

Installing the Axivox suite in Odoo.sh 14

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
14 on Odoo.sh	axivox	14.0.1.0.4	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 14.0 https://github.com/Axivox/axivox-odoo.git axivox-odoo
git commit -m "Axivox"
git push
```

The screenshot shows the GitHub repository page for **Axivox / axivox-odoo**. The repository is public and has 7 branches and 0 tags. The current branch is 14.0, which is 2 commits ahead of and 9 commits behind the 17.0 branch. The repository contains a README, a LICENSE, and two submodules: **axivox** and **axivox_softphone**. The README describes the project as "Axivox for Odoo" and provides a table of modules and their functions.

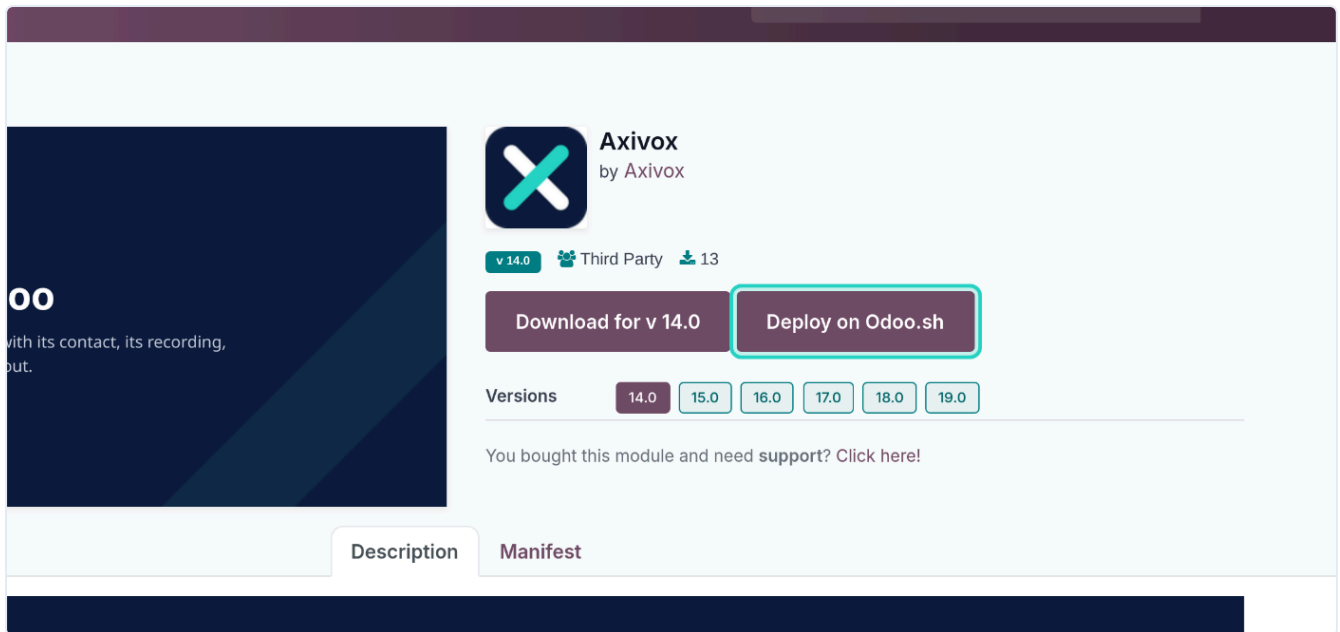
Module	What it does
axivox	The call log Odoo Community does not have: every call, with its contact, its recording, and the transcription and analysis when your account has them.
axivox_softphone	A softphone inside Odoo, in a collapsible panel. Click a number to call it; the panel opens by itself on an incoming call.

The repository also shows a list of contributors (NudjiBe and cyrille-axivox) and a language usage chart (JavaScript 49.5%, HTML 22%, Python 19.6%, SCSS 8.9%).

The public repository, on branch 14.0. 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 14.0. 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

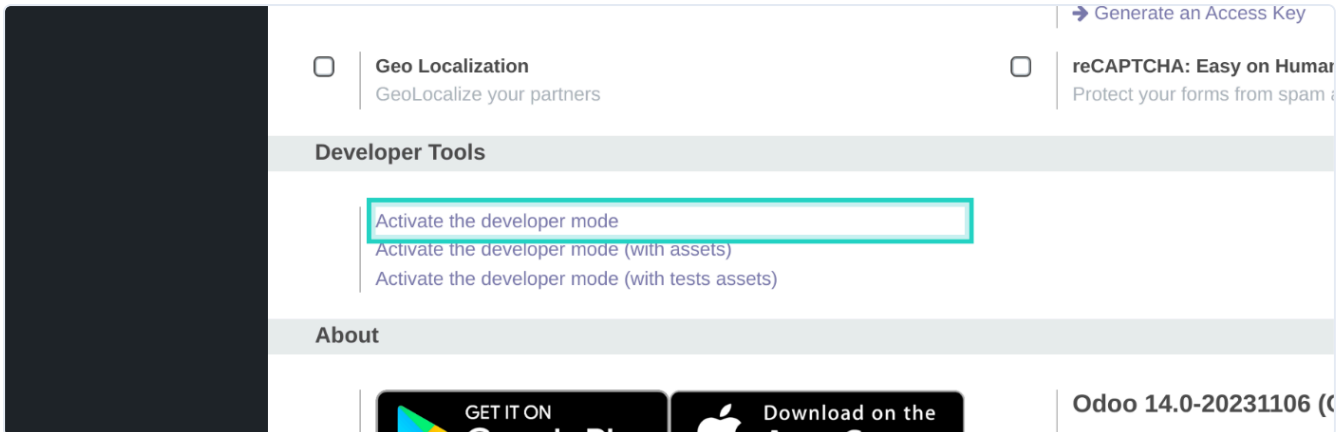
Odoo.sh rebuilds the branch as soon as the commit arrives. Wait for the build to finish and turn green before going further: while it runs, the module is not on the server yet.

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**.



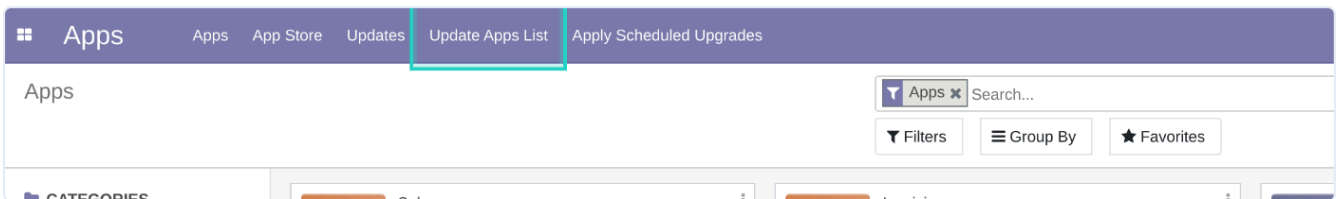
The link sits at the bottom of the general settings, under **Developer Tools**.

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.

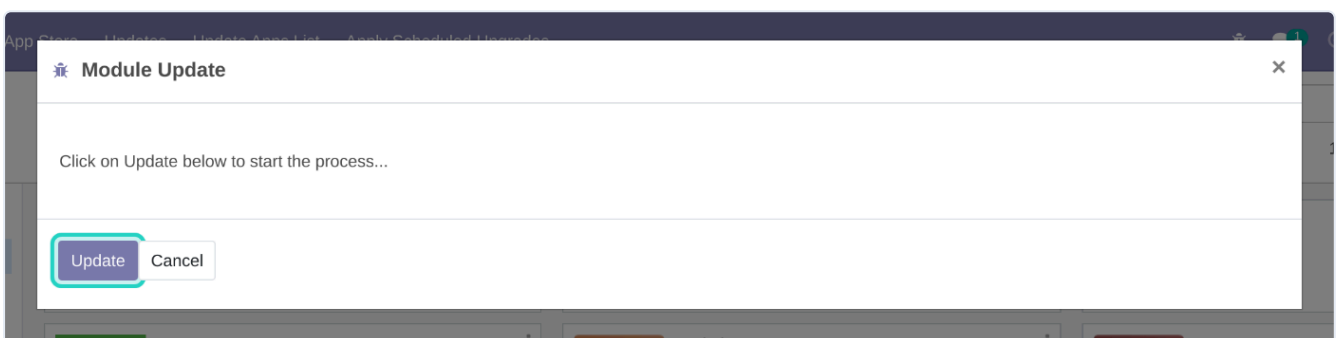
STEP 4 Install the module from Odoo

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



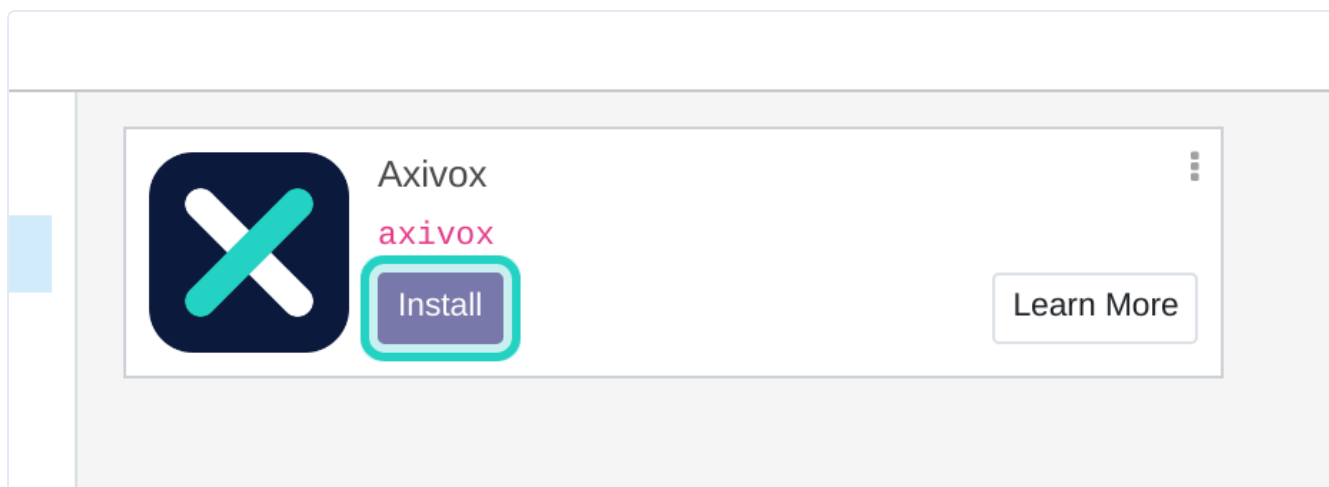
Update Apps List, in the top bar.

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 **Install**.



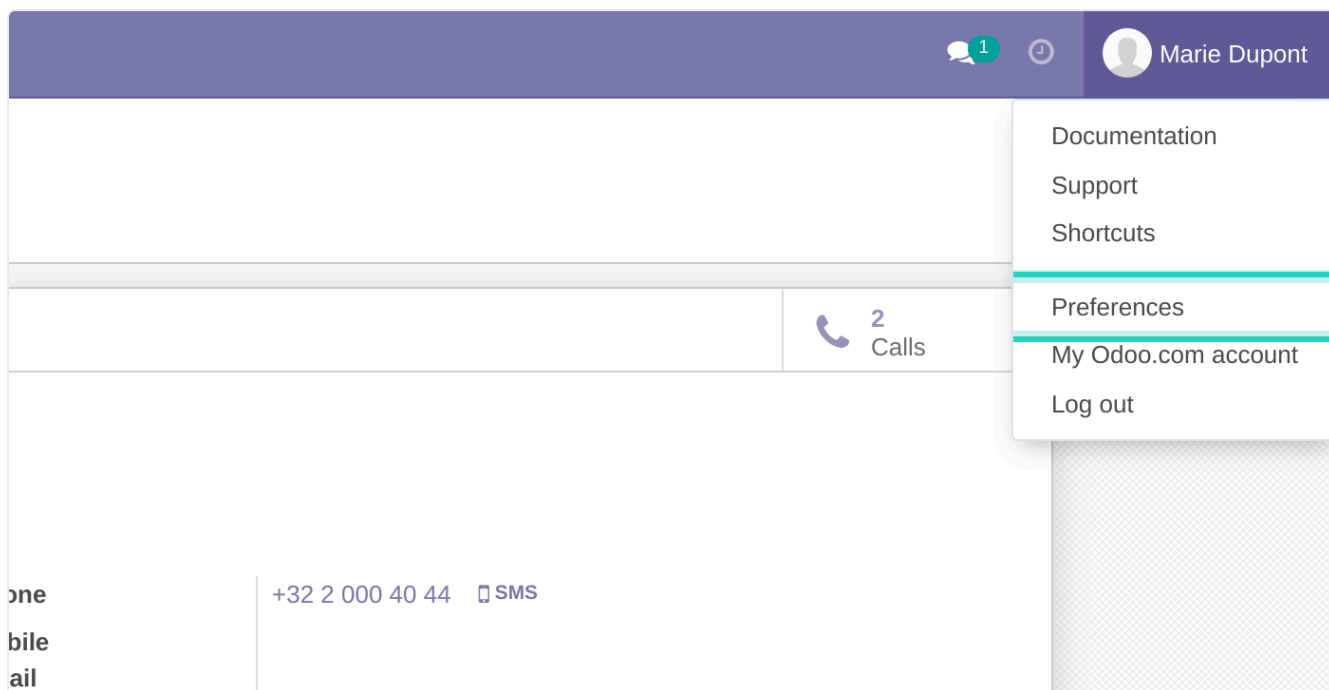
The module card in the apps list.

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

STEP 5 Create an API key in Odoo

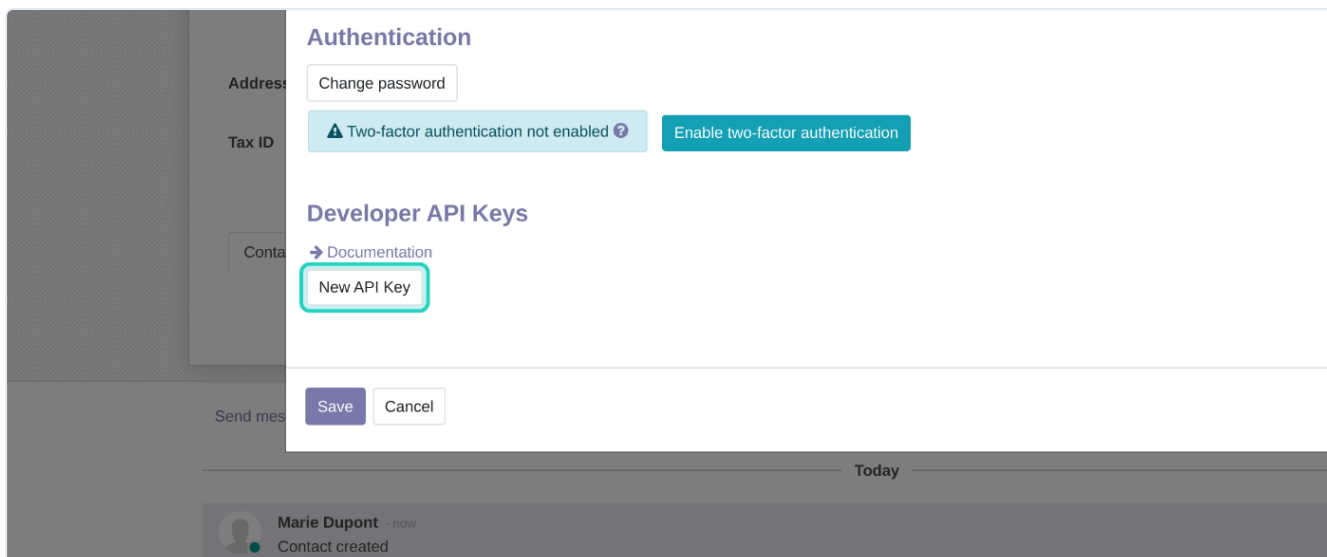
This key lets Axivox write calls into your Odoo. Create it on the account you want to appear as the author of what Axivox writes. An account dedicated to the integration works well.

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



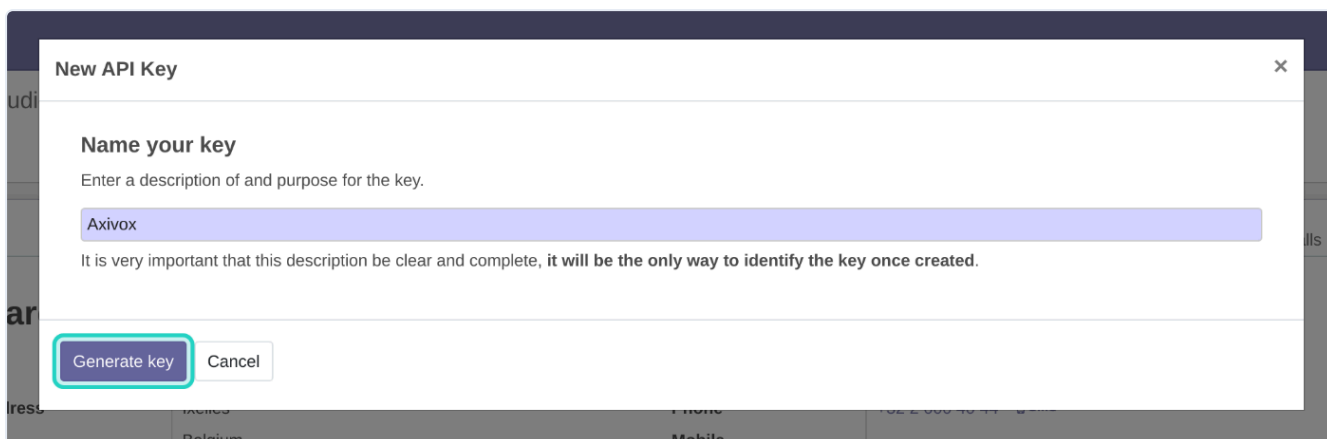
The account menu, at the top right of the screen.

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



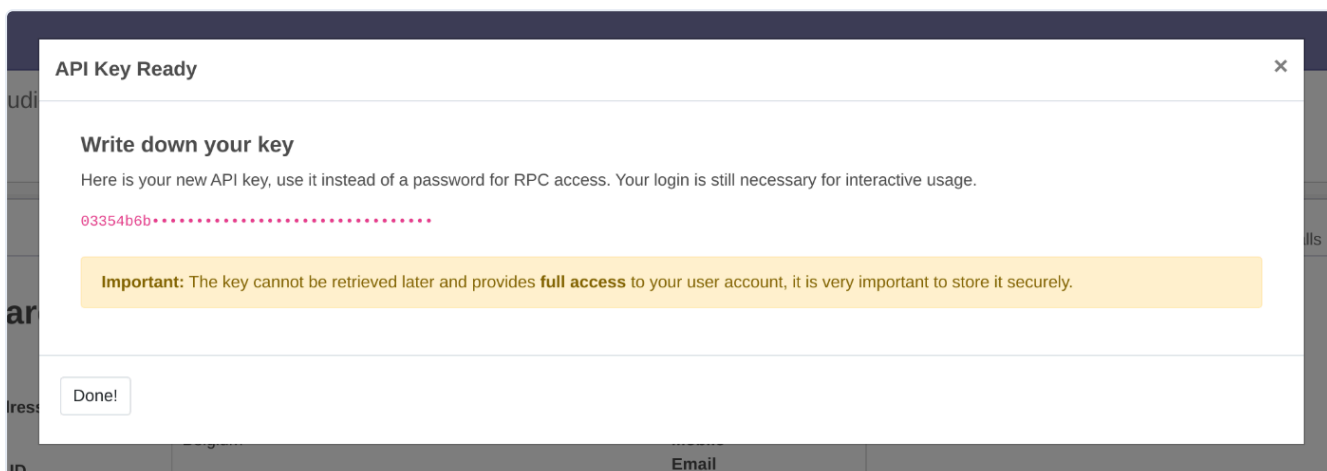
The **Account Security** tab and the button that creates the key.

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.

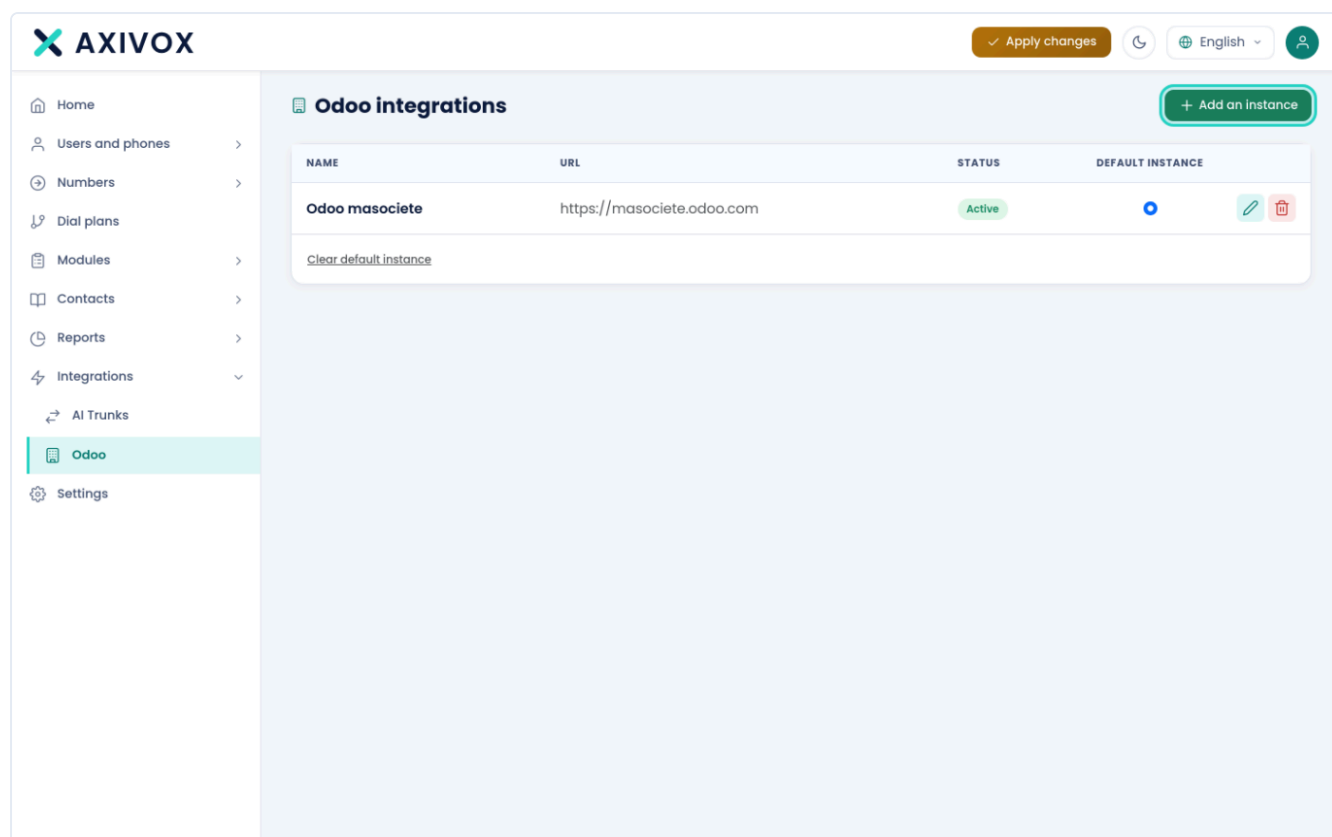
The key is shown once, and only once. Copy it straight away. If you lose it, you have to create another one. It gives full access to the account it was created on, so treat it as a password.



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 <code>https://mycompany.odoo.com</code> .
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.

Apply changes

English

Home

Users and phones

Numbers

Dial plans

Modules

Contacts

Reports

Integrations

AI Trunks

Odoo

Settings

New Odoo instance

Connection

NAME

Our Odoo

INSTANCE URL

https://masociete.odoo.com

Full address, e.g. https://mycompany.odoo.com

LOGIN

marie.dupont@masociete.be

API KEY

.....

Generated in Odoo: Preferences → Account Security → API Keys.

DATABASE

masociete

List databases

Technical database name (often different from the subdomain). Visible via /web/database/list.

Test connection

Features

Active instance

Disable to suspend all synchronisation with this instance.

Identify callers

Look up the contact for every inbound call, including calls that do not go through a dial plan. The name found is kept, so later calls from the same number are instant.

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

Click **Test connection**.

Dial plans

Modules

Contacts

Reports

Integrations

AI Trunks

Odoo

Settings

Our Odoo

INSTANCE URL

https://masociete.odoo.com

Full address, e.g. https://masociete.odoo.com

LOGIN

marie.dupont@masociete.be

API KEY

.....

Generated in Odoo: Preferences → Account Security → API Keys.

DATABASE

masociete

List databases

Technical database name (often different from the subdomain). Visible via /web/database/list.

Test connection

Connection successful (uid 2)

Edition **Community** 14

Call log **axivox.call**

phone_number

direction

state

start_date

end_date

partner_id

user_id

summary

transcript

transcription_status

did

talk_duration_s

sentiment

escalation

themes

Features

Active instance

Disable to suspend all synchronisation with this instance.

Identify callers

Look up the contact for every inbound call, including calls that do not go through a dial plan. The name found is kept, so later calls from the same number are instant. The lookup gets half a second before the call carries on: on a first call from an unknown number the name may therefore be missing, and only show from the next one.

Show the contact name

Replace the presented name with the identified contact name, on the ringing extension and in the call history.

The test passing on an Odoo 14. The edition, the version and the call log found are shown, then the list of the fields Axivox will be able to fill.

Installing the Axivox suite in Odoo.sh 14

Axivox · page 8/13

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.

Users and phones

Numbers

Dial plans

Modules

Contacts

Reports

Integrations

AI Trunks

Odoo

Settings

Connection

NAME

Our Odoo

INSTANCE URL

https://masociete.odoo.com

Full address, e.g. https://masociete.odoo.com

LOGIN

marie.dupont@masociete.be

API KEY

.....

Generated in Odoo: Preferences → Account Security → API Keys.

DATABASE

masociete

List databases

Technical database name (often different from the subdomain). Visible via /web/database/list.

Test connection

Connection successful (uid 2)

Edition Community 14

Call log axivox.call

phone_number

direction

state

start_date

end_date

partner_id

user_id

summary

transcript

transcription_status

did

talk_duration_s

sentiment

escalation

themes

Features

Active instance

Disable to suspend all synchronisation with this instance.

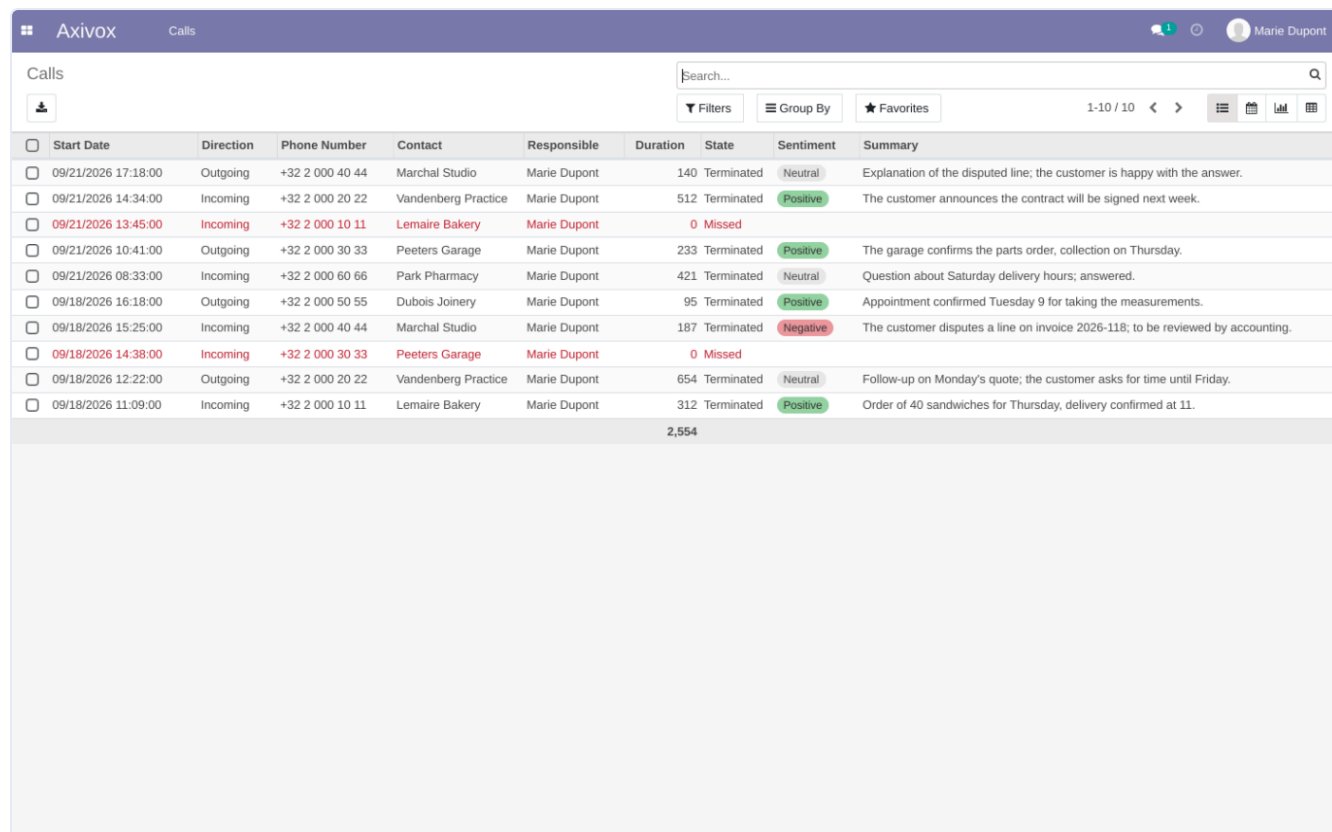
Identify callers

Look up the contact for every inbound call, including calls that do not go through a dial plan. The name found is kept, so later calls from the same number are instant...

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 Number	Contact	Responsible	Duration	State	Sentiment	Summary
☐	09/21/2026 17:18:00	Outgoing	+32 2 000 40 44	Marchal Studio	Marie Dupont	140	Terminated	Neutral	Explanation of the disputed line; the customer is happy with the answer.
☐	09/21/2026 14:34:00	Incoming	+32 2 000 20 22	Vandenberg Practice	Marie Dupont	512	Terminated	Positive	The customer announces the contract will be signed next week.
☐	09/21/2026 13:45:00	Incoming	+32 2 000 10 11	Lemaire Bakery	Marie Dupont	0	Missed		
☐	09/21/2026 10:41:00	Outgoing	+32 2 000 30 33	Peeters Garage	Marie Dupont	233	Terminated	Positive	The garage confirms the parts order, collection on Thursday.
☐	09/21/2026 08:33:00	Incoming	+32 2 000 60 66	Park Pharmacy	Marie Dupont	421	Terminated	Neutral	Question about Saturday delivery hours; answered.
☐	09/18/2026 16:18:00	Outgoing	+32 2 000 50 55	Dubois Joinery	Marie Dupont	95	Terminated	Positive	Appointment confirmed Tuesday 9 for taking the measurements.
☐	09/18/2026 15:25:00	Incoming	+32 2 000 40 44	Marchal Studio	Marie Dupont	187	Terminated	Negative	The customer disputes a line on invoice 2026-118; to be reviewed by accounting.
☐	09/18/2026 14:38:00	Incoming	+32 2 000 30 33	Peeters Garage	Marie Dupont	0	Missed		
☐	09/18/2026 12:22:00	Outgoing	+32 2 000 20 22	Vandenberg Practice	Marie Dupont	654	Terminated	Neutral	Follow-up on Monday's quote; the customer asks for time until Friday.
☐	09/18/2026 11:09:00	Incoming	+32 2 000 10 11	Lemaire Bakery	Marie Dupont	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.

calls Axivox Calls Marie Dupont

Calls / +32 2 000 40 44

Action 1 / 10

+32 2 000 40 44

Contact	Marchal Studio	Start Date	09/21/2026 17:18:00
Responsible	Marie Dupont	End Date	09/21/2026 17:20:20
Direction	Outgoing	Duration	140
Axivox Number	+32 2 000 00 00	Talk Duration	118
State	Terminated	Duration (hours)	00:02
		Transcription Status	Done

Analysis

Sentiment	Neutral
Escalation	☐
Suggested Themes	invoice

Summary

Explanation of the disputed line; the customer is happy with the answer.

Transcript

Agent: Hello, I am coming back to you about invoice 118.
 Caller: Ah yes, thank you.
 Agent: The line is the site visit of 3 September.
 Caller: All right, that is clear now.

Send message Log note 0 Following 1

Today

Marie Dupont - now

One call in detail.

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

Customers / Marchal Studio

Edit Create Action 1 / 1

Marchal Studio

Address	Ixelles Belgium	Phone	+32 2 000 40 44 SMS
Tax ID		Mobile	
		Email	
		Website Link	
		Tags	

2 Calls

Contacts & Addresses

Sales & Purchase

Internal Notes

Send message Log note Schedule activity 0 Following 1

Today

Marie Dupont - now
Contact created

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 14 in English.