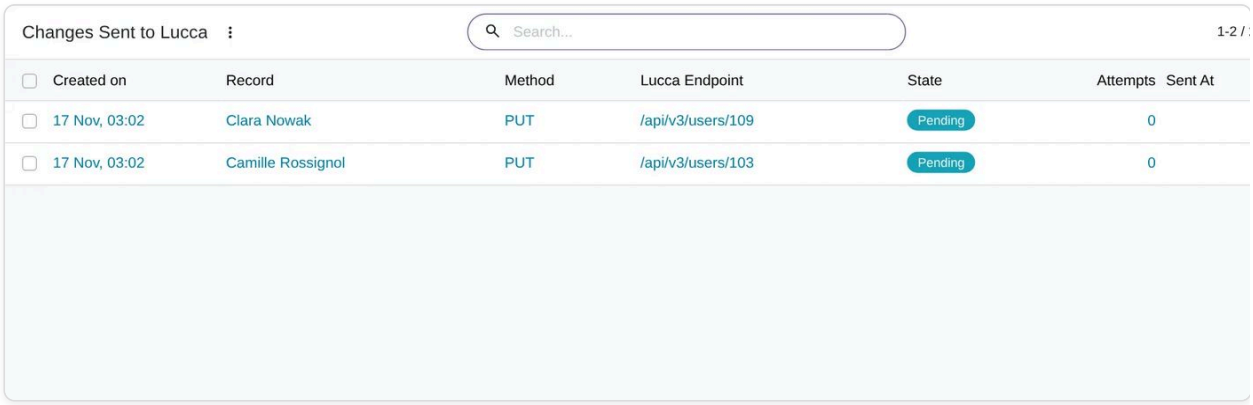
Statuses

When the contract end date in Lucca has passed, the employee is archived at the next synchronisation. If that date is cleared in Lucca, the employee is unarchived.

1. **Change a work detail**
   Edit the employee in Odoo, for example their job position or their department.

2. **Save**
   With **Employee changes** switched on, the change goes into the queue and is sent to Lucca straight away.

3. **Check the send**
   Click **Changes Sent to Lucca** on the record to see its status. If it could not be sent, the send task tries again every 5 minutes.



- **Real Odoo 20 screenshot.** The **Changes Sent to Lucca** list with two changes in Pending status.

> Employees are created in Lucca: creating one in Odoo writes nothing to Lucca. Archiving an employee in Odoo is not a leaver in Lucca either, where leavers are recorded with the contract end date.

What comes from Lucca and how it is matched

Lucca is the master system. Odoo identifies each person by their **Lucca ID**, not by the employee number, which in Lucca is only unique within an establishment. The first time, if the company has a single employee with no link yet and the same work email as in Lucca, the connector links to that employee instead of creating another. Only work details are requested for an employee. Bank details, date of birth and personal address are never requested.

Each synchronisation imports what has changed since the last one. If Lucca rejects that filter, a full read is done, and a full read also brings in former employees. Afterwards the stored leaving dates are checked, department heads are linked and, at the end, managers. The manager from Lucca takes precedence over any other. It is left empty if Lucca says the person is their own manager, that they have no manager, or that the manager has already left. It is also left empty if the manager is not visible to the connection's key.

Joiners and leavers

- Contract end date already past: the employee is archived at the next synchronisation.
- End date cleared or moved into the future: the employee is unarchived.
- Archived by hand in Odoo with no past end date: it stays archived.
- A leaver that happened since the last read is applied even if Lucca does not send that person again.

Archiving this way does not start the Odoo leaving process. The end date stays in **Lucca: Contract End**.

What is sent to Lucca and when

Only an employee already linked to Lucca is sent, and only if their company has **Employee changes** switched on. The work email and phone, mobile, job position, department and manager are sent. Odoo compares the values before and after saving and sends only what changed, one send per employee. A value you clear is sent as empty. If the department or the manager does not exist in Lucca, that field is left out and a note is recorded. Changes made by the import itself are not sent back.

Job position and department live in the employee's version. A new version effective today is sent like any other edit. A version with a future date is sent when it becomes the current one.

Record fields and tabs

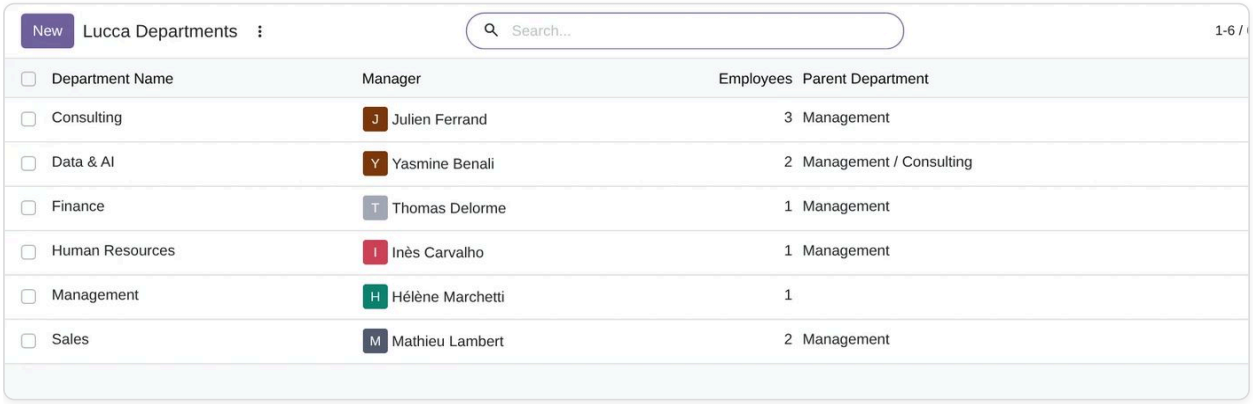
The **Lucca** tab on the record is visible to HR managers, and the **Changes Sent to Lucca** button only to the connector's **Manager** group. All Lucca fields are read-only, except **Lucca ID**.

End-to-end example

After the import, the **Lucca Employees** list shows ten employees, such as Antoine Gauthier, senior consultant, or Hélène Marchetti, managing partner. You change Clara Nowak's job position in Odoo and save. In **Lucca › Configuration › Changes Sent to Lucca** her change appears as **Pending** with 0 attempts until it is sent. The send task sends it straight away and the change moves to **Sent**. The last synchronisation date is shown on the **Lucca** tab of the record.

CONNECTOR GUIDE · LUCCA

8 Lucca departments in Odoo



- **Real Odoo 20 screenshot.** The **Lucca Departments** list with name, manager, employees and parent department.

Where the data comes from

Departments are pulled from Lucca by the scheduled task **Lucca: sync departments**, which runs every hour and is active. You see them under **Lucca › Departments**, which opens the **Lucca Departments** list. To pull them without waiting for the task, run the import from **Lucca › Configuration › Run Import**.

Name	The name of the department.
Manager	The employee of the same company whose Lucca user is the department's head.
Employees	The employees who belong to the department.
Parent department	The department it reports to, according to the Lucca hierarchy.
Lucca ID	The department's identifier in Lucca.
Lucca Head (User ID)	The identifier of the Lucca user who leads the department.

How the parent department and manager are filled in

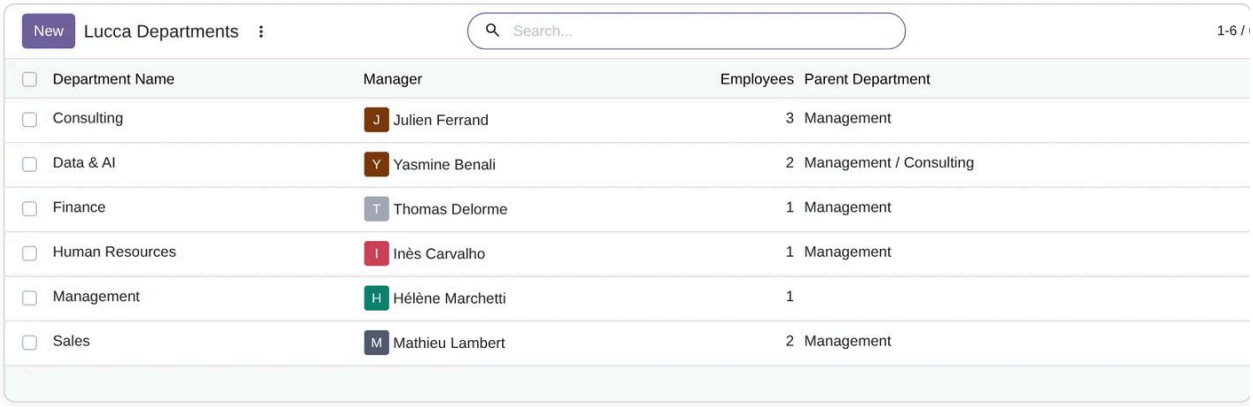
1. **Import the employees**
   The manager is only assigned if their employee is already in Odoo.

2 **Wait for the hourly task or run the import**

Lucca sends the hierarchy and Odoo places each department under its parent.

3 **Open Lucca › Departments**

Check the parent and manager of each department.



- **Real Odoo 20 screenshot.** Each row shows manager, employees and parent department.

> If the head Lucca names is not imported into Odoo, the manager does not change. A department that Lucca moves to the top level loses the parent that came from Lucca. A parent chosen in Odoo from departments that are not from Lucca is kept. Assigning the manager from Lucca does not move employees under the new manager.

The hierarchy and the manager follow fixed rules. At the end of the chapter there is an example using the data of a company already imported.

How each department is placed in the hierarchy

For each department, Lucca gives the path of its parent. Odoo finds the department with that path and sets it as the parent. If Lucca gives no parent or it matches no known department, Odoo uses the department's own path without its last segment to find the parent. Path segments are never interpreted separately.

- A department that Lucca moves to the top level loses the parent it had received from Lucca.
- A parent chosen in Odoo from departments that are not from Lucca is kept.
- If a Lucca server refuses to provide the head or the parent, the departments are still synchronised with their name, path and archived status.

How the manager is assigned

The manager is the active employee of the same company whose Lucca identifier matches the department's head. It is assigned at the end of the department synchronisation and also at the end of the employee synchronisation, so a manager not yet imported is assigned once they arrive. If the head does not exist in Odoo, because they have not been imported, belong to another company or have already left, the manager stays as it was.

> Assigning the manager from Lucca does not change the head of any employee linked to Lucca. When Odoo changes a department's manager it usually moves its members under the new manager. With a manager coming from Lucca it does not: each user's manager comes from Lucca.

Record fields

Lucca ID	The department's identifier in Lucca.
Lucca Head (User ID)	The Lucca user who leads the department. The **Manager** in Odoo comes from them. It is read-only.

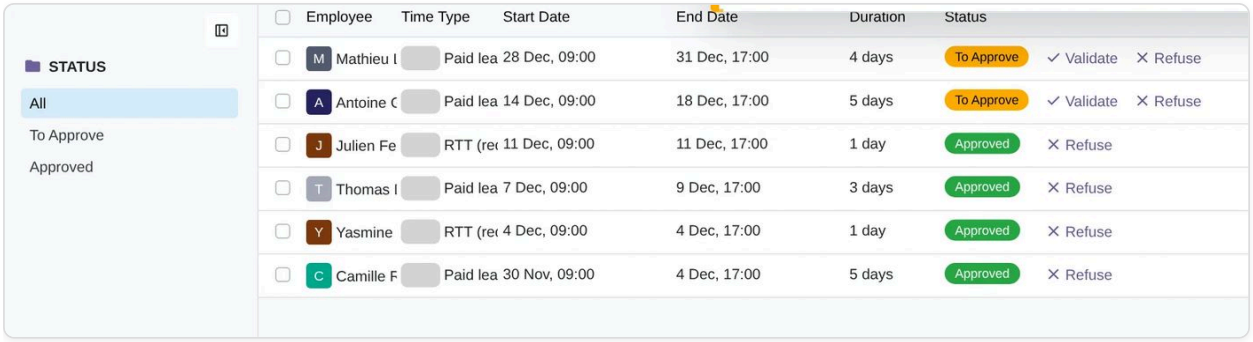
The Lucca hierarchy path is stored, but it does not appear on any screen of the module.

End-to-end example

In the sample company, 10 employees and 6 departments have been imported. Management is the top-level department, and Sales, Consulting, Finance and Human Resources all report to it. Data & AI reports to Consulting: the parent column shows "Management / Consulting". Hélène Marchetti is the manager of Management, and among the others, Mathieu Lambert manages Sales, Julien Ferrand manages Consulting and Inès Carvalho manages Human Resources. If you import the departments before the employees, the managers are assigned when the employee import finishes.

CONNECTOR GUIDE · LUCCA

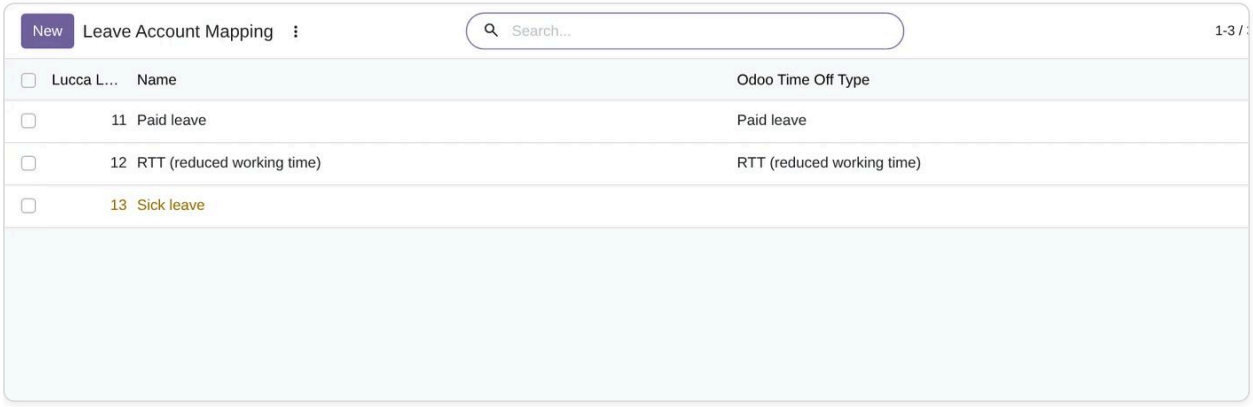
9 From Lucca absence to Odoo absence



- **Real Odoo 20 screenshot.** The **Lucca Absences** list with the imported requests and their statuses.

With **Absences** switched on in the settings, absences arrive from Lucca through the task **Lucca: sync absences**, every 15 minutes, and **Lucca: re-read all absences (nightly)** goes over everything once a day. You can also bring them in straight away with **Run Import**. You find them under **Lucca › Absences**, in the **Lucca Absences** list.

Leave account mapping



- **Real Odoo 20 screenshot.** The **Leave Account Mapping** list with Lucca accounts and their Odoo time off type.

Lucca Leave Account ID	The account's identifier in Lucca.
Name	The account's name in Lucca.
Odoo Time Off Type	The type under which absences from that account are imported. If you leave it empty, they are skipped.

The mapping is in **Lucca › Configuration › Leave Account Mapping**. Each account appears in the list the first time the synchronisation sees it in an absence. The **Unmapped** filter shows those that still have no time off type.

What you will see on each absence

On the leave form there is a **Lucca** group with **Lucca Leave Period ID**, **Lucca Leave Account** and **Lucca: Last Synced**, which is read-only.

How they are imported

- An absence confirmed in Lucca is approved in Odoo.
- An absence cancelled in Lucca is cancelled in Odoo.
- An unconfirmed absence is only imported, as pending approval, if the company has pending import switched on.
- When Odoo rejects an absence, the synchronisation log stores the reason Odoo gives, for example a time off type that needs an allocation.

> An absence is not imported if its account has no Odoo time off type, if its holder is not linked to an employee of this company or if it errors. This is noted in the log, and you choose the type in the mapping.

What is synchronised and within which window

Lucca reports each half day separately. The connector groups them by leave period and creates a single Odoo request per period. There is no call per employee: a full read requests everything at once.

Reading starts from a configurable number of days back and runs up to one year ahead. An absence whose start falls before that window is read again in full, so that its start date does not move each day.

The 15-minute read asks only for what changed since the last synchronisation. The nightly one reads everything. The starting point is rounded to the minute and one minute is subtracted, so at most some extra data is re-read and no change is lost.

What each result means

Approved	An absence confirmed in Lucca is approved in Odoo. Odoo creates the absence in the employee's work calendar.
Cancelled	An absence cancelled in Lucca is voided in Odoo from any live status. It is removed from the calendar and managers are not notified. If it never reached Odoo, nothing is created.
To approve	An unconfirmed absence is imported this way only if **Import Pending Absences** is switched on. Otherwise it is skipped. An absence already approved in Odoo is not reverted by unconfirmed data.

Skipped	It is not imported if its holder is not linked to an employee of the company, if its account has no time off type or if that period has an error. That period is rolled back and noted in the log.
Up to date	If nothing has changed, the request is left untouched.

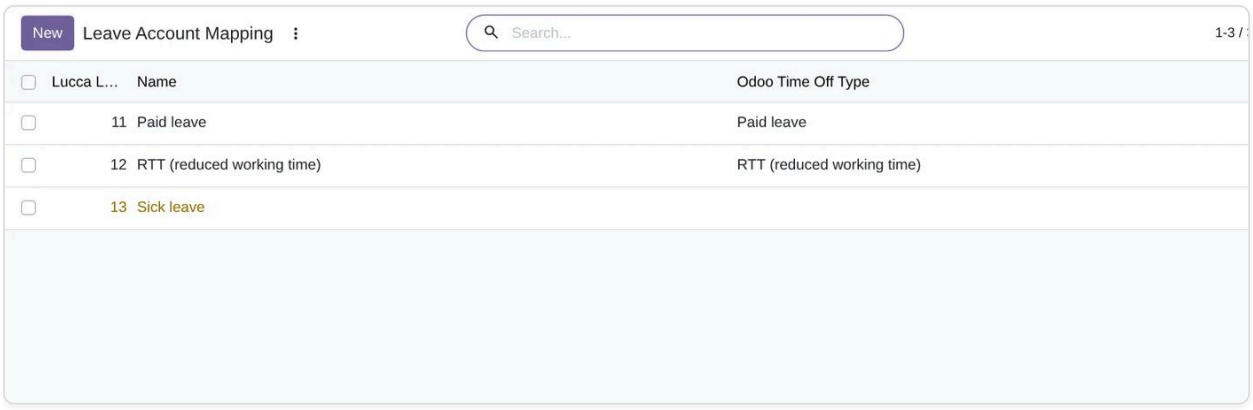
Rules and conditions

- An absence is only imported if its account has an **Odoo Time Off Type**. The account is registered automatically when first seen, and you choose the type.
- Only the employees of the current company are touched, even if the same Lucca identifiers are linked in another company.
- A request is only linked to the employee holding it in that company.
- If the dates of an approved absence change, the connector restores its associated records so that it does not disappear from the employee's schedule.
- A single half day is imported as a half day if the **Odoo Time Off Type** is requested in half days. Otherwise it is entered as a full day, because that type does not allow half days. A value is only written if it has changed.
- Changes made by the synchronisation are not sent back to Lucca, and notification emails are queued rather than sent one by one.

Record fields

Lucca Leave Period ID	The period number in Lucca to which the request corresponds.
Lucca Leave Account	The Lucca account it comes from, linked to the mapping.
Lucca: Last Synced	The last time the absence sync changed this request.

End-to-end example



- **Real Odoo 20 screenshot.** The Lucca leave accounts with their Odoo Time Off Type, if they already have one.

1 Open the mapping

Go to **Lucca › Configuration › Leave Account Mapping**. The accounts the sync has seen are listed there. In the sample company there are three: Sick leave (13), RTT (reduced working time) (12) and Paid leave (11).

2 Choose a type for each account

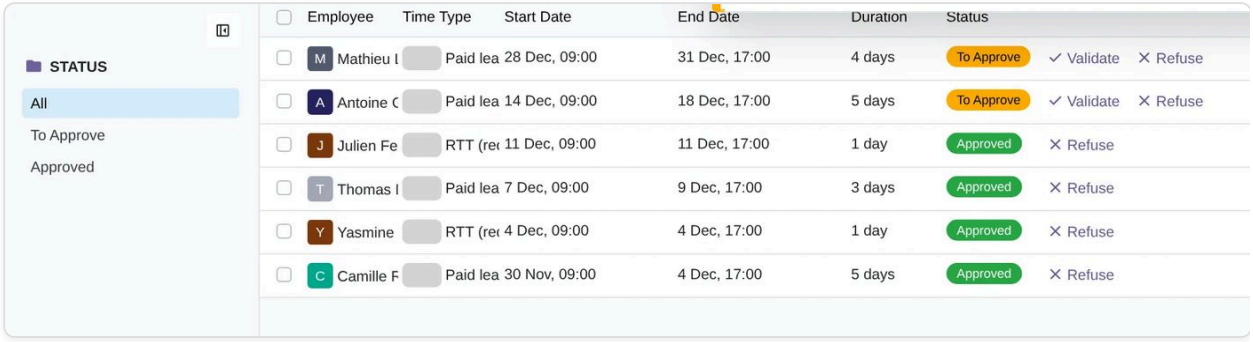
In **Odoo Time Off Type**, set the type that matches each one. An account without a type leaves its absences out, and the dashboard reminds you of this.

3 **Wait for the job or run the import**

The **Lucca: sync absences** job runs every 15 minutes.

4 **Check the result**

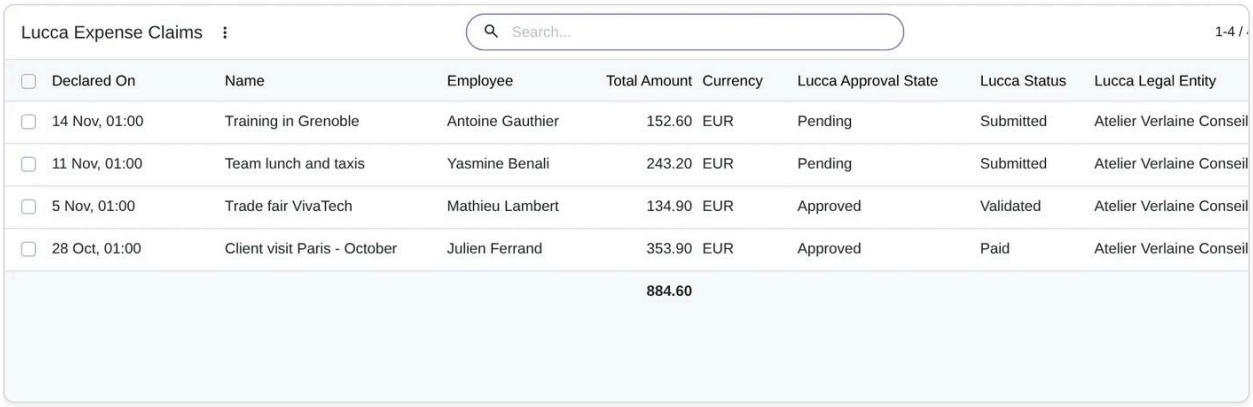
In **Lucca › Absences** you can see each absence with its employee, dates, duration and status. If one is missing, check the **Sync Log**.



- **Real Odoo 20 screenshot.** The absences that come from Lucca, with their employee, dates, duration and status.

CONNECTOR GUIDE · LUCCA

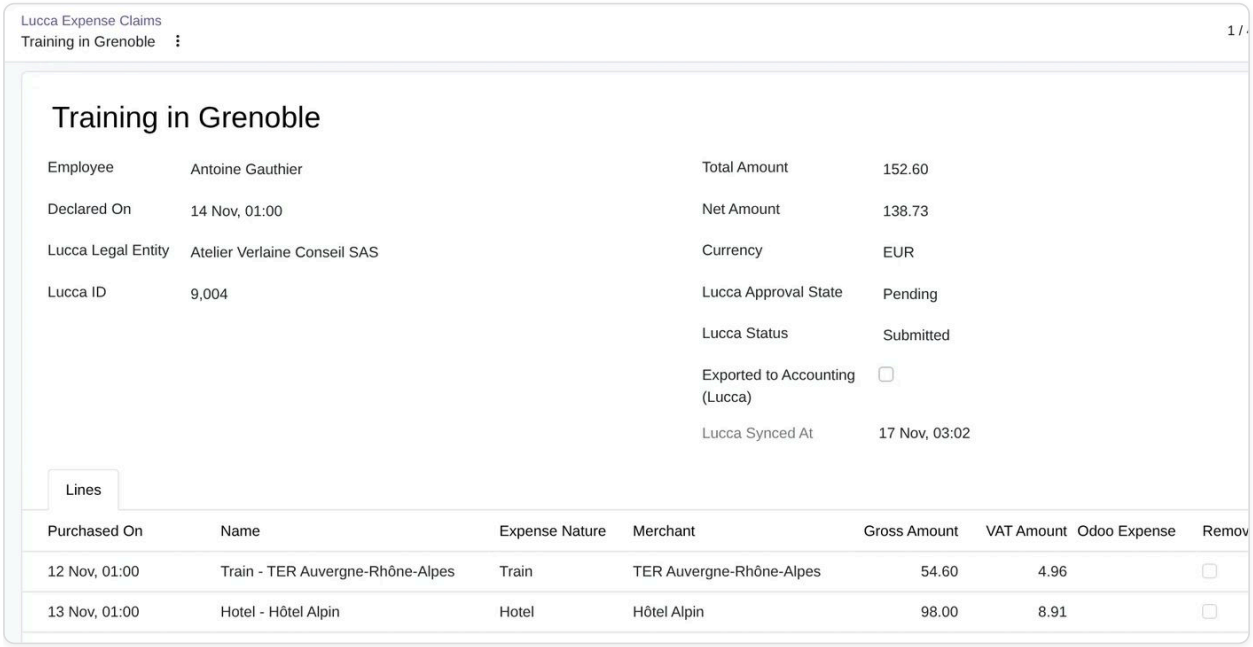
10 Lucca expense claims in Odoo



- **Real Odoo 20 screenshot.** The **Lucca Expense Claims** list with its approval status and its status in Lucca.

Enter through **Lucca › Expense Claims**. With **Expense Claims** switched on in the settings, they arrive from Lucca through the scheduled job **Lucca: sync expense claims**, which runs every hour. Data only flows from Lucca to Odoo: what is read from Lucca is not sent back to Lucca.

The record of a claim



- **Real Odoo 20 screenshot.** The claim "Training in Grenoble" with its **Lines** tab.

Create Odoo Expenses	Creates a draft expense for each line that does not yet have one, and shows you the expenses of the claim. Only the Lucca managers group can see it.
Lines	The lines of the claim: purchase date, name, expense nature, merchant, gross amount, VAT amount and Odoo expense.
Expense Count	How many Odoo expenses come from the claim.
Lucca Synced At	When the claim was last read. It is not editable.

In the list you can filter by **Approved** and **Paid**, and group by **Employee** or by **State**.

States

Each claim shows two pieces of data that come from Lucca: **Lucca Approval State** and **Lucca Status**. Odoo does not change them.

Turning a claim into Odoo expenses

1. **Open the claim**
   Go to **Lucca › Expense Claims** and click the claim.

2. **Click Create Odoo Expenses**
   A draft expense is created for each line without one, and the list of expenses of the claim opens.

> Expenses are never created on their own: you always create them with the button. If the claim has no linked Odoo employee, the button gives an error: sync the employees first.

What is brought in and from which dates

The scheduled job **Lucca: sync expense claims** reads the claims declared within the number of days set in **Expense window (days back)** in **Settings › Lucca**. If that setting is empty or not positive, the last 365 days are used. It then fetches the lines of the claims read, in batches and without one query per claim. A claim with bad data is skipped and the others carry on. If reading the lines of a batch fails, those claims keep the lines they already had.

Claim states

Lucca Approval State	The approval state Lucca gives the claim. The screens show, for example, Pending and Approved.
Lucca Status	The state of the claim in the Lucca workflow. The screens show, for example, Submitted, Validated and Paid.
Exported to Accounting (Lucca)	Checkbox that shows whether Lucca has already exported the claim to accounting.

Rules when creating expenses

- The **Create Odoo Expenses** button creates a draft expense for each line that does not yet have an Odoo expense.
- The expense uses the gross amount of the line in its own currency. If that currency is not active in Odoo, it uses the company currency.
- The expense is created without tax: its claim notes the VAT according to Lucca, so you can choose the tax before posting.
- If the claim has no linked Odoo employee, Odoo replies that the expense claim has no linked Odoo employee and that you should sync the employees first. The claim name and that warning appear: "The expense claim "..." has no linked Odoo employee".

If a line disappears in Lucca

When a line no longer exists in Lucca, Odoo deletes it from the claim. If an Odoo expense was already created from that line, the line is kept and marked in the **Removed in Lucca** column.

Record fields

Employee	The Odoo employee linked to the claim.
Declared On	Date on which the claim was declared.
Lucca Legal Entity	The Lucca legal entity to which the claim belongs.
Lucca ID	Identifier of the claim in Lucca.
Total Amount and **Net Amount**	The amounts of the claim.
Currency	Defaults to EUR.
Lines	Tab with purchase date, name, expense nature, merchant, gross amount, VAT amount and Odoo expense for each line.

End-to-end example

The claim "Training in Grenoble" belongs to Antoine Gauthier and totals 152.60 EUR. It has two lines: a train, TER Auvergne-Rhône-Alpes, and a hotel, Hôtel Alpin. The claim is in Pending and Submitted, and the list shows four claims. You open the claim and click **Create Odoo Expenses**. Two draft expenses are created, one per line, and their list opens.

CONNECTOR GUIDE · LUCCA

11 Analytic accounts to Lucca

The push goes from Odoo to Lucca and requires **Analytic accounts → Lucca axis** switched on in the settings, with an **Odoo plan** and a **Lucca Axis ID**. When you create, rename, archive or delete an account in that plan, the change goes into the **Changes Sent to Lucca** queue and leaves for Lucca straight away. For accounts that already existed, use the initial load.

Initial load

The menu **Lucca › Configuration › Push Analytic Accounts** opens the **Push Analytic Accounts to Lucca** wizard.

Load Axes from Lucca	Asks your Lucca instance for the list of analytic axes. It is only visible when choosing the axis.
Lucca Axis ID	The axis that will receive the sections. It is filled in from **Settings** > **Lucca**.
Analytic Accounts	The accounts that are sent. They are filled in with those of the analytic plan chosen in the settings.
Push to Lucca	Queues one change per account. Only visible in the preview.
Open Changes Sent to Lucca	Opens the list of queued changes. Only visible at the end.
Close	Closes the wizard.

The wizard steps

The wizard has three steps: you load the Lucca axes and choose one, you review the accounts to be sent, and at the end you see the summary of the result.

1. **Open the menu**
   Go to **Lucca › Configuration › Push Analytic Accounts**.
2. **Click Load Axes from Lucca**
   The axes of your instance are read.
3. **Choose the axis and review the accounts**
   You move on to the preview.
4. **Click Push to Lucca**
   The changes are queued and leave for Lucca straight away.

> Nothing is sent if the instance URL or the API key is missing, if there is no axis, if you do not specify the axis or if you choose no accounts. Deleting an account in Odoo does not delete the section in Lucca: it deactivates it.

> How it was tested: the analytic account push has only been tested against simulated responses from the Lucca API. In the test environment Lucca gave us, the key did not have permission to create sections and Lucca answered with a 401 error, so we have not yet seen a section arrive in a Lucca account. Your API key needs that permission.

When an account is mirrored

Mirroring works when the company has **Analytic accounts → Lucca axis** switched on, with a **Lucca Axis ID** and an **Odoo plan**. Then each account in that plan is a section of that axis. Everything goes through the queue of changes to Lucca, never directly.

What change each action generates

IN ODOO	IN LUCCA
Create an account	The section is created and Odoo stores the ID that Lucca returns and the sync date.
Change name or code, archive or unarchive	The section is updated. If it does not yet exist in Lucca, the change is merged into the pending push.
Move an account into the plan	The section is created or updated.
Remove an account from the plan, or delete it	The section is deactivated. It is never deleted, so expenses already posted keep a valid section.

An archived account that never reached Lucca is not sent, except on creation, in the initial load, or if a send is already pending (so it arrives deactivated).

Rules and conditions

- A person who does not work for the account's company does not send changes to that company's Lucca. A shared account, assigned to another company, can be edited from several companies, but only someone working for the owning company writes to Lucca.
- Only the connector decides which Lucca account each analytic account maps to, and this cannot be changed by hand.
- If a send fails, those following the account are notified.

Fields on the Lucca tab

Lucca Axis Section ID	The section number in Lucca. Read only.
Lucca: Last Synced	Date and time of the last confirmed send. Read only.
Lucca Tenant of Company	The company whose Lucca tenant contains the section that mirrors the account. Read only.

End-to-end example

1. **Create the account**
   You create an account in the chosen analytic plan. A change for Lucca is queued.

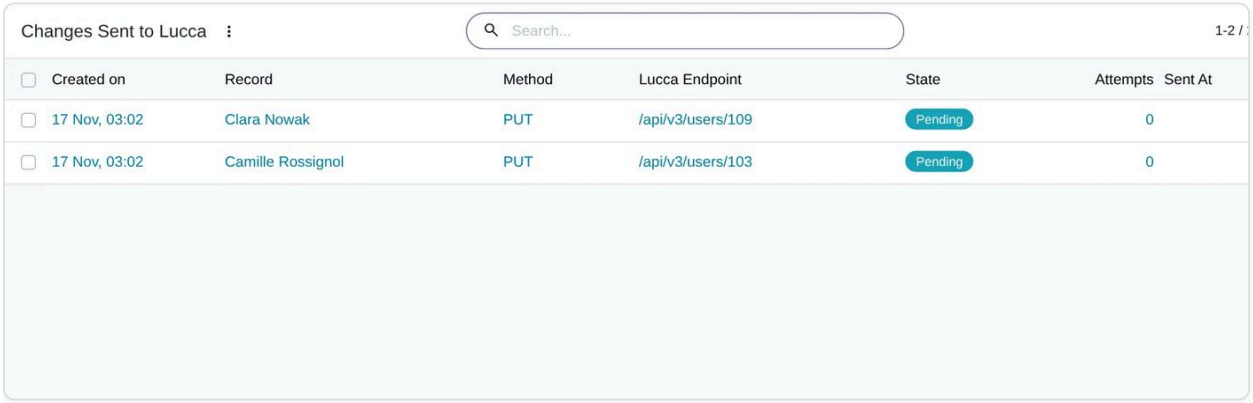
2. **Wait for the send**
   It goes out straight away with **Lucca: send changes to Lucca**. You can see it in **Lucca › Configuration › Changes Sent to Lucca**.

3. **Check the result**
   The **Lucca** tab of the account shows the section ID and the sync date.

4. **Archive the account**
   The section is deactivated in Lucca. It is not deleted.

CONNECTOR GUIDE · LUCCA

12 Changes sent, webhooks and log



- **Real Odoo 20 screenshot.** The **Changes Sent to Lucca** list with two **Pending** changes.

The changes Odoo sends to Lucca appear in **Lucca › Configuration › Changes Sent to Lucca**. They go out as soon as they are saved, and the scheduled action **Lucca: send changes to Lucca** clears the queue every 5 minutes.

Notifications coming from Lucca are in **Lucca › Configuration › Webhook Events**. Each notification triggers the scheduled action **Lucca: process webhook events**, which processes it straight away and also goes through the list every 15 minutes.

Every call to Lucca and every run of a sync is recorded in **Lucca › Configuration › Sync Log**. The connector writes the records; you consult them.

Record buttons

Retry (sent change)	Appears only if the change is **Failed**. Puts it back in the queue.
Open Odoo Record	Opens the Odoo record the change refers to.
Retry (webhook event)	Appears only if the event is in **Error**. Queues it again for processing.

Statuses

Sent change	**Pending**, **Sent** or **Failed**.
Webhook event	**Received**, **Processed**, **Duplicate**, **Ignored** or **Error**.
Sync log	**Done** or **Error**.

> These lists cannot be used to create or correct data. You cannot manually change the content of a change or an event. Resolved Lucca notifications and sent changes are deleted after 90 days. Those in error or in the queue are kept until someone reviews them.

What happens to a change made in Odoo

When you change something in Odoo that must reach Lucca, the change is saved as a pending send in the same operation. If Odoo rolls back that operation, Lucca receives nothing. If Lucca does not respond at that moment, the send stays pending and is retried. Several changes in a row on the same record and the same Lucca direction are merged into a single send.

Statuses of a sent change

Pending	It is in the queue and has not gone out yet. This is the status it is created with.
Sent	Lucca has accepted it. **Sent At** is filled in.
Failed	The send returned an error. **Error Message** gives the reason and **Attempts** counts the tries. Use **Retry** to put it back in the queue.

Statuses of a webhook event

Received	Saved and waiting to be processed. This is the initial status.
Processed	The Lucca change has been applied in Odoo.
Duplicate	Lucca sent it more than once. Only the first one is processed.
Ignored	It could not be read or does not need to be applied.
Error	It could not be applied. **Retry** puts it back in the queue.

Rules and messages

- If the Lucca instance is not configured, the event fails with «Lucca is not configured for company "...": set the tenant URL and API key in Settings > Lucca.».
- If the notification does not contain a valid employee number, you get "The event does not contain any usable Lucca employee ID (...)."
- If Lucca returns no employee data, you get "Lucca returned no data for user ...."
- If the ID or the name is missing, you get "Lucca user ... could not be imported (missing id or name).".
- If another notice in the same run has already triggered the synchronisation and it failed, you get "Covered by a synchronisation earlier in this run, which failed: ...".

Lucca sends notifications for employees and departments, among others, but not for absences or expense reports. Those two continue to sync with their scheduled actions.

Fields and tabs of the record

Kind	API call, connection test, scheduled action, webhook, manual run or sent to Lucca.
Records and **Error Count**	Rows processed and errors in that run.
Status Code	Code returned by Lucca.
Source, **Topic** and **Resource Type**	For an event, the Lucca account that emitted it, the notification topic and the resource it refers to.
Data sent, **Payload** and **Error**	Form groups: what was sent (for a change), the notification content (for an event) and the reason for the failure, which only appears if there was one.

To find something, the filters on each list are **Pending**, **Failed** and **Sent** for sends; **Queued** and **Errors** for notifications; and **Errors**, **Scheduled** and **Webhooks** for the log.

End-to-end example

After you change Clara Nowak and Camille Rossignol in Odoo, **Changes Sent to Lucca** shows two **Pending** rows with method **PUT** and **Attempts** at 0, at /api/v3/users/109 and /api/v3/users/103. On the **Lucca** dashboard, the counter "Changes waiting for Lucca" shows 2. The send task sends them straight away. If Lucca accepts them, they move to **Sent**; if one fails, it stays in **Failed** with its reason and you retry it with **Retry**. In **Sync Log** the runs appear as **Done**, with 0 errors.

CONNECTOR GUIDE · LUCCA

13 Webhooks: Lucca changes in real time

Without webhooks, Lucca changes arrive with the scheduled actions: employees every 30 minutes and departments every hour. With webhooks, Lucca notifies your Odoo when an employee or a department changes, and Odoo picks up the change straight away. Lucca does not send notifications for absences or expense reports, which continue to arrive with their scheduled actions.

How they are declared

1. **Save the settings once**

   In **Settings › Lucca**, the **Webhooks** block shows the **Webhook URL** after you save the settings for the first time. The URL contains a secret specific to your company, so do not publish it.

2. **Declare it in Lucca**

   A Lucca administrator declares it in their account, at /integrations/endpoints, for the employee.* and department.* events. When declared, Lucca sends a test code to that URL and Odoo returns it, which is what Lucca requires to activate it.

3. **Paste the signing secret**

   Lucca shows a secret for that endpoint. Paste it into **Signing secret** and click **Save**. Without this secret, Odoo rejects all deliveries.

What Odoo checks in each notification

- Lucca signs each notification with that secret, and Odoo rejects any that arrives unsigned or with a signature that does not match.
- It also rejects a notification whose timestamp is more than 5 minutes away from the current time.
- Lucca may send the same notification more than once. The repeat is marked **Duplicate** and is not applied twice.
- Lucca waits at most 3 seconds for the response. Odoo saves the notification, replies immediately and processes it right after.

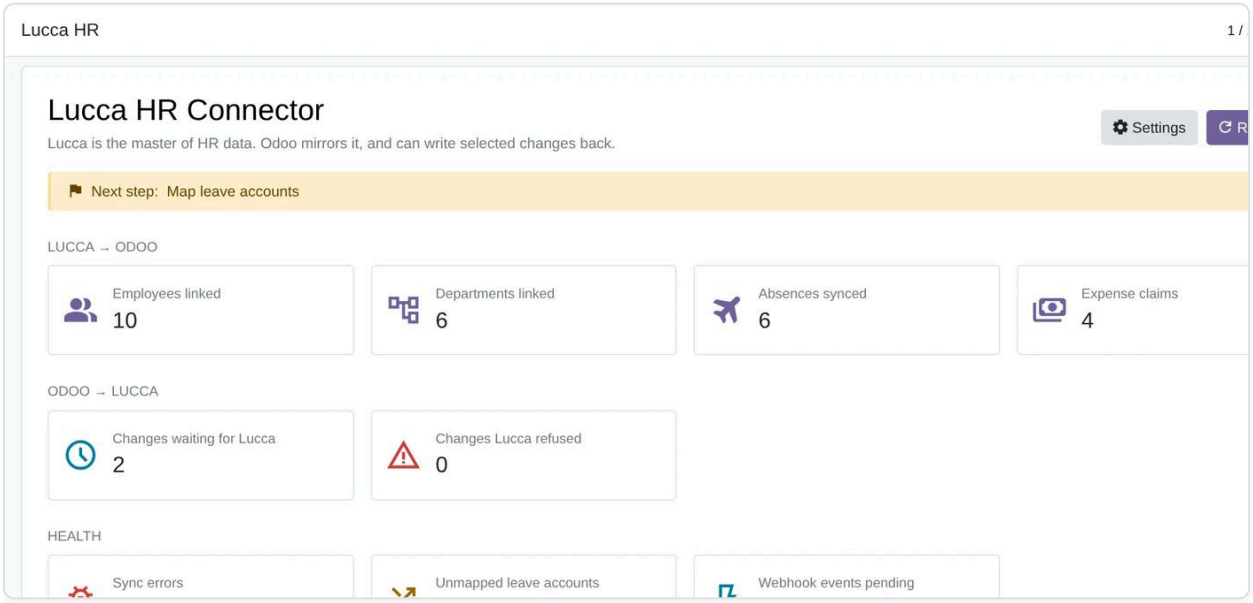
What Odoo does with each notification

LUCCA NOTIFICATION	WHAT ODOO DOES
employee.created and employee.updated	Reads that employee from Lucca again and updates it in Odoo.
employee.deleted	Archives the linked employee.
employment.* and job-position.*	Starts an employee sync with what changed, one per company even if many notifications arrive at once.
department.*	It triggers a department sync, also a single one.
Any other	It saves it as **Ignored**, so there is a record of it.

With each notice, Odoo knows what has changed and reads the current state from Lucca, so a repeated or late notice does not write out-of-date data. Each notice is kept in **Lucca › Configuration › Webhook Events** with its status, and one in **Error** goes back to the queue with **Retry**. Resolved notices are deleted after 90 days.

> How it has been tested: with simulated deliveries that follow Lucca's webhook specification, in the automated tests and, in September 2026, with an earlier version of the connector on Odoo 19 connected to the test environment Lucca provided. That environment did not allow registering endpoints, so no notice sent by Lucca has reached Odoo yet. If your account does not allow it either, the scheduled tasks still bring everything in.

14 Initial import and dashboard



- **Real Odoo 20 screenshot.** The dashboard with the linked data and the next step, **Map leave accounts**.

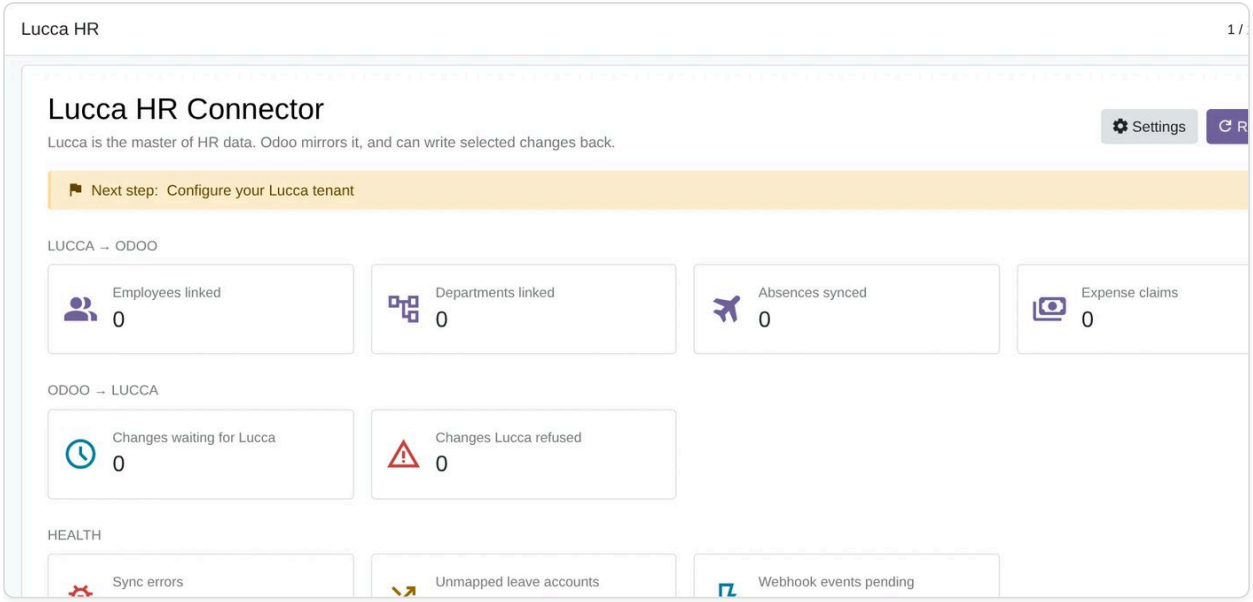
Data reaches Odoo in two ways. The scheduled tasks bring it in automatically: **Lucca: sync employees** every 30 minutes, **Lucca: sync departments** every hour, **Lucca: sync absences** every 15 minutes and **Lucca: sync expense claims** every hour. The other way is the **Run Import** button, which starts the import immediately.

The dashboard buttons

Settings	Opens the connector settings. Only the Lucca managers group can see it.
Run Import	Opens the **Lucca Import** window. It is also under **Lucca › Configuration › Run Import**. Only the Lucca managers group can see it.

The next step

The dashboard tells you what to do now. There are four states: **Configure your Lucca tenant**, **Run your first employee sync**, **Map leave accounts** and **Ready**.



- **Real Odoo 20 screenshot.** The freshly installed dashboard, with the **Configure your Lucca tenant** step and everything at zero.

Running an import

1. **Click Run Import**

   The **Lucca Import** window opens.

2. **Select what you want to bring in**

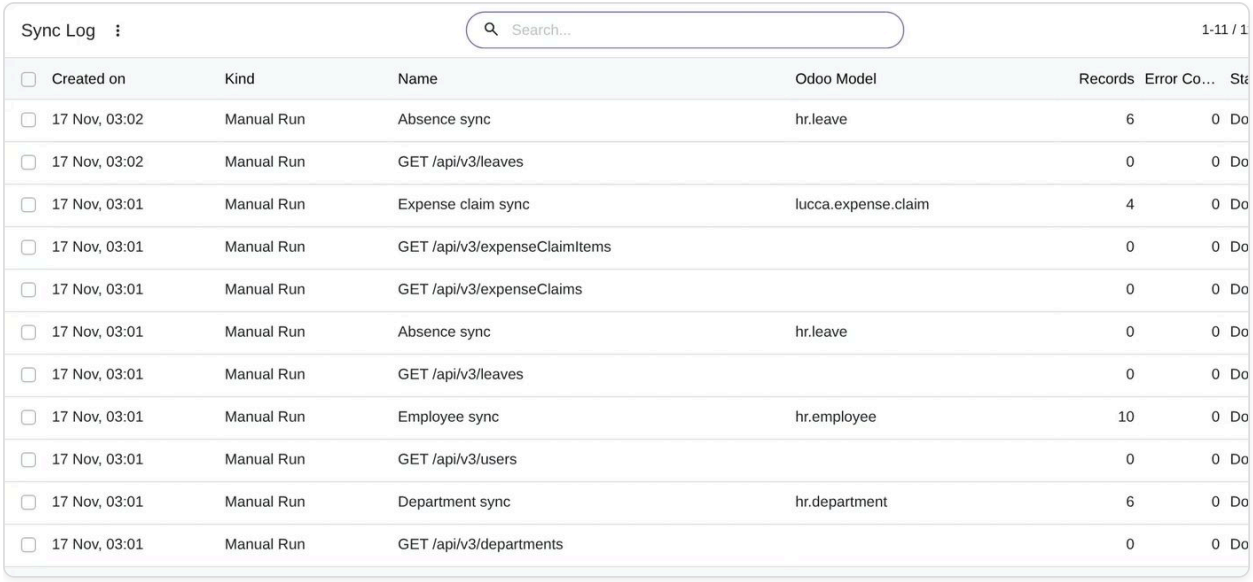
   The checkboxes are **Departments**, **Employees**, **Absences** and **Expense Claims**. They are ticked according to what you have enabled in the settings.

3. **Click Run Sync**

   When it finishes, the result appears in the **Result** group. Click **Close** to leave.

> If you have not configured the instance URL and the API key, you get "Configure the Lucca tenant URL and API key first (Settings > Lucca).". If you tick no box, you get "Pick at least one thing to sync.".

The **Lucca Import** window does by hand what the scheduled tasks do: it uses the same synchronisation code and leaves the same entries in **Lucca › Configuration › Sync Log**, where the runs appear as "Manual Run".



- **Real Odoo 20 screenshot.** The sync log with the manual runs for employees, absences and expense claims.

The window options

Company	The company the import is for. By default, the active company.
Departments, **Employees**, **Absences**, **Expense Claims**	What gets imported. Each checkbox starts from what that company has enabled in the settings.
Force full employee pull	Ignores the incremental checkpoint and reads all users again.
Re-read every absence of the window	Ignores the incremental checkpoint and reads again from Lucca every absence within the sync window.

While you choose, **Run Sync** is at the bottom. When it finishes, the window shows the **Result** and the **Close** button.

When it does not let you run it

- Without the instance URL and the API key: "Configure the Lucca tenant URL and API key first (Settings > Lucca).". You fix it in the settings, in the Lucca section.
- With no box ticked: "Pick at least one thing to sync.".

The dashboard states

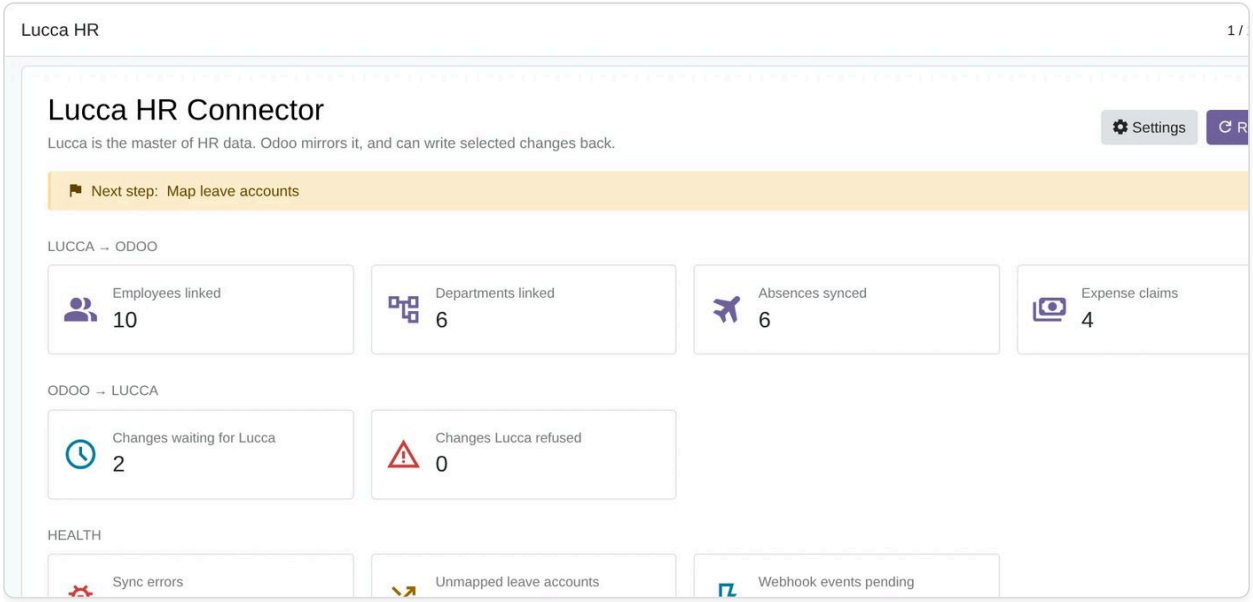
STATUS	WHAT IT MEANS
Configure your Lucca tenant	The instance URL is missing or the key is not saved in FlexigoTech Cloud.
Run your first employee sync	The connection is ready and no employee in the company is linked to Lucca yet.
Map leave accounts	The employees are in, **Absences** is on and some Lucca leave account has no Odoo leave type.
Ready	No step is pending.

What the dashboard counts

The counters are calculated only for the active company, even if you have several enabled. There is a single shared dashboard card. The blocks are: linked employees, departments and absences, expense claims, changes pending and rejected, sync errors, unmapped leave accounts, pending webhook events and the last employee, absence and expense sync.

An example with the data from the screens

After installing, the dashboard shows **Configure your Lucca tenant** and everything at zero. With the connection made, an import brought in 10 employees, 6 departments and 4 expense claims, with no errors and status **Done**. While any leave account remains unmapped, the dashboard stays at **Map leave accounts**. The screenshot also shows 2 changes pending for Lucca and 0 rejected.



- **Real Odoo 20 screenshot.** The dashboard after the first import: 10 employees and 6 departments linked, 4 expense claims.

CONNECTOR GUIDE · LUCCA

15 The data that goes and comes back

DATA	DIRECTION	WHEN
Employees	From Lucca to Odoo	**Lucca: sync employees**, every 30 minutes, or **Run Import**
Departments and their manager	From Lucca to Odoo	**Lucca: sync departments**, every hour, or **Run Import**
Absences	From Lucca to Odoo	**Lucca: sync absences**, every 15 minutes, and **Lucca: re-read all absences (nightly)**, every day
Expense claims and their lines	From Lucca to Odoo	**Lucca: sync expense claims**, every hour, or **Run Import**
Notices of changes in Lucca	From Lucca to Odoo	On arrival of each notice, with **Lucca: process webhook events** (and every 15 minutes)

DATA	DIRECTION	WHEN
Changes to the data of an employee linked to Lucca	From Odoo to Lucca	On save, with **Lucca: send changes to Lucca** (and every 5 minutes)
Analytic accounts, as sections of a Lucca axis	From Odoo to Lucca	On save, with **Lucca: send changes to Lucca** (and every 5 minutes), or **Push to Lucca** in **Lucca › Configuration › Push Analytic Accounts**

Employee changes only go to Lucca if the employee is linked to Lucca and the company has **Employee changes** switched on. Analytic accounts only when **Analytic accounts → Lucca axis** is on. What the connector reads from Lucca is never sent back to Lucca.

Calls to your Lucca account go through FlexigoTech Cloud on the way to Lucca. The API key and your instance URL are stored encrypted there, not in Odoo, and the content of the calls is not stored. The service requires a FlexigoTech Cloud subscription, with a 14-day trial and no card required.

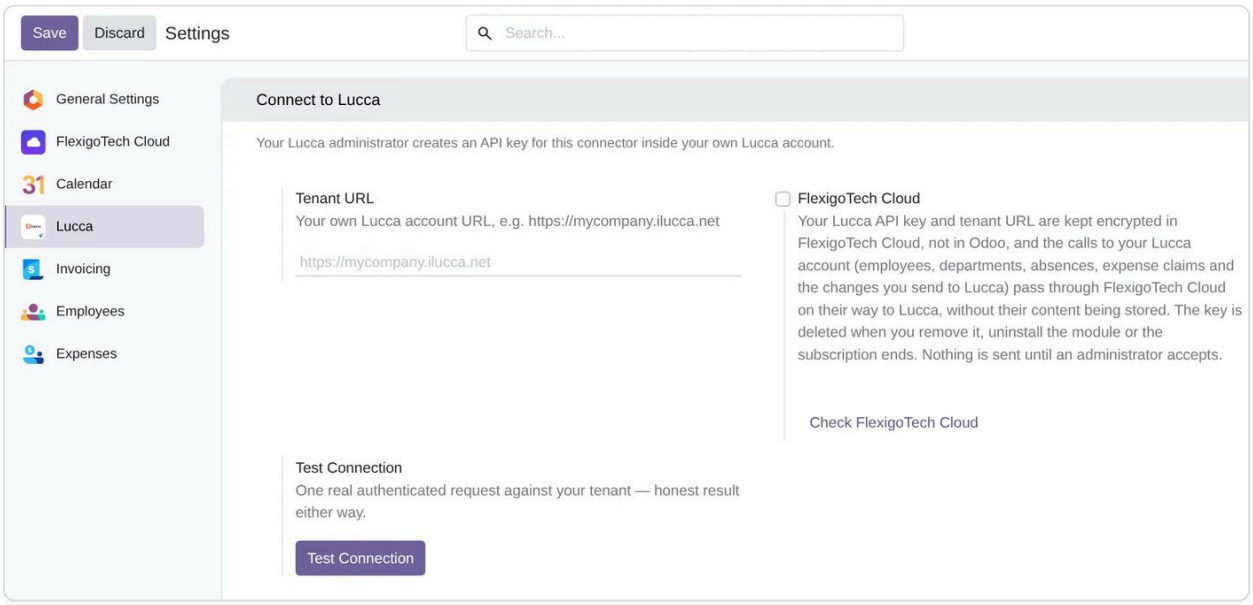
CONNECTOR GUIDE · LUCCA

16 What happens automatically

TASK	FREQUENCY	WHAT IT DOES
Lucca: sync employees	Every 30 minutes	Brings the new employees and changes from Lucca.
Lucca: sync departments	Every hour	Brings the departments from Lucca.
Lucca: sync absences	Every 15 minutes	Brings the absence changes.
Lucca: re-read all absences (nightly)	Every day	Reads all absences again, not just the changes.
Lucca: sync expense claims	Every hour	Brings the expense claims and their lines.
Lucca: process webhook events	Every 15 minutes	Processes the notices Lucca has sent to your Odoo. Each notice wakes it up on arrival.
Lucca: send changes to Lucca	Every 5 minutes	Sends the changes made in Odoo to Lucca. Each change wakes it up when saved.

The frequencies are changed in **Settings › Technical › Automation › Scheduled Actions**, with developer mode activated.

17 The Lucca options and how they appear



- **Real Odoo 20 screenshot.** The settings under the **Lucca** group and the **Connect to Lucca** block.

OPTION	ON INSTALLATION	WHAT IT DOES
Employees	On	Brings Lucca employees into Odoo.
Departments	On	Brings Lucca departments into Odoo.
Absences	Off	Brings Lucca absences into Odoo, according to the leave account mapping.
Absence Window (days back)	90	Absences are brought in from this many days back up to one year ahead.
Import Pending Absences	On	Also imports absences awaiting approval in Lucca, as pending approval. Approved ones are imported as approved and cancelled ones are removed.
Expense Claims	Off	Brings Lucca expense claims into Odoo.
Expense window (days back)	365	Only expense claims declared within this many days are imported.
Employee changes	Off	When on, an linked employee's work email and phone, mobile, job title, department or manager are written to Lucca when the record is saved in Odoo. All other data continues to be sent by Lucca.
Analytic accounts → Lucca axis	Off	Creates, renames and archives sections in a Lucca axis based on the analytic accounts of the chosen plan.
Odoo plan	Empty	The Odoo analytic plan whose accounts are copied. Only visible when the option above is on.
Lucca Axis ID	0	The number of the Lucca axis that receives the sections. Also only visible when that option is on.

If you extend the absence days or turn on the import of pending absences, Odoo re-reads absences from the beginning at the next sync. A change that does not extend the window does not trigger that re-read.

What is done by hand

TAB OR GROUP	BUTTONS
Connect to Lucca	**Test Connection**; for administrators: **Check FlexigoTech Cloud** and **Delete my Lucca key from FlexigoTech Cloud**
Lucca (menu)	**Run Import** and **Push Analytic Accounts**, in **Lucca › Configuration**

Check FlexigoTech Cloud shows what FlexigoTech Cloud stores, without the key. **Delete my Lucca key from FlexigoTech Cloud** only appears if the key is already stored there. When you delete it, the company's syncs stop.

What each part stores

Connect to Lucca	The instance URL, which must be a complete https:// URL and only the root, and the API key. The key is sent once to FlexigoTech Cloud, which stores it encrypted, and is not stored in Odoo. Beforehand, an administrator must tick the acceptance checkbox.
Changing the URL	If you change the URL after saving the key, you must enter the key again.
Webhook URL	The address registered in your Lucca account to receive events. It appears after saving the settings once.
Signing secret	The secret Lucca shows when declaring the webhook. Deliveries with an incorrect signature are rejected.

> The options belong to the active company. If you work with several, each has its own: choose the company at the top right before opening the settings.

18 Security and personal data

- The Lucca API key and your instance URL are stored encrypted in FlexigoTech Cloud, not in your Odoo. Calls to your Lucca account pass through FlexigoTech Cloud on the way to Lucca, and their content is not stored.
- You can remove the key at any time with **Delete my Lucca key from FlexigoTech Cloud**. Doing so stops that company's syncs. The button does not appear while no key is stored. The key is also deleted from FlexigoTech Cloud when the module is uninstalled or the subscription ends, and what has already reached Odoo stays in Odoo.
- **Check FlexigoTech Cloud** tells you what FlexigoTech Cloud stores without showing you the key.
- Before saving the key, an administrator must accept that the key and the calls to Lucca pass through FlexigoTech Cloud. If they have not accepted, the connector warns that nothing has been sent.
- The company, the email of the person accepting, the date and the version of the accepted text are stored.
- **Check FlexigoTech Cloud** and **Delete my Lucca key from FlexigoTech Cloud** are buttons in **Settings** that only an Odoo administrator can see.
- **Settings** and **Run Import** on the Lucca dashboard, and **Create Odoo Expenses** on an expense claim, are for the **Manager** group. The **User** group views the data without those buttons.
- From Lucca only each employee's work data is requested: name, surname and email, among others. Bank details, dates of birth and personal addresses are not requested.
- The instance URL must be a public https:// address: the connector rejects internal addresses when saving it.
- Each Odoo company stores its Lucca data separately. A user does not write to the Lucca account of a company they do not work for.

Sync log

Each sync attempt is recorded in the **Sync Log**: a call to Lucca, a batch of scheduled tasks, a notice received from Lucca or a manual import, with its type, the records touched, the number of errors and the status. If a sync fails, the connector leaves an activity for the administrator, at most one a day for each failing sync and company, so as not to flood them with alerts.

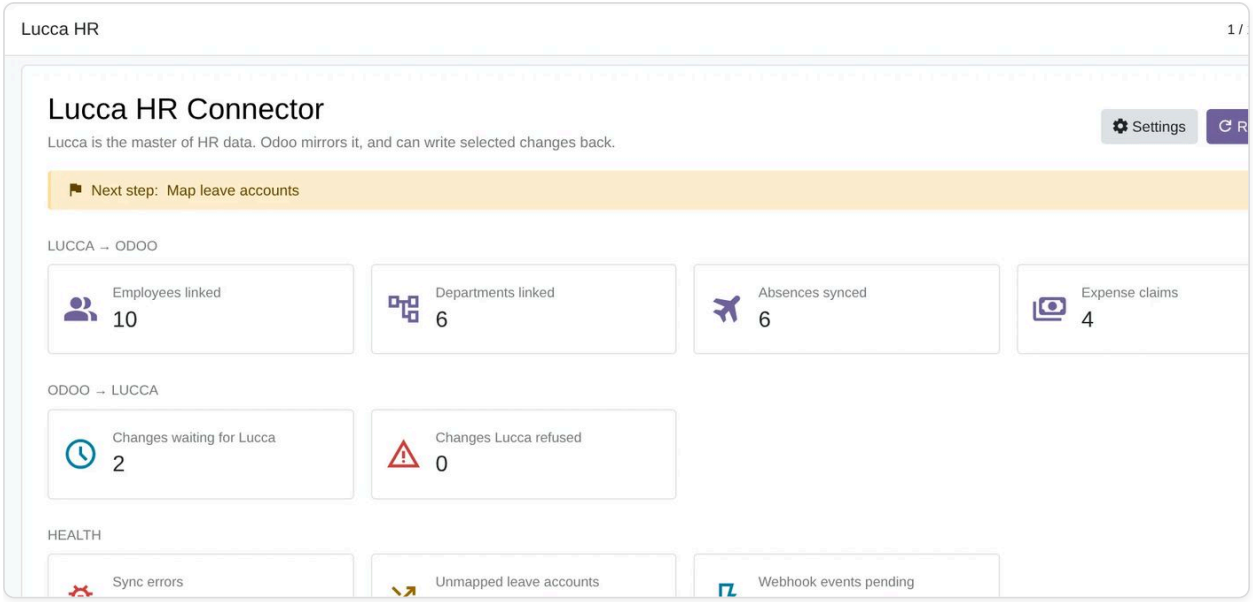
Changes that Odoo sends to Lucca are kept in a queue visible under **Changes Sent to Lucca**, with the method, the endpoint, the status and the attempts. A change that Odoo reverts never reaches Lucca.

19 Several Lucca accounts and several companies

Each Odoo company connects to its Lucca account in **Settings › Lucca**, with the active company chosen at the top. Each has its own URL, its own API key and its own options. The API key always belongs to one specific Lucca account, so it is not shared between companies.

Data is not mixed between companies. Absence sync only matches each person with the employee in the same company, and Lucca leave accounts are stored per company. Changes to an analytic account only go to the Lucca account of the company the person making them belongs to. If you work in company B, you do not write to company A's Lucca.

In FlexigoTech Cloud, each connected company counts as one Lucca account: the Standard plan supports up to 3 and the Pro plan up to 10.



- **Real Odoo 20 screenshot.** Lucca dashboard with the counters for the current company.

CONNECTOR GUIDE · LUCCA

20 If something fails

Lucca does not respond and there is no new data	Open the **Sync Log**. Each call to Lucca is recorded there with its status and its number of errors. If a sync fails, Odoo leaves an activity for the administrator, at most one a day for each failing sync and company.
A record is not created	Check the **Sync Log**. Each absence that is rejected or fails is recorded with the reason given by Odoo itself, for example a leave type that requires an allocation.
An absence is not imported	It is skipped if it is pending and the import of pending absences is off, if its holder is not linked to an employee of the company, or if its absence account has no Odoo leave type. Check **Leave Account Mapping**.
An expense claim gives an error	If the claim has no linked Odoo employee, the message asks you to sync employees first.
A change sent to Lucca does not go out	If Lucca does not respond or gives an error of its own, it is retried up to 5 times. If Lucca rejects it, for example because the key lacks a permission, it is left as **Failed** with the reason. Check **Changes Sent to Lucca** and its **Attempts** column.
An event from Lucca was not processed	You can requeue it with the retry button. Events with errors are kept until someone reviews them.

Without the FlexigoTech Cloud free trial or subscription, the connector does not talk to Lucca. If the key is not stored in FlexigoTech Cloud or there is no service, scheduled tasks stop and do not call Lucca. Only the stop code reaches FlexigoTech Cloud.

Other notices you may see

«Lucca is not configured for company "...". Set the tenant URL and API key in Settings > Lucca.»	The URL or key is missing. Fill them in in the Lucca settings and try again.

"The Lucca tenant URL must be a full https:// URL, e.g. https://mycompany.ilucca.net";	Enter the full address of your Lucca account, starting with https://.
"Enter only the tenant root URL (no path), e.g. https://mycompany.ilucca.net";	Remove everything after the account name.
"This host is not allowed as a Lucca tenant URL: ..."	Odoo does not accept that address. Use the real URL of your Lucca account.
"Before saving the key, an administrator has to accept that the Lucca API key and the calls to Lucca go through FlexigoTech Cloud. Nothing was sent."	An administrator must give consent before the key is saved.
"Enter your Lucca tenant URL before the API key. Nothing was sent."	Fill in the URL first, then the key.
"Only an administrator can manage the Lucca connection."	Ask an Odoo administrator to do it.
"Your Lucca tenant exposes no analytic axes (or the API key lacks permission to read them)."	Check the API key permissions in Lucca.
"Pick at least one thing to sync."	Tick departments, employees, absences or expense claims before running the import.

CONNECTOR GUIDE · LUCCA

21 Other ways to do it

WAY	WHAT IT INVOLVES
Manual changes from Lucca's panel	Someone checks Lucca's panel and re-enters employees, departments, absences and expense claims into Odoo. Every piece of data is entered twice, and the two systems fall out of step as soon as anything changes.
Odoo alone	Odoo 17, 18, 19 and 20 have no connection to Lucca, either in Community or Enterprise. Odoo on its own does not talk to Lucca.
This connector	It brings employees, departments, absences and expense claims from Lucca, and writes employee changes sent from Odoo back to Lucca. It needs your own Lucca account with an API key, a FlexigoTech Cloud subscription and the module licence. Every import is recorded in the **Sync Log**.

If you already track older expenses with another tool, **Expense window (days back)** limits the notes brought in to those declared within that many days. That way you start from a recent date and do not pull in Lucca's whole history.

22 Plans and pricing

€9

License per Odoo version.

+ FlexigoTech Cloud

STANDARD

€49/month

or €490 a year

- First response within 1 working day
- Up to 3 accounts

PRO

€98/month

or €980 a year

- First response within 4 working hours
- Up to 10 accounts
- Video call if needed

14-day trial, no card

Prices exclude VAT

The module license is per Odoo version. FlexigoTech Cloud is paid monthly or yearly, and the 14-day trial does not ask for a card. With Standard, up to 3 Lucca accounts; with Pro, up to 10. Each connected Odoo company counts as one.

23 Frequently asked questions

What does the connector do?

It brings employees, departments, absences and expense claims from Lucca into Odoo, and sends the employee changes you choose to Lucca. Lucca remains the master system.

Does it work without FlexigoTech Cloud?

No.

How much does the subscription cost?

The Standard plan costs €49 per month or €490 per year, and the Pro plan €98 per month or €980 per year, excluding VAT. On top of that, the module license costs €9 per Odoo version.

Is there a free trial?

Yes, 14 days with no card required. If you do not subscribe when it ends, the service stops and nothing is charged.

Does it work on Odoo Online?

No. Odoo Online does not allow third-party modules to be installed. It works on Odoo.sh or on your own server.

Which Odoo versions are supported?

Odoo 17, 18, 19 and 20, on Odoo.sh or on your own server.

Does it need Odoo Enterprise?

No. It works on Community and Enterprise: it only requires Employees, Time Off, Expenses and Analytic, which are Community apps, plus FlexigoTech Cloud.

Do I need a Lucca account?

Yes, your own Lucca account and an API key that your Lucca administrator generates for this connector. Lucca's fees are not included.

Who can connect the account?

Only an Odoo administrator can. If you are not one, Odoo replies "Only an administrator can manage the Lucca connection."

How many Lucca accounts does each plan allow?

On FlexigoTech Cloud, each connected company counts as one Lucca account: the Standard plan allows up to 3 and the Pro plan up to 10.

Can I connect several Odoo companies?

Yes. Each company is connected in **Settings › Lucca** with its own URL, API key and options. Data is not mixed between companies.

Where is the Lucca API key stored?

Encrypted on FlexigoTech Cloud, not in your Odoo. Calls to your Lucca account pass through FlexigoTech Cloud and their content is not stored.

Can I remove the key whenever I want?

Yes, with **Delete my Lucca key from FlexigoTech Cloud**. Doing so stops the synchronisations for that company.

How often do employees arrive?

The scheduled task brings them every 30 minutes, and arrivals and departures are applied at the next synchronisation.

How often do departments arrive?

Every hour.

How often do absences arrive?

Changes arrive every 15 minutes, and all absences are re-read once a day.

How often do expense claims arrive?

Every hour.

How often do Odoo changes go out to Lucca?

As soon as you save, through the queue in **Changes Sent to Lucca**. The task **Lucca: send changes to Lucca** checks every 5 minutes for anything that has not yet gone out.

Can I bring in the data without waiting for the tasks?

Yes, with **Run Import** in **Lucca › Configuration**. You choose what to bring in and the result is recorded in the **Sync Log**.

What happens to employees I already had in Odoo before installing it?

If the company has a single employee with no link and the same work email as in Lucca, the connector links them instead of creating another. If there is none, or more than one, it creates a new employee. From then on, Odoo recognises each person by their **Lucca ID**.

Are employees also created in Lucca?

No. Employees are created in Lucca: creating one in Odoo writes nothing to Lucca. Archiving one in Odoo is not a departure in Lucca either.

Which employee changes are sent to Lucca?

Only those for employees linked to Lucca, if the company has this enabled. Only what has changed is sent.

What happens to absences already in Odoo?

They stay as they are: the connector only touches absences that come from Lucca. From Lucca it reads those that fall within the **Absence Window (days back)**, up to a year ahead.

Does it import pending absences?

Only if the import of pending items is switched on. Confirmed ones are approved and cancelled ones are cancelled.

Do I need to prepare anything for absences?

Yes, the absence account mapping: each Lucca account needs an **Odoo Time Off Type**. Without a type, its absences are skipped.

Does it create Odoo expenses on its own?

No. You always create them yourself with **Create Odoo Expenses**, which generates a draft expense for each line without one.

How many expense claims does it bring in?

Those declared within the days set in **Expense window (days back)**, 365 by default.

Is the data I read from Lucca sent back to Lucca?

No. What the connector reads from Lucca is never sent back to Lucca.

Does it also write analytic accounts to Lucca?

Yes, if the company enables it: each account in the chosen analytic plan becomes a section of a Lucca axis. For those that already existed, you use **Push Analytic Accounts**. This has never been tested against a Lucca account: the sandbox key did not have permission to create sections.

How do I see what happened in each synchronisation?

In the **Sync Log**, with the type, the records touched, the errors and the status of each run.

Does it warn me if synchronisation fails?

Yes. It leaves an activity for the administrator, at most one per day for each failing synchronisation and company.

What happens if one of my changes does not reach Lucca?

It stays in **Changes Sent to Lucca** as **Failed**, with its reason and the number of attempts. **Retry** puts it back in the queue.

Has it been tested with a real Lucca account?

The published version, no: it was tested on Odoo 17, 18, 19 and 20 with simulated responses from Lucca's API and includes 373 automated tests. An earlier version was tested on Odoo 19 against a sandbox environment provided by Lucca.

Does it do payroll?

No. It keeps employees, absences and expenses up to date between Lucca and Odoo.

CONNECTOR GUIDE · LUCCA

24 Glossary

Lucca	A human resources platform. It is the master system: Odoo replicates its data.
Lucca instance	Your Lucca account, with its own URL. Each Odoo company connects to its own in **Settings › Lucca**.
Instance URL	The root address of your Lucca account, with https:// and no path, for example **https://mycompany.ilucca.net**.
API key	The key that the Lucca administrator generates for this connector. It is stored encrypted on FlexigoTech Cloud, not in Odoo.
FlexigoTech Cloud	The service through which calls to your Lucca account pass. It requires the 14-day free trial or a subscription.
Standard plan	The FlexigoTech Cloud plan for this connector, with up to 3 accounts.
Pro plan	The FlexigoTech Cloud plan with up to 10 accounts and video call support if needed.
Master system	The one that governs a piece of data. What the connector reads from Lucca is not sent back to Lucca.
Synchronisation	Bringing data from Lucca into Odoo, or sending the changes you choose to Lucca.
Scheduled task	An Odoo process that synchronises on its own at set intervals. The connector brings seven, all active on installation.
Run Import	A menu and button that start the load manually, without waiting for the scheduled tasks.
Absence account	Each type of absence in Lucca, with its identifier and name.
Absence account mapping	The relationship between each Lucca absence account and an Odoo time off type. Without a type, its absences are skipped.
Expense note	A set of expense lines that an employee declares in Lucca.
Odoo expense	A draft expense created from an expense claim with **Create Odoo Expenses**.
Changes Sent to Lucca	Queue of changes that Odoo sends to Lucca, with the statuses **Pending**, **Sent** and **Failed**.
Webhook event	Notice that Lucca sends to Odoo when something changes. They are listed under **Lucca › Configuration › Webhook Events**.
Sync Log	List where each call, batch of jobs, notice or manual import is recorded.
Dashboard	Home screen of the app, under **Lucca › Dashboard**, with counters for what arrives from Lucca and what Odoo sends.

25 Checklist and contact

- [ ] You have your own Lucca account and the URL of its instance.
- [ ] Your Lucca administrator has created an API key for this connector.
- [ ] Your Odoo is version 17, 18, 19 or 20, with the Employees, Time Off, Expenses and Analytic apps.
- [ ] You have installed the module from the Odoo app manager.
- [ ] You have activated the 14-day FlexigoTech Cloud trial, without a card, or you already have the subscription.
- [ ] Under **Settings › Lucca** you have entered the root URL and the API key, and an administrator has ticked the acceptance box.
- [ ] You have clicked **Test Connection** and Lucca responds.
- [ ] You have reviewed **What to sync**: employees and departments are switched on; leave and expense reports are switched off.
- [ ] You have run **Run Import** and the **Sync Log** shows the runs as **Done** and without errors.
- [ ] You have mapped the leave accounts under **Leave Account Mapping** and the dashboard shows **Ready**.

If you would like us to set it up with you, write to us. On the Pro plan, if needed, we can look at it by video call.

Email

To install it, set it up or for any question.
comercial@flexigotech.com

On the web

The connector page, with the guide and the prices.
flexigotech.com/en/lucca-connector-odoo.html

On Odoo Apps

The module page on the Odoo store.
apps.odoo.com

Lucca is a trademark of its owner. FlexigoTech publishes this connector on its own, with no endorsement or sponsorship from the trademark owner. FlexigoTech is a brand of Flexibles y Accesorios Gobe, S.L. (Barcelona, Spain).