AXIVOX

AXIVOX FOR ODOO · INSTALLATION GUIDE

# Installing the Axivox suite in Odoo.sh 19

The Axivox call log on an Odoo.sh project: every call, with its contact, its recording, and what the conversation was about.

| Odoo | Module | Module version | Document |
|---|---|---|---|
| 19 on Odoo.sh | axivox | 19.0.1.0.4 | 21 September 2026 |

1. Add the repository to your project
2. Push, and let Odoo.sh rebuild
3. Turn on the developer mode
4. Install the module from Odoo
5. Create an API key in Odoo
6. Declare your Odoo in the Axivox portal
7. Check that everything works

## What you need

- An active **Axivox account**.
- **Administrator** access to the Odoo.sh database concerned.
- Access to the **Git repository** of your Odoo.sh project, and the right to push to it.
- About ten minutes, plus the time the branch takes to rebuild.

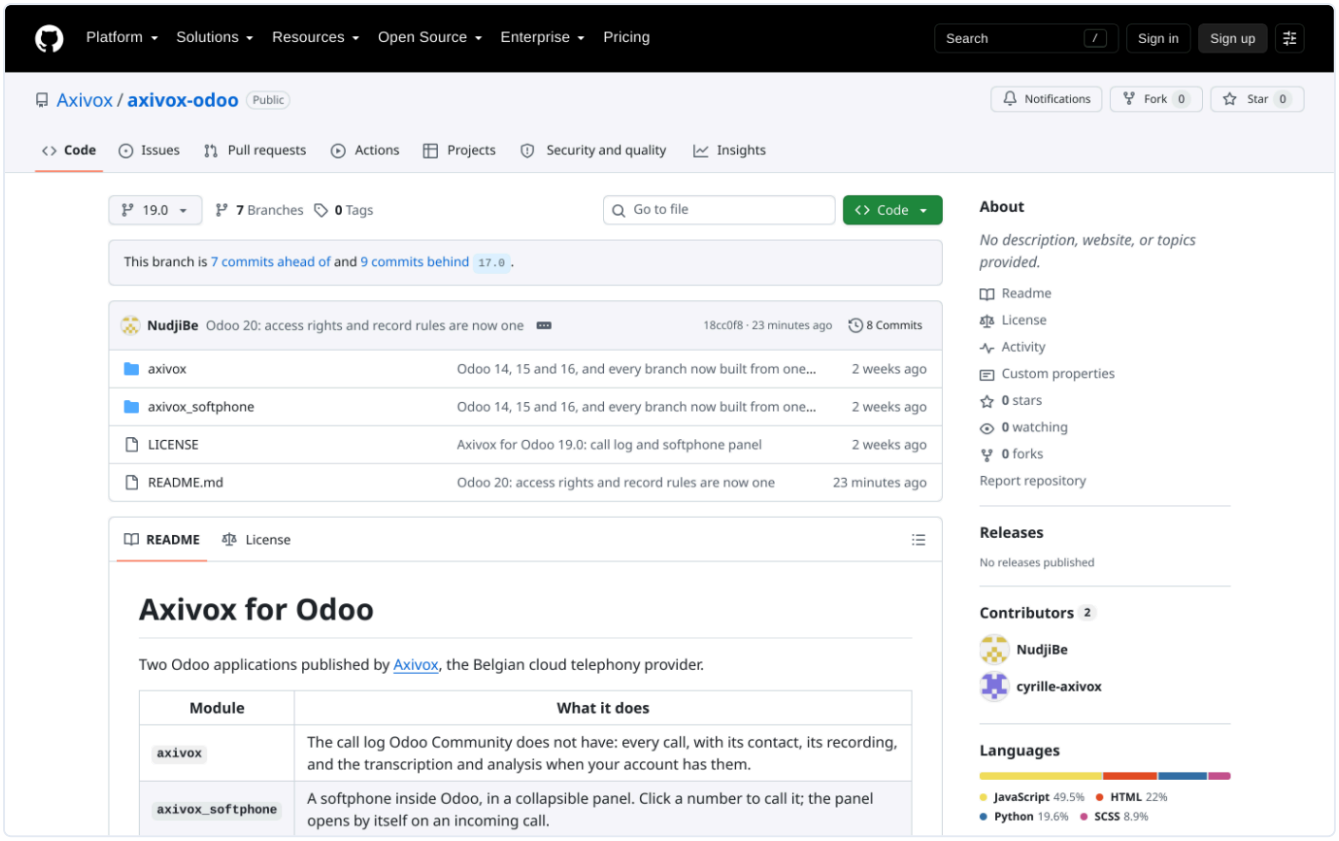
> **On Odoo.sh you already have Odoo's Phone application.** Odoo.sh projects run the Enterprise edition, where the call log exists. This module remains useful if you prefer ours, which also carries the number that was called, the talk time, the sentiment, the themes and the suggested escalation. Axivox writes to one or the other depending on what it finds: nothing to choose in the portal.

> **Do not run both call logs side by side.** If Odoo's Phone application is installed, it is enough: a second log would create two places to look for the same call.

## STEP 1 Add the repository to your project

The module code is public on GitHub, one branch per Odoo version. On Odoo.sh it is added as a **submodule** of your project repository: Odoo.sh detects it and puts it in the addons path on its own.

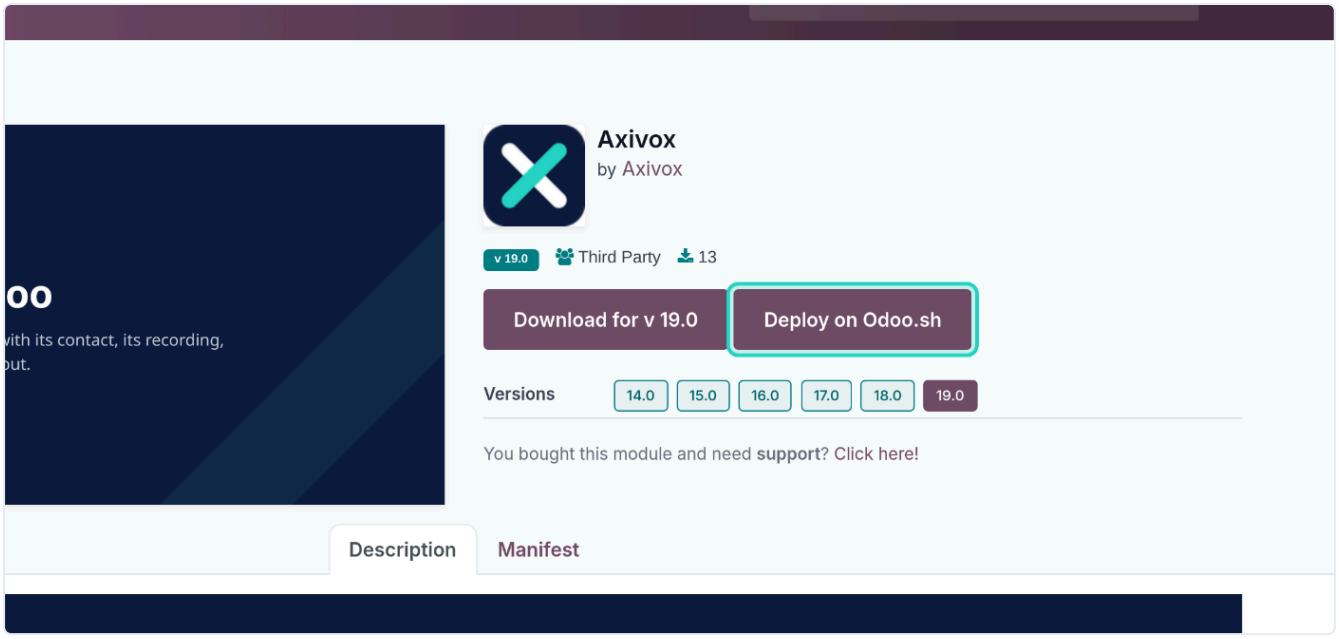
```
git submodule add -b 19.0 https://github.com/Axivox/axivox-odoo.git axivox-odoo
git commit -m "Axivox"
git push
```




The public repository, on branch 19.0. The two module folders sit at its root, which is the shape Odoo.sh expects.

### Or from the Odoo store

The module page carries a **Deploy on Odoo.sh** button, which does the same from your browser, with no command line.




The **Deploy on Odoo.sh** button, next to the download.

> **Take branch 19.0.** Odoo's screens are not written the same way from one version to the next. A package meant for another version sometimes refuses to install, and sometimes installs with part of its screens left empty, without saying anything.

## STEP 2 Push, and let Odoo.sh rebuild

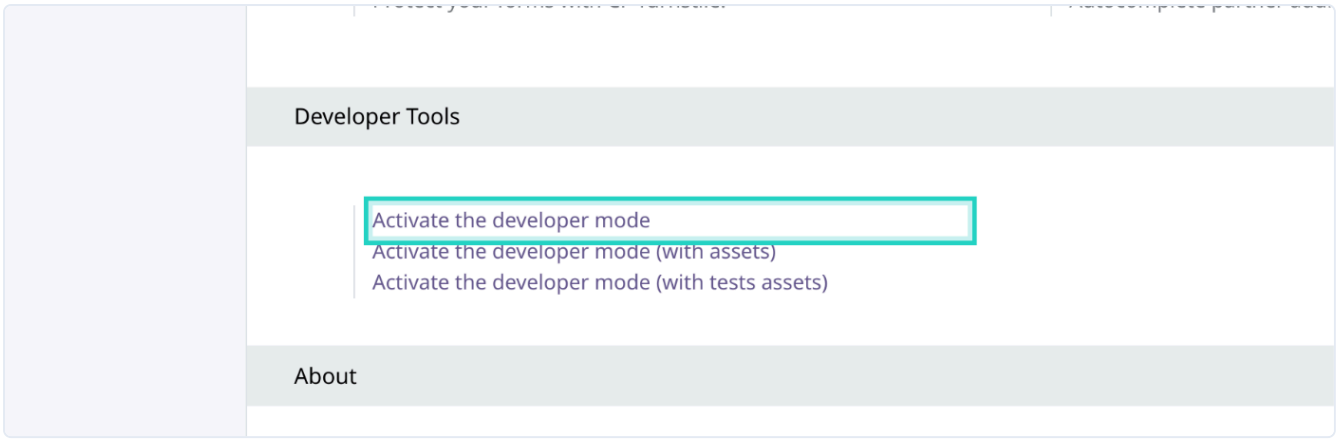
Odoo.sh rebuilds the branch as soon as the commit arrives. Wait for the build to finish and turn green before going further: while it runs, the module is not on the server yet.

> **Try it on a test branch first.** That is what Odoo.sh is for. Add the submodule on a development or staging branch, check the installation on its database, then merge to production.

The server restart, needed for a module carrying code to be read, is part of the rebuild. There is nothing more for you to do.

## STEP 3 Turn on the developer mode in Odoo

Open **Settings**, scroll to the very bottom of the page, section **Developer Tools**, then click **Activate the developer mode**.



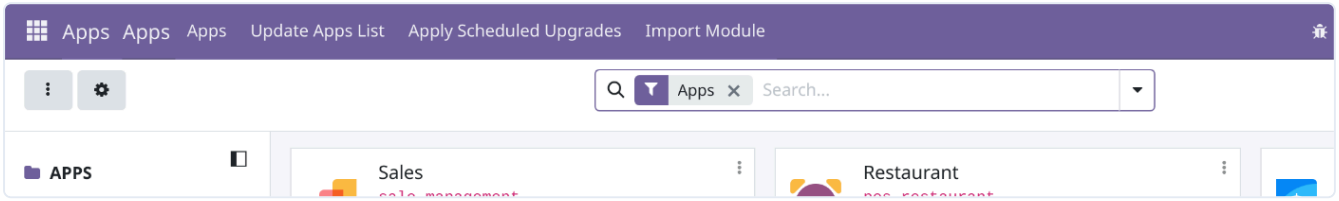

The link sits at the bottom of the general settings, under **Developer Tools**.

The page reloads. The developer mode opens the menu entries used in the next step. You can turn it off once the installation is done, the module keeps working.

> **There is a shortcut.** Adding `?debug=1` at the end of your Odoo address does the same, from any screen.

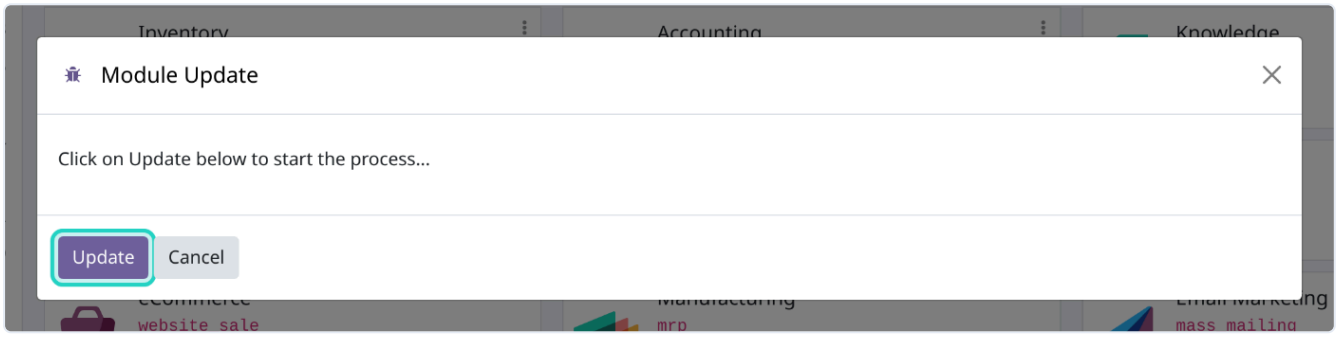
## STEP 4 Install the module from Odoo

Open the **Apps** menu. Two extra entries appeared at the top of the screen with the developer mode.



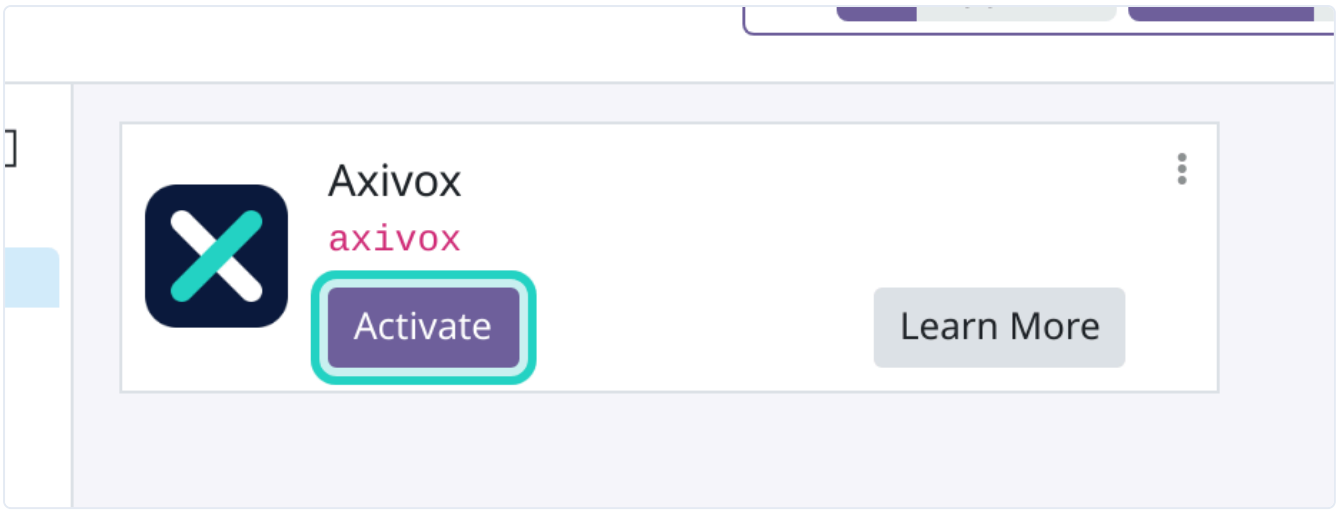

**Update Apps List**, in the top bar.

Click it, then confirm. Odoo reads the modules directory again; without that pass, the module stays invisible.




The confirmation. The operation touches none of your data.

Then type **Axivox** in the search. The module card appears. Click **Activate**.



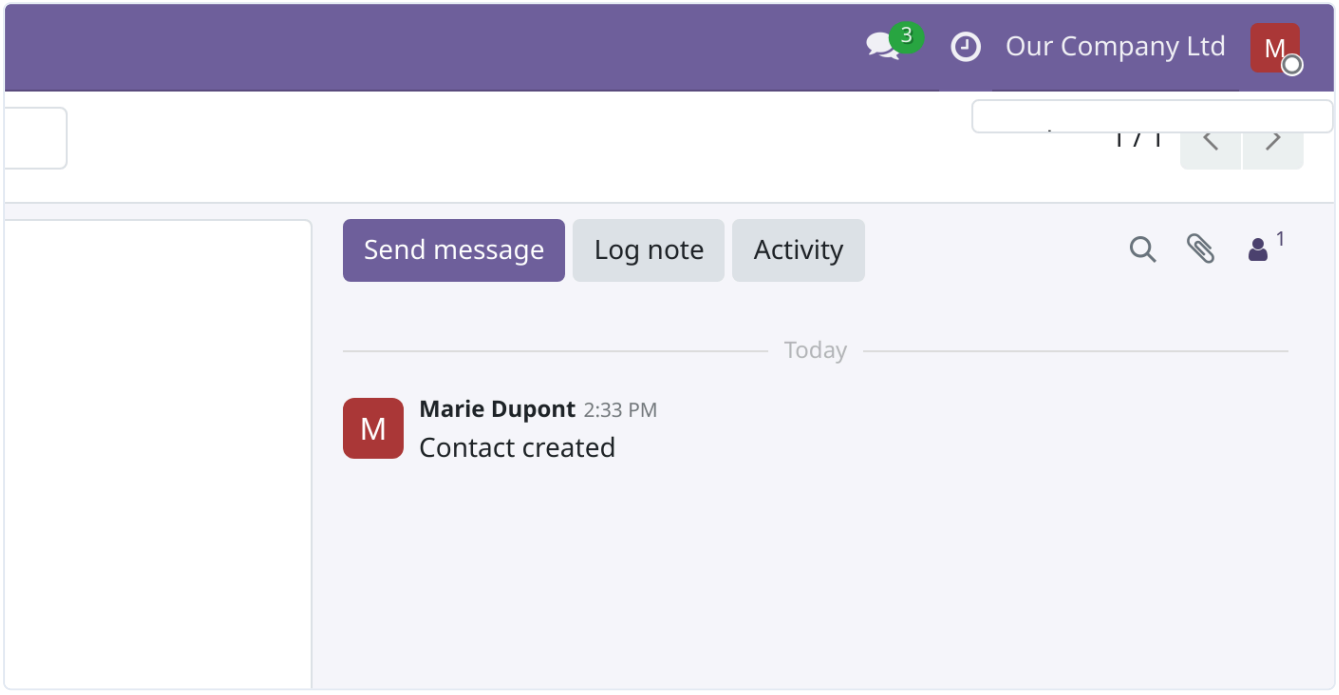

The module card in the apps list.

The installation takes a few seconds. An **Axivox** menu then appears in your list of applications.

## STEP 5 Create an API key in Odoo

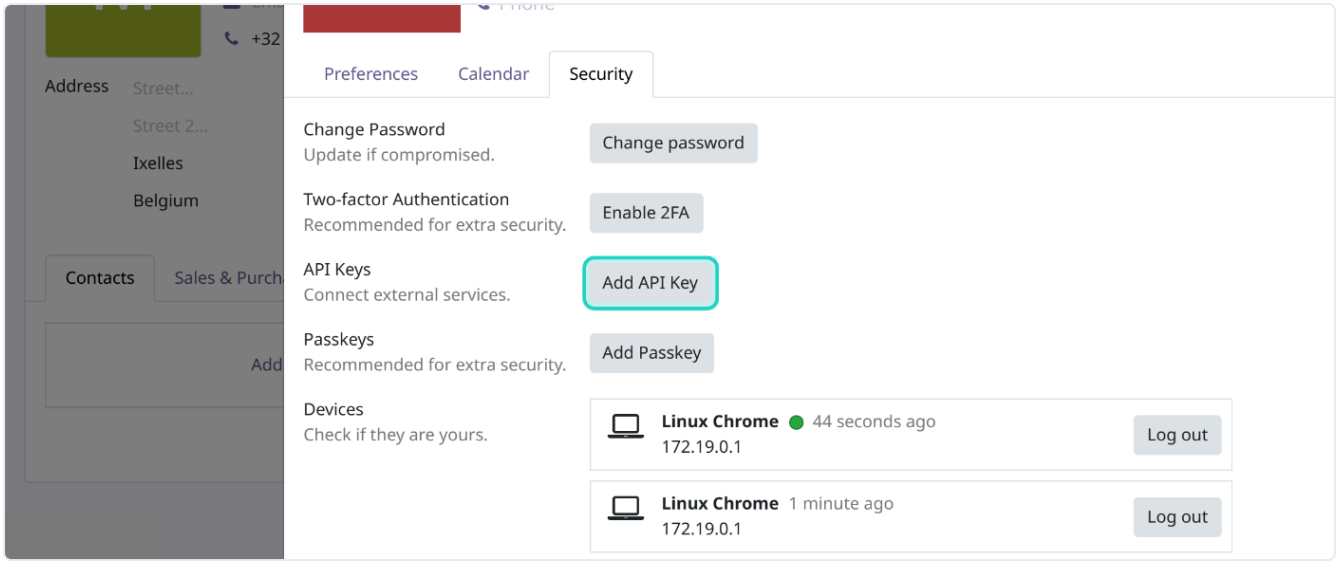
This key lets Axivox write calls into your Odoo. Create it on the account you want to appear as the author of what Axivox writes. An account dedicated to the integration works well.

Open the menu under your name, top right, then **My Preferences**.



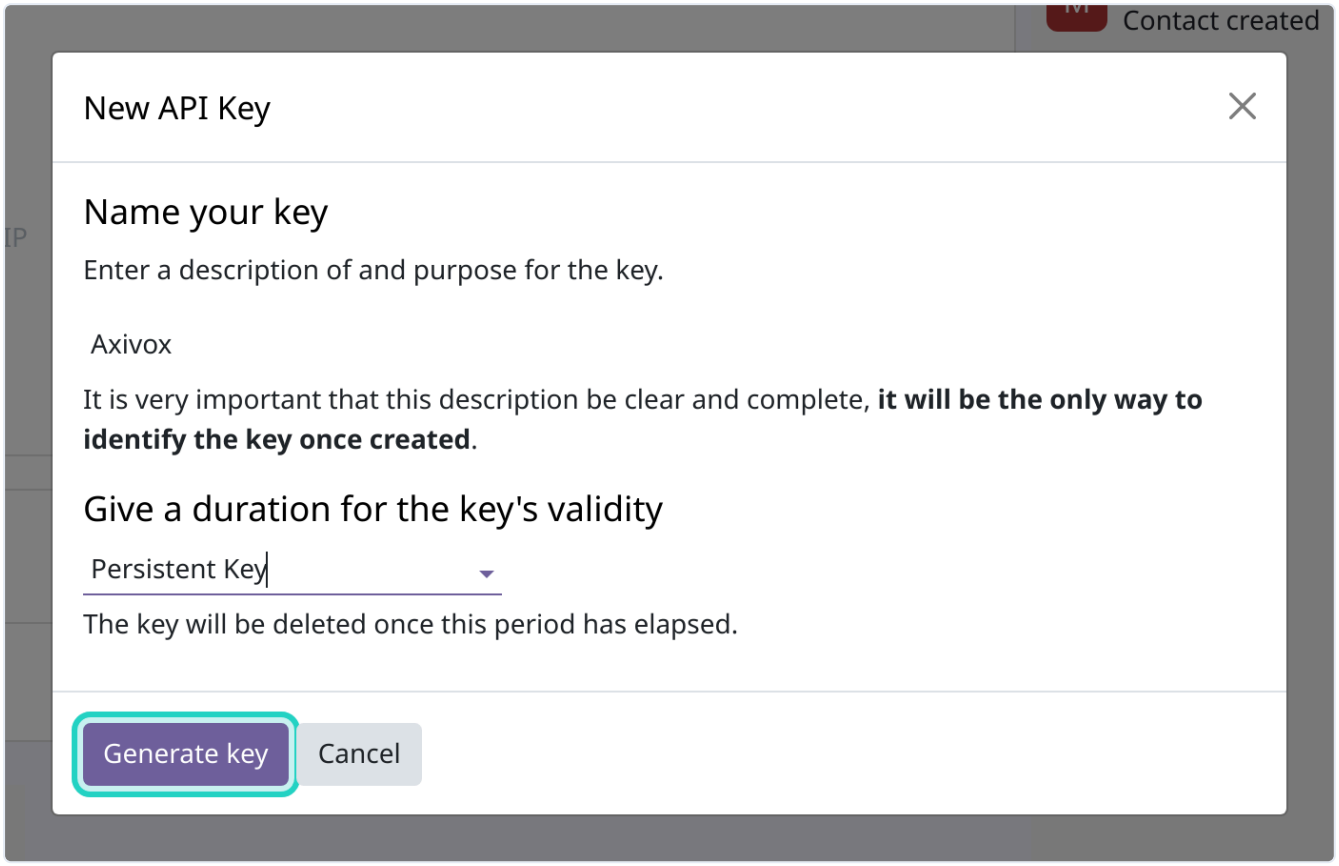

The account menu, at the top right of the screen.

In the form that opens, go to the **Security** tab, then **Add API Key**.




The **Security** tab and the button that creates the key.

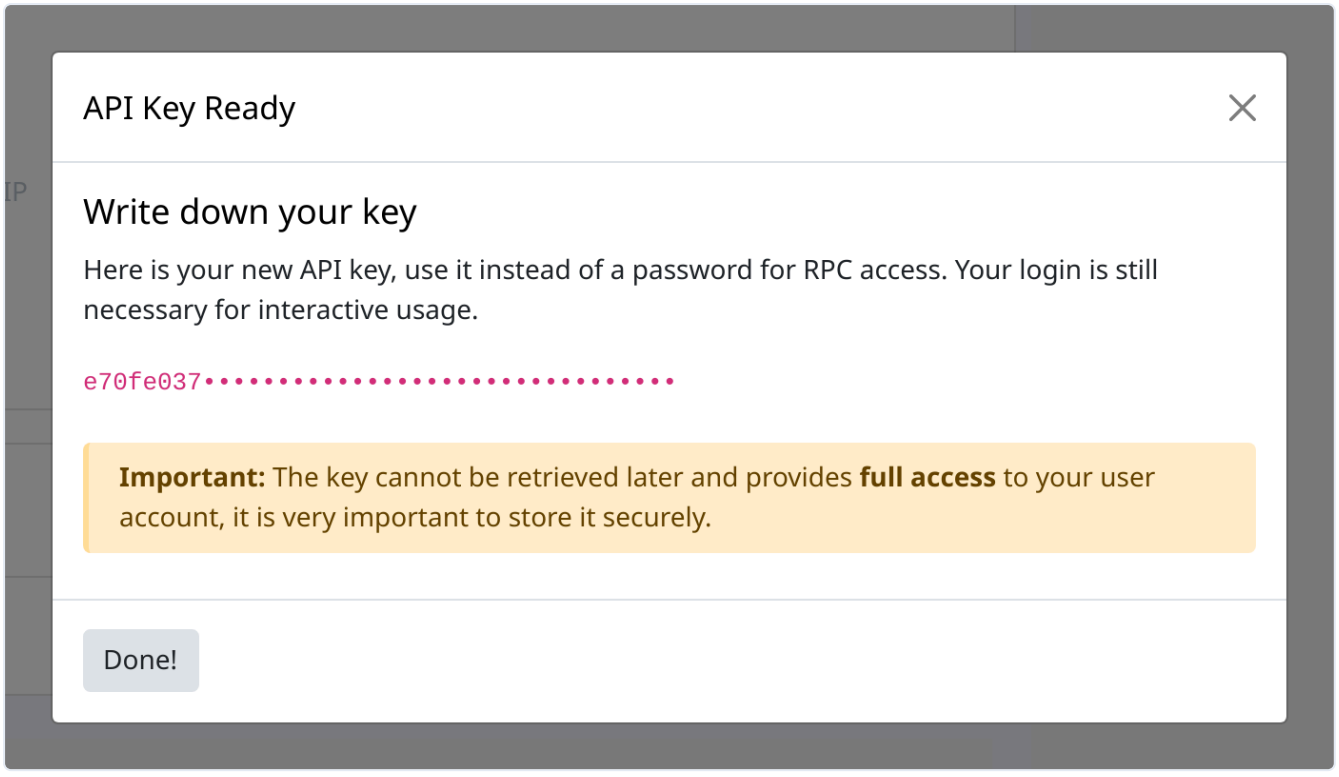
Odoo asks for your password, then for a description. Write for instance `Axivox connector`: it is the only way to recognise this key later.




The description of the key. It identifies the key, it is not used to connect.

> **Choose « Persistent Key » as the validity.** Odoo offers one day by default. A key that expires cuts the flow of calls overnight, with no other sign than a connection test that starts failing.

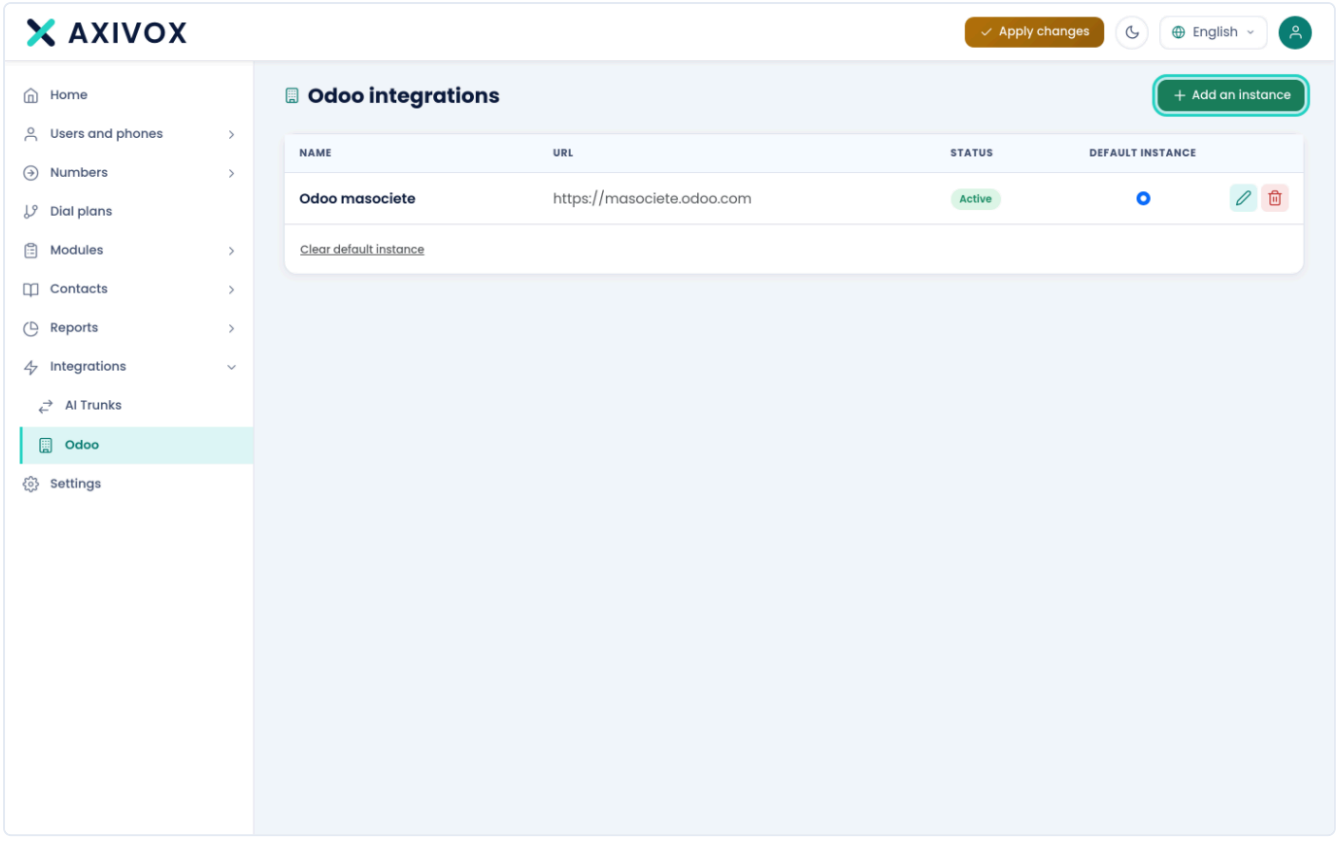
**The key is shown once, and only once.** Copy it straight away. If you lose it, you have to create another one. It gives full access to the account it was created on, so treat it as a password.




The key, once created. The one in this screenshot is masked.

## STEP 6 Declare your Odoo in the Axivox portal

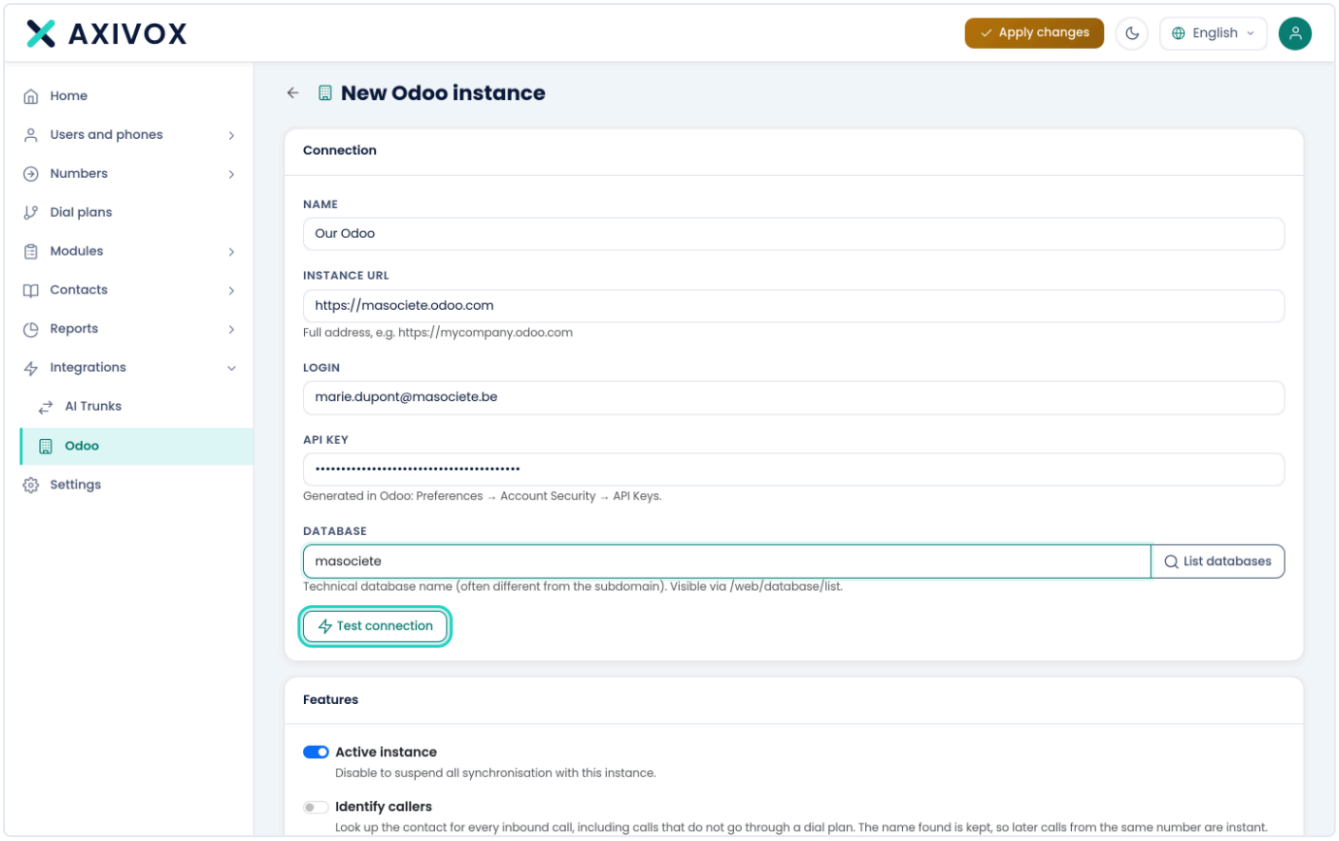
Sign in to **manage.axivox.com**, open **Odoo integrations** and click **Add an instance**.




The **Odoo integrations** screen of the portal. Several Odoo can live there.

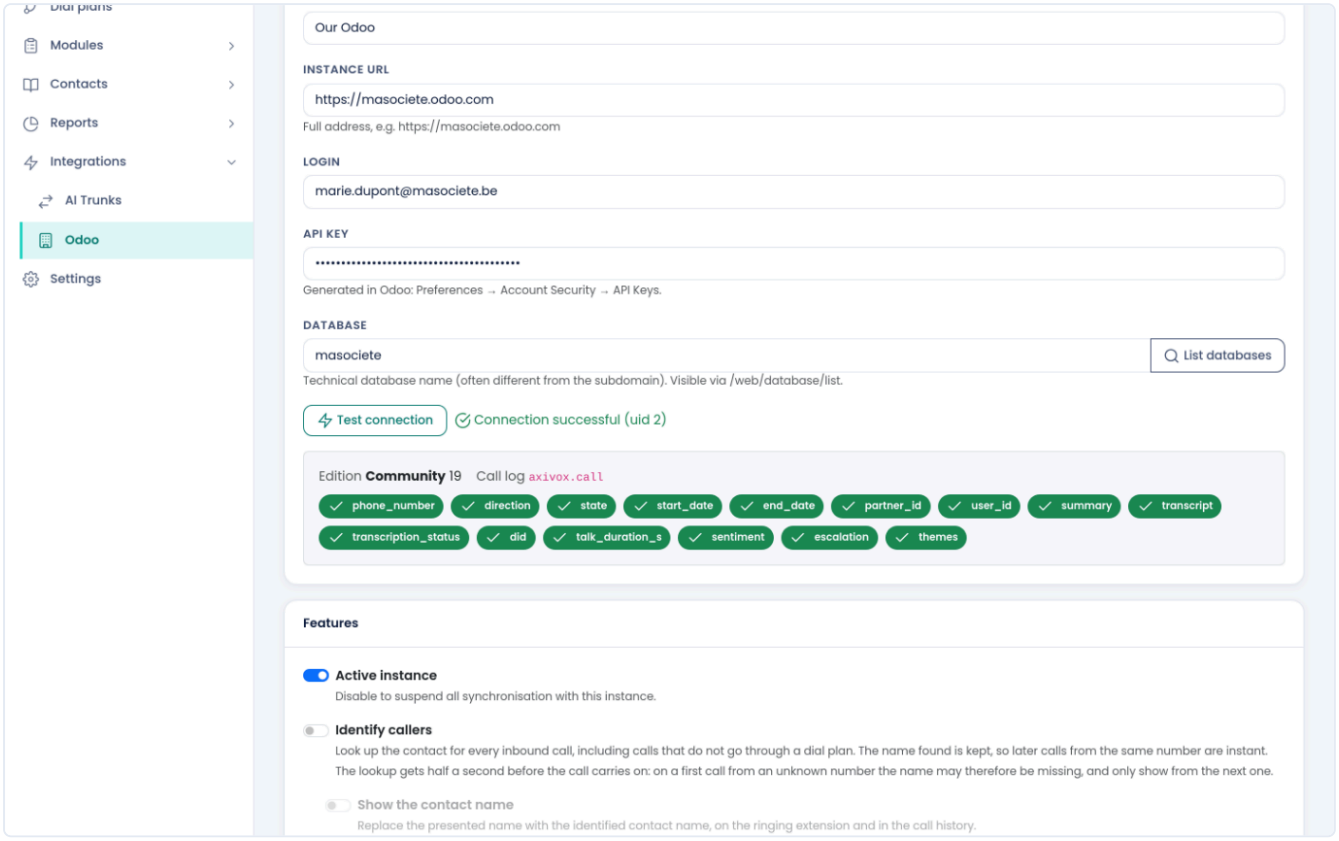
Fill in the five fields of the **Connection** section:

| FIELD | WHAT IT EXPECTS |
|---|---|
| **Name** | The name you give this Odoo. It is for you only. |
| **Instance URL** | The full address, for instance `https://mycompany.odoo.com`. |
| **Login** | The sign-in address of the account you just created the key on. |
| **API key** | The key copied at the previous step. |
| **Database** | The technical name of the database. It often differs from the subdomain: use the **List databases** button rather than guess it. |




The form filled in, before the test. The values in this screenshot are invented.

## Click **Test connection**.

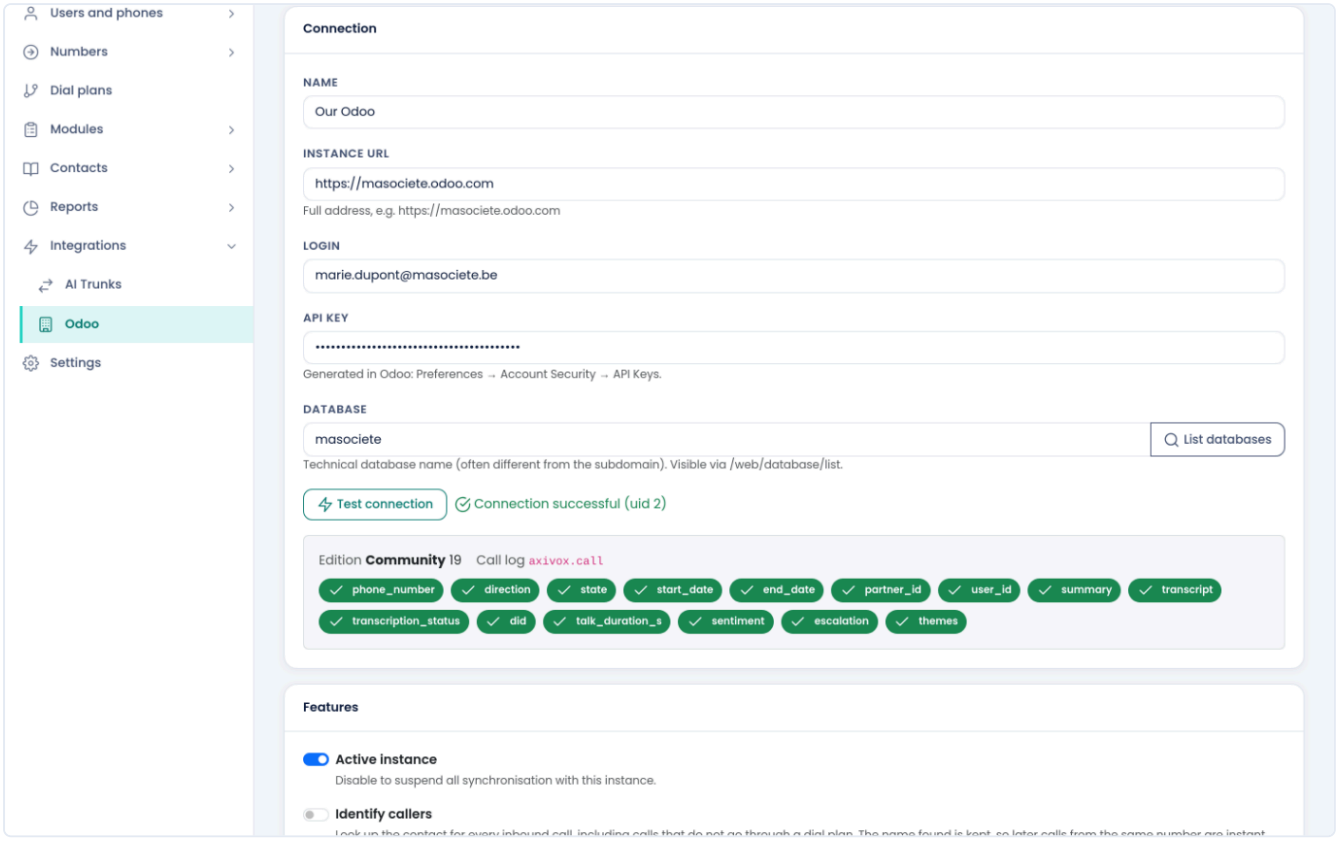



The test passing on an Odoo 19. The edition, the version and the call log found are shown, then the list of the fields Axivox will be able to fill.

> **The test does more than check a password.** It records what your Odoo can receive, and that is the list used afterwards. It also adds the connecting account to the group that lets it write the calls of the whole team, without which it could only file its own.

Save. The **Features** section then decides what Axivox does with this Odoo:

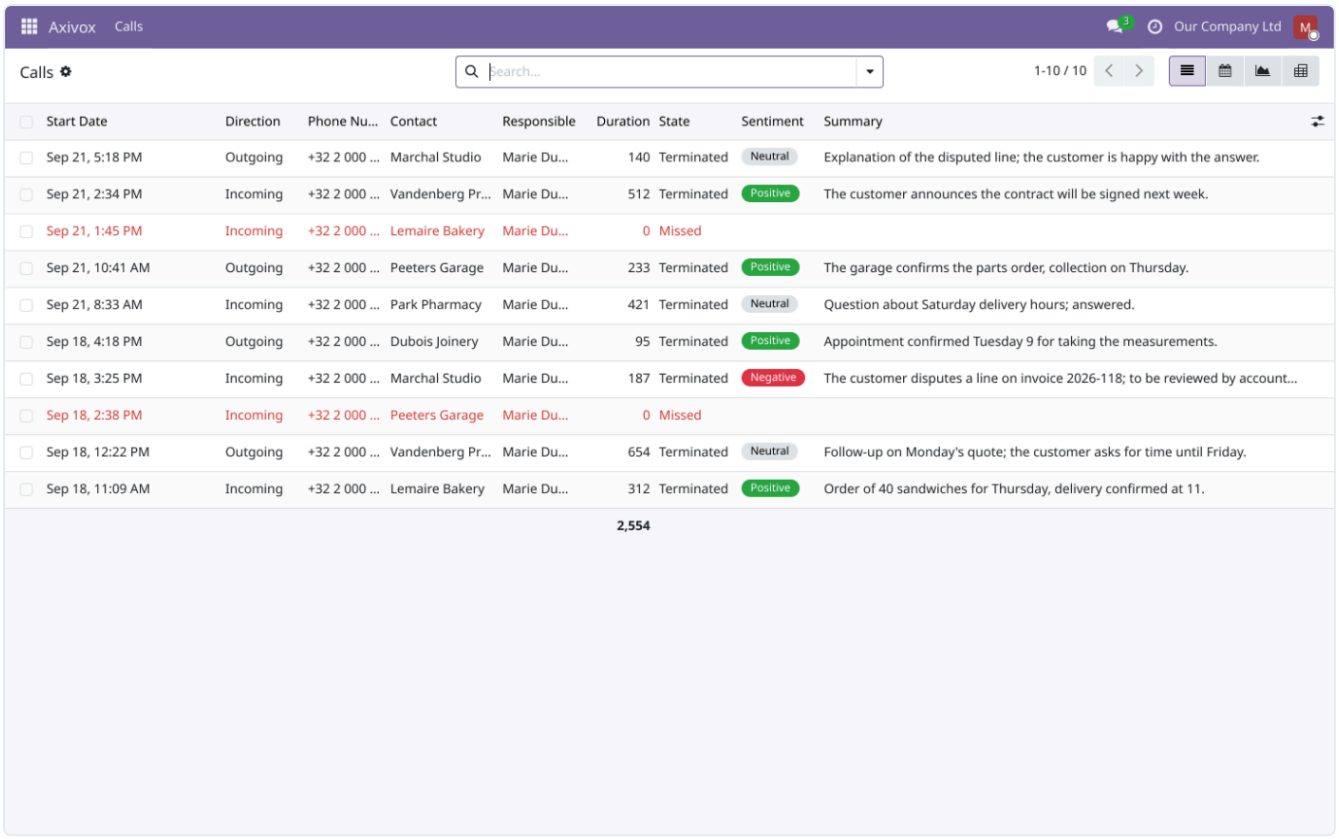
| SETTING | EFFECT |
|---|---|
| **Active instance** | Clear it to suspend every exchange with this Odoo, without losing the configuration. |
| **Identify callers** | The name of the Odoo contact comes up on incoming calls. |
| **Log calls** | Every call becomes a record in the call log. This is the main setting. |
| **Publish the transcript and the analysis** | The text of the call and its analysis are posted on the contact, when your Axivox account has them. |
| **Attach the audio recording** | The audio file comes with that message. It only appears if the transcript is published too. |




The **Features** section, below the connection.

## STEP 7 Check that everything works

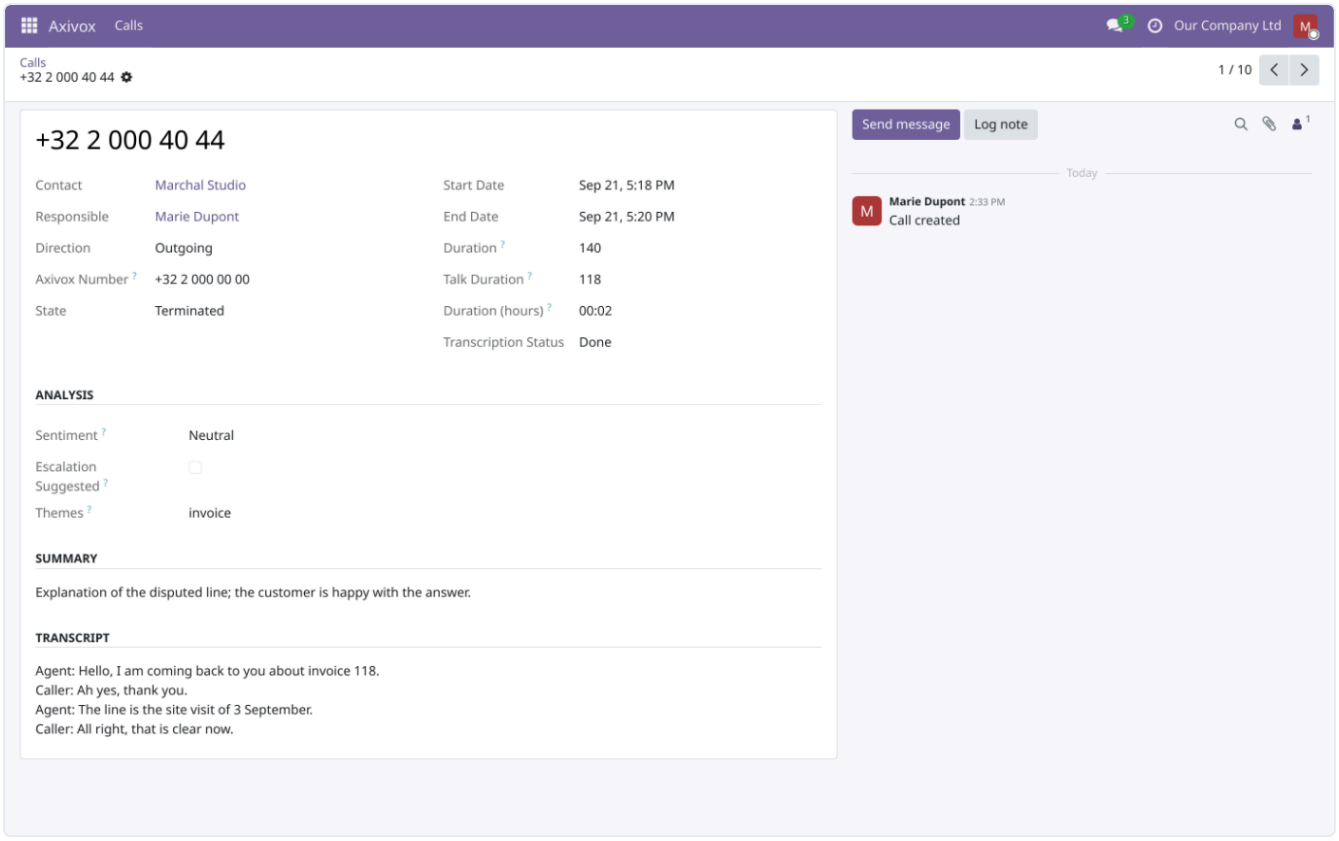
Make a call, or have someone call you. Within the minute it appears in the **Axivox** menu of your Odoo.




| Start Date | Direction | Phone Nu... | Contact | Responsible | Duration | State | Sentiment | Summary |
|---|---|---|---|---|---|---|---|---|
| Sep 21, 5:18 PM | Outgoing | +32 2 000 ... | Marchal Studio | Marie Du... | 140 | Terminated | Neutral | Explanation of the disputed line; the customer is happy with the answer. |
| Sep 21, 2:34 PM | Incoming | +32 2 000 ... | Vandenberg Pr... | Marie Du... | 512 | Terminated | Positive | The customer announces the contract will be signed next week. |
| Sep 21, 1:45 PM | Incoming | +32 2 000 ... | Lemaire Bakery | Marie Du... | 0 | Missed | | |
| Sep 21, 10:41 AM | Outgoing | +32 2 000 ... | Peeters Garage | Marie Du... | 233 | Terminated | Positive | The garage confirms the parts order, collection on Thursday. |
| Sep 21, 8:33 AM | Incoming | +32 2 000 ... | Park Pharmacy | Marie Du... | 421 | Terminated | Neutral | Question about Saturday delivery hours; answered. |
| Sep 18, 4:18 PM | Outgoing | +32 2 000 ... | Dubois Joinery | Marie Du... | 95 | Terminated | Positive | Appointment confirmed Tuesday 9 for taking the measurements. |
| Sep 18, 3:25 PM | Incoming | +32 2 000 ... | Marchal Studio | Marie Du... | 187 | Terminated | Negative | The customer disputes a line on invoice 2026-118; to be reviewed by account... |
| Sep 18, 2:38 PM | Incoming | +32 2 000 ... | Peeters Garage | Marie Du... | 0 | Missed | | |
| Sep 18, 12:22 PM | Outgoing | +32 2 000 ... | Vandenberg Pr... | Marie Du... | 654 | Terminated | Neutral | Follow-up on Monday's quote; the customer asks for time until Friday. |
| Sep 18, 11:09 AM | Incoming | +32 2 000 ... | Lemaire Bakery | Marie Du... | 312 | Terminated | Positive | Order of 40 sandwiches for Thursday, delivery confirmed at 11. |
| | | | | | 2,554 | | | |

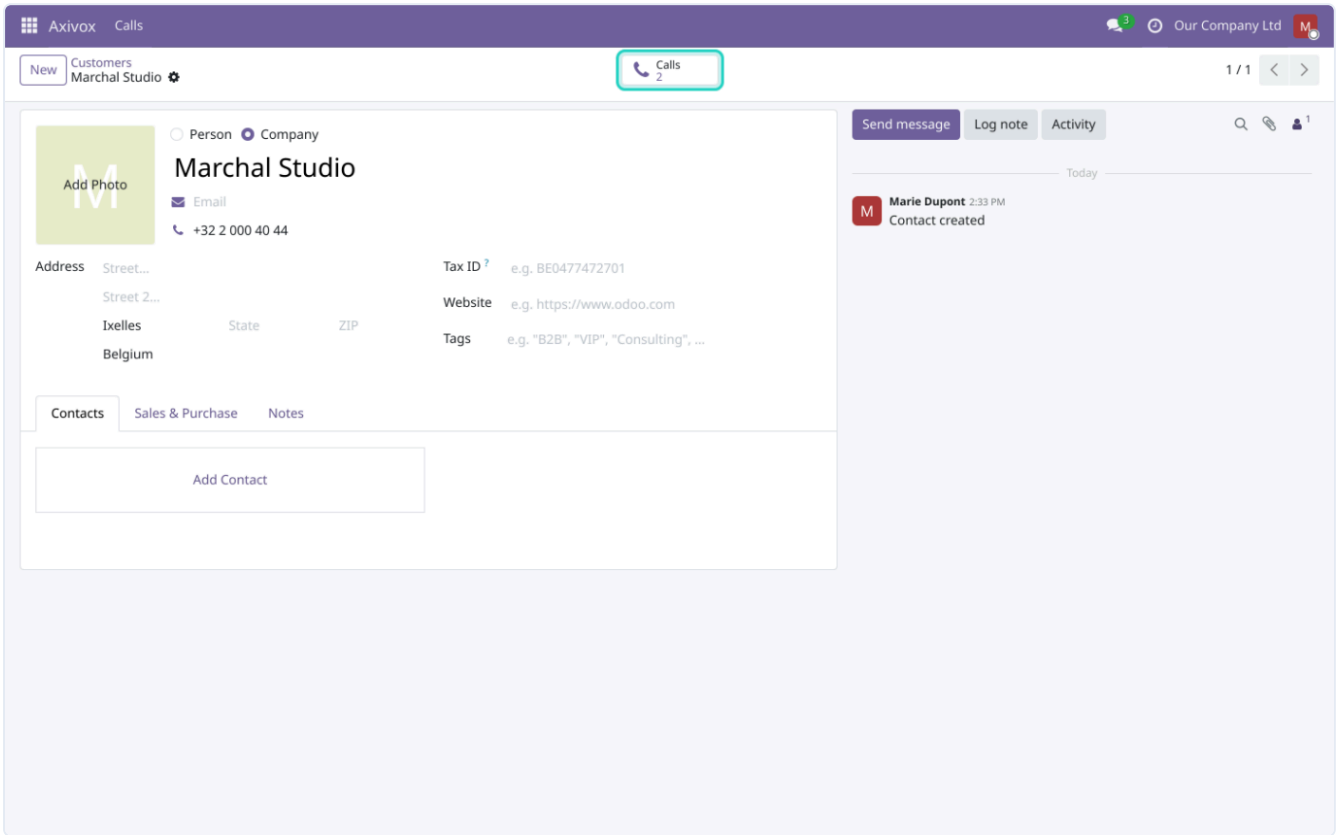
The call log. Missed calls stand out in red, and the sentiment read from the conversation shows as a badge.

A call opens like any Odoo record: both durations, the contact, the analysis, the summary, then the transcript.




One call in detail.

Every contact also carries its number of calls, and opens its own history.




The **Calls** button at the top of the contact, with the number of calls. It opens the history of that contact.

> **The screens of the module are in English.** Its labels deliberately reuse those of Odoo's own call log, which carry the same names. The rest of your Odoo is untouched.

> **A call that belongs to nobody appears in nobody's list.** Axivox attaches the call to the Odoo user whose sign-in address matches the e-mail address of the extension in Axivox. If the two differ, the call is created and attached to the contact, but no user sees it among « their » calls. Check the addresses of the extensions in the Axivox portal before concluding that something is broken.

## Updating later

Odoo never downloads the new version of a third-party module. The **Apps** screen only reads what is already on the server, and offers the upgrade once it finds a version number higher than the one recorded at install time. The files first, Odoo second.

1. Update the submodule, then push: Odoo.sh rebuilds the branch on its own.
2. In **Apps**, search for Axivox and click **Upgrade**.
3. Come back to the Axivox portal and run **Test connection** again.

> **The third step is not optional.** Axivox only sends your Odoo the fields it has seen it declare, and that list is recorded at the moment of the connection test. Until the test is run again, a version that brings new fields will see them dropped in silence.

> **Do not uninstall the module to reinstall it.** Uninstalling erases your calls. The upgrade keeps them, and leaves the new columns empty on the calls already there.

## Adding the softphone too

The repository you just added holds a second application, `axivox_softphone`, which puts your softphone in a panel of Odoo: a click on a number dials it, and the panel opens by itself when someone calls you. It is already on the server, you only have to install it from **Apps**. It adds no model, no field and no table, and needs no configuration: you sign in to your Axivox account in the panel, once.

> **That one does go through « Import Module ».** As it carries no code, the button in the interface installs it without touching the server. That is the path described in the Odoo Online guide, and it holds here too.

## If something goes wrong

| WHAT YOU SEE | WHAT TO LOOK AT |
|---|---|
| The module does not show up in Apps | Check that the branch build is finished and green, then run **Update Apps List** again. |
| The build fails | Most often the submodule branch does not match the Odoo version of the project. The Odoo.sh build log names the module at fault. |
| The connection test fails | Most often the database name: on Odoo.sh it carries the name of the branch. Use **List databases**. Then check that the API key belongs to the account given in **Login**. |
| Calls do not reach Odoo | Check that **Log calls** is ticked on the instance, in the Axivox portal. That is the most frequent cause. |
| Calls are there but belong to nobody | The e-mail address of the extension in Axivox must match the sign-in address of the Odoo user. Without a match, the call stays without an owner rather than being attributed at random. |
| The same call twice | Odoo's Phone application and this module are both installed. Keep only one. |

**A question?** Your Axivox account manager answers on both applications, as on the rest of your installation. www.axivox.com

The screenshots in this document were taken on an Odoo 19 in English.