Ciotto

Feature guide

The browser that works for you

Contents: every feature of Ciotto, one after another. For each one: its name, where it is in the interface and what it does, in a few lines.

Document version: 1.1 — November 7, 2026

Reference Ciotto version: 0.5.0

Platforms covered: desktop Windows / macOS / Linux, Android, iOS, the companion Web app, Chromium / Firefox / Safari extensions.

Command names are given in English because that is how the application shows them when the interface language is English. With the interface set to another language, the same items appear translated in the same place.

This guide is translated from the Italian original with the help of a language model. If a passage is unclear, the original prevails.

How to read this document

Each feature is a numbered entry made up of three parts:

- **Name** — what the feature is called in the product.
- **Where to find it** — the exact path in the interface (menu, button, shortcut), and which platforms it is available on.
- **What it does** — two or three lines in plain language.

Conventions used in the paths:

- ⋮ is the menu button at the far right of the toolbar. On desktop many items sit in submenus: ⋮ → **Tools** → **Open a List of URLs…** means opening ⋮, then the **Tools** submenu, then the item. The submenus are **History**, **Favorites and reading list**, **Notes**, **Zoom**, **Page**, **Layout**, **Workspaces**, **Profiles**, **Macros**, **Tools**, **Privacy and identity**, **Appearance and language**, **Customise Toolbar** and **Extensions**, plus **Release Channel** in public builds only.
- On the phone, ⋮ opens a pop-up sheet: at the top a row of five quick actions, then the groups **Page**, **Workspaces and layout**, **Tools** and **Privacy and identity**, which open one at a time when you tap their title. In the sheet, items are sometimes written differently than on the desktop (**New tab** instead of **New Tab**). A long press on a tab opens the tab's sheet, the equivalent of a right-click.
- **Menu bar** means the native system menu (File, Edit, View, Layout, History, Extensions, Window, Help), present on desktop only.
- **Settings** means the window opened from ⋮ → **Settings…** (or `Ctrl/Cmd+,`), divided into fourteen sections gathered in four groups. On the phone it opens full screen and closes with **Done**.
- Shortcuts are written as `Ctrl/Cmd+…`: Ctrl on Windows and Linux, Cmd on macOS.

Index of areas

Area	Entries	What it covers
A. Startup, window and interface	1–10	How the application opens, start page, theme, language, accessibility, the Discover Ciotto page, the see-through window
B. Tabs	11–28	Tab life cycle, names, locking, groups, moving tabs, link previews, suspension
C. Address bar, navigation and search	29–35	Navigation, local suggestions, search engines, zoom, site information
D. Split view and panels	36–41	Several pages in the same window, draggable borders, drawing the layout

E. Identities and workspaces	42–49	Profiles, workspaces, projects, private context, session restore
F. History and Memory	50–58	Timeline, browsing maps, memory of what you have read, clearing data
G. Working tools	59–71	Scheduled research, URL lists, questions to several tabs, parked tabs, favorites, reading, media, notes
H. Macros and automation	72–78	Recording, editing and running macros, scheduled runs, data sheets
I. AI assistant	79–82	Sidebar, providers and models, page translation
J. Voice	83–85	Commands spoken out loud, spoken answers, tasks handed to the agent
K. Privacy and security	86–110	The shield, the core protections, the full catalog, the signed lists, burn everything
L. Passwords and credentials	111–121	Vault, autofill, addresses and cards, password health, TOTP, passkeys
M. Lightning identity	122–125	Local key, sign-in without a password, WebLN and Nostr
N. Encrypted sync and tab transfer	126–137	Devices, contacts, transferring tabs and the session, categories
O. Chrome extensions	138–141	Web Store, unpacked extensions, action bar, extensions per profile
P. Downloads, files and printing	142–150	Downloads, local files, saving pages, printing, PDF, capture, web apps
Q. Settings, updates and distribution	151–159	The fourteen sections, shortcuts, import, updates, Help menu, package protection
R. Platform differences	160–165	What changes on Android, iOS, Web and extensions, and the phone interface

A. Startup, window and interface

1. Starting the application and the main window

> **Where to find it:** the Ciotto icon (a light “c” carved from a pebble, on a dark square with rounded corners) in the Start menu, in the Dock, in the Android app drawer or on the iOS Home Screen.

> **What it does:** opens a window with a tab bar, an address bar, a toolbar and the page area. Pages are not HTML frames: they are native views placed over the interface by the main process.

Platforms: all native platforms (desktop, Android, iOS).

2. Startup modes

> **Where to find it:** Settings → **New tab & startup** → **On startup**.

> **What it does:** decides what Ciotto shows when it opens: carrying on where you left off, opening a new tab, or opening a list of pages entered by hand.

3. Start page and customizable blocks

> **Where to find it:** the page that appears in every new tab. The blocks are switched on in Settings → **New tab & startup**, with the switches **New tab: favorites**, **New tab: frequent**, **New tab: projects**, **New tab: reading** and **New tab: remote tabs**; in the same list, **New tab: Points of View** brings back the suggested site if you removed it.

> **What it does:** shows the Ciotto logo, the word “ciotto” and the claim “The browser that works for you” (“Il browser che lavora per te” in Italian), with no news feeds. The five blocks (favorites, frequent sites, projects, reading list, tabs from other devices) are off until you switch them on. By default there are only two suggestions: the **Discover Ciotto** button in the top corner and the tile of a suggested site, **Points of View**, which opens in the same tab. Each has its own ×, which removes it for good on that device. The list of suggested sites and their icon are built into the program: to draw the page, Ciotto asks nothing of any server.

Platforms: desktop, Android and iOS.

4. Home button and home page

> **Where to find it:** Settings → **New tab & startup** → **Show Home button** and **Home page**. The button then appears in the toolbar, between Back and Forward. On Windows and Linux the same action has the shortcut `Alt+Home`.

What it does: adds a button that takes you back to the chosen page. If the address is left empty, the button opens a new tab.

5. Light, dark or system theme

Where to find it: ⋮ → **Appearance and language** → **System** / **Light** / **Dark**, or Settings → **Appearance & accessibility** → **Theme**.

What it does: switches the look of Ciotto's interface between light, dark and "same as the operating system". The choice is a portable preference: it also reaches your other synced devices.

6. Interface language

Where to find it: ⋮ → **Appearance and language**, in the list below the three themes (each language is written in its own name), or Settings → **General** → **Interface language**.

What it does: changes the language of menus, panels and messages. There are sixteen languages: English, Italian, French, Spanish, German, Simplified Chinese, Russian, Korean, Portuguese, Japanese, Arabic, Swahili, Hausa, Yoruba, Amharic and Klingon (tlhIngan Hol). In Arabic the interface is laid out from right to left. The language is a portable preference and follows you to your other synced devices.

7. Accessibility: minimum zoom, contrast, motion, focus

Where to find it: Settings → **Appearance & accessibility**.

What it does: brings together **Default page zoom**, **Minimum font size**, **Contrast** (System, Normal or More contrast), **Reduce motion**, **Animate images and GIFs**, **Caret browsing** and **Always show keyboard focus**. The settings apply both to Ciotto's interface and to the pages it loads.

8. Favorites bar

Where to find it: Settings → **New tab & startup** → **Show favorites bar**. The bar appears below the address bar.

What it does: shows, in a row that is always visible, your personal favorites and those of the workspace in use. Folders become buttons with a drop-down menu (subfolders open to the side), and **Manage** at the end of the row opens the saved items page. Clicking an item opens it in a new tab.

9. Discover Ciotto

> **Where to find it:** ⋮ → **Discover Ciotto**; menu bar → **Help** → **Discover Ciotto**; the **Discover Ciotto** button in the corner of the start page. On the phone: ⋮ sheet → **Discover Ciotto**.

> **What it does:** an internal page, opened in a tab, that describes what Ciotto can do: one card per feature, grouped into fifteen areas. Each card has a title, a sentence, a **How it works** item with the instructions and a real case, and, where it can, the **Take me there** button, which opens the panel, the Settings section or the command the card is about. On desktop the shortcut assigned at that moment appears next to the title. At the top, a field searches titles and sentences, and a row of labels leads to each area. Features the platform does not have are not shown; under the privacy catalog card is the list of protections available on that device.

Platforms: desktop, Android and iOS. On the phone there are no shortcuts, and no buttons for commands that do not exist there.

10. Looking through Ciotto

> **Where to find it:** a button in the toolbar (titled **See through Ciotto and click below**); menu bar → **View** → **See through Ciotto and click below**; ⋮ → **Appearance and language** → **See through Ciotto and click below**.

> **What it does:** makes the window semi-transparent, keeps it above the other applications and lets clicks and scrolling pass through to the application underneath: you keep an eye on a page while you work elsewhere. The tab strip and the toolbar stay opaque and usable, and in place of the button the notice **Click-through mode** appears, with the key combination for going back. You return to the normal window with `Ctrl/Cmd+Shift+U`, a system-wide shortcut; if another application already uses it, the mode does not start and Ciotto says so.

Platforms: Windows and macOS. On Linux the command does not appear, because there the window cannot become semi-transparent; it does not exist on the phone.

B. Tabs

11. Opening, closing and switching tabs

> **Where to find it:** the tab strip at the top; the **+** button; ⋮ → **New Tab**; `Ctrl/Cmd+T` to open, `Ctrl/Cmd+W` or a middle-click on the tab to close; `Ctrl/Cmd+1…8` goes to the tab in that position and `Ctrl/Cmd+9` to the last one. On the phone: ⋮ sheet → **New tab**, and the button showing the number of tabs opens the list of all of them.

> **What it does:** the basic life cycle of a tab. Each panel has its own strip: the one shown always belongs to the panel that has focus. On desktop the ✕ of an inactive tab appears when you hover over it.

12. The “+” button next to the last tab

> **Where to find it:** in the tab strip, right after the last tab. There is nothing to switch on.

> **What it does:** the button for opening a tab follows the last tab instead of sitting against the edge of the window: tabs have a width of their own and do not stretch to fill the row. When there are many tabs and the row has to scroll, the button **anchors itself to the edge** of the row instead of going off screen: in neither case do you have to cross the strip to open a tab. The empty space left over still drags the window. On the phone, where the strip shows only icons, the “+” and the button that closes the active tab stay fixed next to the row.

13. Reopening a closed tab

> **Where to find it:** `Ctrl/Cmd+Shift+T`; menu bar → **File** → **Reopen Closed Tab**; ⋮ → **Reopen Closed Tab**; right-click a tab → **Reopen Closed Tab**. On the phone: ⋮ sheet → **Reopen closed tab**, the same item in the tab's sheet, or **Undo** in the **Tab closed** notice that appears for a few seconds after a tab is closed.

> **What it does:** puts the last closed tab back at its address, with its Back/Forward history. Each window remembers its last twenty-five closed tabs, even after a restart, and the record works even with Memory turned off.

14. Duplicating a tab

> **Where to find it:** right-click a tab → **Duplicate tab**. On the phone: long-press a tab → **Duplicate tab**.

> **What it does:** opens a second tab at the same address, next to the original, with the same identity.

15. Giving a tab a name

> **Where to find it: double-click** a tab's title; or right-click the tab → **Rename Tab…**. On the phone: long-press the tab → **Rename tab**.

> **What it does:** a tab's title is the page's own, and the page changes it whenever it likes: with ten tabs open on the same site, the strip becomes the same truncated word ten times over. A tab can instead carry the **name you give it**. The name does not replace the title, it sits next to it: the page keeps saying what it is called, and **emptying the field** gives back the real title, already updated to wherever the tab has navigated in the meantime. **Navigating does not erase the name**: the point of naming a tab is to tell it apart from many others, and losing the name at the first click would make no sense. The name survives restarts, projects and duplication. A pinned tab does not show its title, so you rename it from the menu and read the name in the tooltip.

16. Pinning a tab

> **Where to find it:** right-click a tab → **Pin Tab** / **Unpin Tab**; can be assigned to a shortcut (**Pin or unpin tab**). On the phone: long-press the tab → **Pin tab**.

> **What it does:** fixes the tab at the start of the strip, in compact form and without a ×, so it does not get lost among the others. A pinned tab is never put to sleep, and bulk closes skip it.

17. Locking a tab

> **Where to find it:** right-click a tab → **Lock Tab** / **Unlock Tab**. On the phone: long-press the tab → **Lock tab**. Can also be assigned to a shortcut (**Lock or unlock tab**, entry 152), with no default combination.

> **What it does:** a locked tab refuses the normal ways of being closed: the ×, the middle-click, `Ctrl/Cmd+W`, the menu item, an extension, a macro node. A **padlock** appears next to the name and the × is no longer drawn, because a button that refuses when you press it is worse than no button at all. Bulk closes (close others, close to the right, burn everything) skip it, and putting it off until later is refused with a message, instead of recording a postponement that closes nothing. **A stated limit:** closing the window or deleting the workspace takes it away all the same. The padlock protects a tab from a wrong click; it does not hold a window hostage.

18. Muting a tab and the audio indicator

> **Where to find it:** the ♪ button on the tab that is playing sound (it becomes × when the tab is muted); right-click the tab → **Mute Tab** / **Unmute Tab**; can be assigned to a shortcut (**Mute or unmute tab**). On the phone: long-press the tab → **Mute tab**.

> **What it does:** shows which tab is producing audio and lets you silence it without stopping playback.

19. Closing the other tabs, or those to the right

> **Where to find it:** right-click a tab → **Close Other Tabs** / **Close Tabs to the Right**. On the phone: long-press the tab → **Close other tabs** / **Close tabs to the right**.

> **What it does:** multiple closes limited to the panel of the tab you clicked: it does not touch other panels or other workspaces, and it skips pinned and locked tabs.

20. Tab groups

> **Where to find it:** right-click a tab → **Add to New Group**, **Move to Group** (appears when the panel already has a group), **Remove from Group**. On the phone: long-press the tab → **Add to new group**, **Collapse group** / **Expand group**, **Remove from group**.

> **What it does:** gathers several tabs under a colored label in the strip. A new group is called **Group 1**, **Group 2** and so on, always in the same color: for now, the name and color cannot be changed from the interface. Clicking the label collapses the group, leaving only its active tab visible. Groups belong to the panel and come back with session restore.

21. Selecting several tabs

> **Where to find it:** the tab strip, holding down Ctrl (Cmd on macOS) while clicking. Desktop only.

> **What it does:** puts several tabs of the same panel into a selection, highlighted in the strip. For now the selection is used for groups: **Add to New Group**, **Move to Group** and **Remove from Group**, chosen from the menu of a selected tab, apply to all the selected tabs. The other menu items act only on the tab you clicked.

22. Moving a tab between panels

> **Where to find it:** right-click a tab → **Move to Panel 1**, **Move to Panel 2**... (one item for each other panel); or by dragging it. On the phone: long-press the tab → **Move to other panel**, or the same drag with your finger.

> **What it does:** moves the tab to another pane of the window **without changing its identity**: whoever was signed in stays signed in. While you drag a tab, a row of destinations appears in the strip: the numbers of the other panels, ▸ **+** to open it in a new panel alongside, ▾ **+** for a new panel below, and the names of the other workspaces (entry 23). The destinations are in the strip because the pages cover the interface. The same menu also moves the tab within the row: **Move Tab Left**, **Move Tab Right**, **Move Tab to Start**, **Move Tab to End**.

23. Moving a tab to another workspace

> **Where to find it:** ⋮ → **Workspaces** → **Move This Tab To...** (for the tab in the foreground); or by dragging a tab onto the name of another workspace in the row of destinations that appears

while dragging (entry 22).

What it does: takes the tab to another workspace. If the destination workspace browses with a different identity, the tab is reopened at the same address with that identity. Workspaces already open in another window are not offered.

24. Link preview

Where to find it: Settings → **Tabs & performance** → **Preview links on Shift+click** to switch it on; then **Shift+click** a link, or right-click the link → **Preview Link**. Off by default.

What it does: opens the link in a **floating frame** centered on the window, instead of opening a tab. At the top, a thin strip names the site and offers two buttons: **Open as tab**, which turns the preview into a real tab in the window behind, and **Close preview**. It also closes with Esc and by clicking elsewhere: a preview that survives the next click is not a preview, it is one more window to tidy up. The frame browses in the **profile of the tab the click came from**, so a site you are signed in to is seen signed in; and if you have followed something inside the frame, **Open as tab** opens **the page you are looking at**, not the one you started from. One preview per window: a second click replaces the first. **Why it is off by default:** normally Shift+click opens a tab, and silently changing what a gesture you have already learned does would be a surprise, not a feature.

Platforms: desktop only.

25. Horizontal or vertical tab strip

Where to find it: Settings → **Tabs & performance** → **Tab strip**. Desktop only.

What it does: moves the strip to the left side, vertically, where titles stay readable even with many tabs.

26. Tabs that fall asleep

Where to find it: Settings → **Tabs & performance** → **Suspend inactive tabs**, **Suspend after (minutes)** and **Never suspend**.

What it does: pauses tabs that have been left idle, to free up resources. It is on by default, with a wait of 30 minutes (from 5 to 1440). It never touches a pinned tab, a tab playing sound, or one with a filled-in form, media playing, a video call in progress or an active download. The **Never suspend** list excludes whole sites, one name per line. In the strip a sleeping tab has a dimmed icon with a small moon, and its tooltip says **Asleep to save memory — opens instantly**. On desktop, when the machine is short of memory, the wait is skipped: all the other conditions still apply.

27. Giving memory back (discarded tab)

> **Where to find it:** Settings → **Tabs & performance** → **Give the memory back after (minutes)**.

> **What it does:** after an additional period asleep (60 minutes by default, counted from when the tab fell asleep), the tab gives up its process entirely, and only its address, title, icon and history remain. When you open it again, the page reloads: the tab's tooltip says so with **Asleep to save memory — loads again when opened**. A value of 0 leaves the tab asleep indefinitely.

28. Behavior on battery

> **Where to find it:** automatic, no command. Desktop; on Android and iOS the equivalent is the response to the system's low-memory warning.

> **What it does:** when the laptop is running on battery, the waits before putting a tab to sleep and before giving its memory back get shorter (to a third and to half respectively, never below one minute). It does not switch the feature back on if you have turned it off, and a value of 0 for giving memory back stays 0.

C. Address bar, navigation and search

29. Address bar and navigation

> **Where to find it:** the central field of the top bar; `Ctrl/Cmd+L` moves focus to it. The Back, Forward, Reload and Stop buttons are to its left.

> **What it does:** opens addresses, starts searches and shows where you are. Reload and Stop are the same button, which changes depending on whether the page is loading or not. On the phone the Reload/Stop button is inside the field, on the right, and at rest the field shows only the site's name.

30. Omnibox suggestions, computed locally

> **Where to find it:** the list that appears below the bar as you type.

> **What it does:** suggests tabs that are already open, favorites, reading list items, tabs from other devices and history. The ranking happens entirely on this machine: no letter you type is sent to a suggestion service.

31. Choosing the search engine

> **Where to find it:** the question appears on its own above the results the **first time you search** from the address bar. After that: Settings → **Search** → **Default search engine**.

> **What it does:** Ciotto searches with DuckDuckGo until you tell it otherwise. At the first search, a bar says **Searching with DuckDuckGo. Use another one?** and offers DuckDuckGo, Google, Bing, Brave Search, Ecosia and Startpage, plus **Keep DuckDuckGo**; the search you already asked for runs anyway, without waiting for the answer. The question is asked only once per device, whatever the answer.

Platforms: desktop, Android and iOS.

32. Custom search engines with a keyword

> **Where to find it:** Settings → **Search**, the box for adding one, with name, keyword and address template.

> **What it does:** lets you define your own engine, with an address template that contains `%s` in place of the words searched for, called up by a keyword that starts with `@`.

33. Find in page

> **Where to find it:** `Ctrl/Cmd+F`; ⋮ → **Find in Page…**; menu bar → **View** → **Find in Page…**. Next and previous match also with `Ctrl/Cmd+G`, `F3` and `Ctrl/Cmd+Shift+G`. On the phone: ⋮ sheet → **Page** → **Find in page**.

> **What it does:** searches for text in the open page, highlights the matches and shows which one you are on out of the total.

34. Page zoom

> **Where to find it:** `Ctrl/Cmd+` and `Ctrl/Cmd-` to zoom in and out, `Ctrl/Cmd+0` to go back to the default zoom; ⋮ → **Zoom (100%)**, a submenu that shows the current percentage, with **Zoom In**, **Zoom Out** and **Actual Size**; menu bar → **View**. The list of remembered sites is in Settings → **Appearance & accessibility** → **Per-site zoom**. On the phone: ⋮ sheet → **Page** → **Zoom in** / **Zoom out** / **Reset zoom**.

> **What it does:** enlarges or reduces the page, and remembers the level per site: the same site, reopened in any tab, starts from there. When the level differs from the default, the percentage appears in the toolbar, and clicking it goes back to the default. In the Settings list each site has its own ✕ to forget it. On the phone the zoom items leave the sheet open and show the percentage at every step.

35. Site information

> **Where to find it:** the padlock to the left of the address, in the bar (open, and colored, when the page is on `http://`).

> **What it does:** opens a popover that says whether the connection is encrypted, which certificate protects it, which protections are active on that site, which permissions it has been granted (Ask, Allow or Block, saved for the workspace) and how many cookies the site keeps in the profile. Permissions are changed from there. On Android and iOS the popover shows the connection and the permissions, without the certificate or the list of protections: camera, microphone and location can be changed, the other permissions show as blocked.

D. Split view and panels

36. Splitting the window

> **Where to find it:** `Ctrl/Cmd+Shift+D` splits to the right, `Ctrl/Cmd+Shift+E` splits downward; menu bar → **Layout** → **Split Right** / **Split Down**; ⋮ → **Layout** → **Split Right** / **Split Down**. On the phone: ⋮ sheet → **Workspaces and layout** → **Split screen**.

> **What it does:** puts several pages in the same window. Splitting opens a copy of the page you started from, so comparing two places in the same document takes one shortcut. On the phone there are at most two panels, and the split follows the shape of the screen: one above the other in portrait, side by side in landscape.

37. Draggable borders

> **Where to find it:** the gutter that separates two panels, directly with the mouse.

> **What it does:** every pair of panels is separated by a gutter you can grab: dragging it resizes the pages as you drag.

38. Layout drawing mode

> **Where to find it:** `Ctrl/Cmd+Shift+L`; ⋮ → **Layout** → **Draw Layout…**; menu bar → **Layout** → **Draw Layout…**. The command can be pinned to the toolbar (⋮ → **Customise Toolbar** → **Draw layout**, entry 154): by default it is not in the toolbar.

> **What it does:** freezes every page into a snapshot and hides the live views, so that you can draw over them. A vertical stroke splits the panel into left and right, a horizontal one into top and bottom. An ✕ appears on every border, which folds the two panels back into one.

Platforms: desktop only.

39. Preset layouts

> **Where to find it:** menu bar → **Layout**, or ⋮ → **Layout**: **Single Pane**, **Two Columns**, **Two Rows**, **Grid 2×2**.

> **What it does:** applies a ready-made arrangement in one go, keeping the identity each tab was opened with. The four layouts can be assigned to shortcuts (**Single pane**, **Two columns**, **Two rows**, **Grid 2×2**), with no default combination.

Platforms: desktop. On the phone, which has at most two panels, the commands do not appear.

40. Closing a panel

> **Where to find it:** `Ctrl/Cmd+Alt+W`; menu bar → **Layout** → **Close Pane**; ⋮ → **Layout** → **Close Panel**. On the phone: ⋮ sheet → **Workspaces and layout** → **Close panel**.

> **What it does:** closes the pane that has focus. Its tabs are not thrown away: they move to the neighboring panel.

41. Numbered panel markers

> **Where to find it:** to the left of the tab strip, when there is more than one panel.

> **What it does:** numbered chips that move focus between panels; when a panel is on a different identity, its chip is filled with that identity's color.

E. Identities and workspaces

42. Profiles (browsing identities)

> **Where to find it:** ⋮ → **Profiles**, a submenu with **This panel browses as** (the list of profiles), **Split Right As…** and **Manage Profiles…**; the same submenu in ⋮ → **Layout** → **This Panel Is…**; a colored chip in the toolbar (it appears as soon as a second profile exists); names can be edited in Settings → **Profiles & workspaces**. On the phone: ⋮ sheet → **Privacy and identity** → **Profiles**.

> **What it does:** a profile is a separate set of cookies, storage and logins. By assigning one to a panel, you can open the same site twice side by side, signed in as two different people; **Split Right As…** directly opens a new panel on another profile. Every profile comes with a name and a color from the start, and that color is how you recognize it without reading: it is carried by the panel outline and by a dot on every tab. From the profiles panel (**Manage Profiles…**) you create a profile with **New profile**, assign it to the panel with **Use here**, decide whether extensions run in it (**Run extensions in this profile**, entry 141) and delete it with **Remove**.

Platforms: desktop. On the phone profiles exist, but the identity is chosen per workspace, not per panel.

43. Clearing a profile's data

> **Where to find it:** ⋮ → **Profiles** → **Manage Profiles…**, the profile's row → **Sign out of everything**. The same cleanup, with more choices, is in Clear browsing data (entry 58) with the **Current profile** scope.

> **What it does:** empties the profile's cookies and storage without deleting the profile, so the identity remains but everything it was signed in to is forgotten. It acts immediately, without asking for confirmation: the button becomes **Signed out**.

44. Workspaces (working environments)

> **Where to find it:** ⋮ → **Workspaces**, a submenu with **This window is showing** (the list of workspaces), **Move This Tab To…** and **Manage Workspaces…**; the colored chip to the left of the tab strip (it appears from the second workspace on) opens the same list; names can be edited in Settings → **Profiles & workspaces**. On the phone: ⋮ sheet → **Workspaces and layout** → **Workspaces**.

> **What it does:** while a profile separates accounts, a workspace separates everything else: its own tabs, its panel arrangement, its own history, its own Memory, its own downloads, its own answers to site requests. A workspace *points to* a profile, so two workspaces can share an account when needed. From the panel (**Manage Workspaces…**) you create a workspace with **New workspace**

and delete it with **Remove**, which closes its tabs and deletes its history after asking you to type **remove**.

45. Changing a workspace's identity

Where to find it: ⋮ → **Workspaces** → **Manage Workspaces…**, the workspace's row → **Browses as**.

What it does: changes the profile the workspace browses with. Its tabs are reopened at the same addresses with the new identity: the previous identity's cookies are not carried over, since carrying them over is exactly what a change of account must avoid. Each row also says whether extensions run in that workspace (**runs the extensions** or **no extensions here**).

46. Projects (saved windows)

Where to find it: ⋮ → **Projects…**; can be pinned to the toolbar. On the phone: ⋮ sheet → **Workspaces and layout** → **Projects**.

What it does: gives a name to a window's arrangement — which pages, split into which panels, each with which identity — and reopens it when needed, in a window of its own, without disturbing what you were doing. The panel offers **Save this window**, and for each project **Open**, **Update**, **Rename** and **Delete**.

47. Private context

Where to find it: `Ctrl/Cmd+Shift+N`; menu bar → **File** → **New private window**. On the phone: ⋮ sheet → **New private tab**. The **Private** label appears in the toolbar while it is active.

What it does: a shared, temporary context, distinct from the “as a stranger” tab. What happens in it is excluded from history, Memory, session restore, projects, sync and tab transfer. In addition, downloads do not stay in the list, a private tab cannot be put off until later, and the Lightning identity is not offered to sites (neither for signing in nor for WebLN and Nostr). Saving, printing or capturing a page asks for confirmation first, because that file ends up outside the private context.

Platforms: desktop, Android and iOS. On Android it needs a system WebView that supports separate profiles; otherwise the private context is not available.

48. Session restore after an abnormal shutdown

Where to find it: a bar that appears at the top of the window on the first launch after an unclean shutdown, when startup is set to **Continue where you left off** (entry 2).

What it does: the session is saved while you browse, so after an abnormal shutdown the windows come back on their own; the bar reports this, with how many tabs came back (**Ciotto closed unexpectedly last time — 5 tabs have been restored.**), and closes with **OK**. The

> message is shown only once, in the first window that asks for it. With the other startup modes nothing is restored.

Platforms: desktop only.

49. Task manager

> **Where to find it:** Settings → **Tabs & performance**, the **Task manager** box at the bottom of the section.

> **What it does:** lists the browser's processes with their real memory and CPU use on desktop. On Android and iOS it reports only the tab's lifecycle state, because WebViews do not expose per-page measurements.

F. History and Memory

50. History: the timeline

> **Where to find it:** `Ctrl+H` (`Cmd+Y` on macOS); menu bar → **History** → **Show Full History**; ⋮ → **History** → **Show Full History**. On the phone: ⋮ sheet → **Page** → **History**.

> **What it does:** lists the pages you have visited, most recent first, grouped by day and then by browsing session (**Timeline** view). You can search it and remove any item. History is kept in a local SQLite database. At the top of the view there are also **Map** (entry 51), **Pause recording** (entry 53) and **Clear all**.

51. Browsing map

> **Where to find it:** the **Map** tab inside the history view; **Expand** on a session; **Open map** on a session and **Show path** on a timeline item. The orientation is chosen with **Sideways** and **Downward**.

> **What it does:** every navigation is recorded together with the page it came from, so a session is a tree: the starting page, everything opened from it, and so on. Following a link creates a child; typing an address or starting a search begins a new tree. Going back and taking another route leaves the abandoned branch where it was.

52. “How did I get here” from the current page

> **Where to find it:** ⋮ → **History** → **Show This Page on the Map**; can be pinned to the toolbar (⋮ → **Customise Toolbar** → **Map of this page**) and assigned to a shortcut. On the phone: ⋮ sheet → **Page** → **Map of this page**. It is not in the menu bar's **History** menu.

> **What it does:** opens the map already centered on the page you are looking at, instead of making you look for it in the timeline.

53. Pausing history recording

> **Where to find it:** menu bar → **History** → **Pause History**; ⋮ → **History** → **Pause History**; the **Pause recording** / **Resume recording** button in the history view; can be assigned to a shortcut.

> **What it does:** stops recording without hiding what has already been recorded. During the pause, not even page text for Memory is captured, not even with **Remember This Page Now**. The paused state is saved and remains after a restart; in the history view a line reminds you that recording is paused.

54. Targeted history deletion

> **Where to find it:** the history view: the ✕ on a **Timeline** item (**Remove … from history**) and the **Clear all** button. A time range is deleted from Clear browsing data (entry 58).

> **What it does:** nothing is deleted automatically. Deletion is always manual: one visit at a time, a time range (last hour, last 24 hours, last 7 days, last 4 weeks) or the workspace's entire history (**Clear all** acts immediately, without asking for confirmation). When no visit to a page is left, its stored text and its entry in the search index go too. After a deletion the index is compacted and the freed space is given back, so the bytes are really gone.

55. Memory: switching it on, and the consent screen

> **Where to find it:** `Ctrl/Cmd+Shift+H`; menu bar → **History** → **Search What You Read…**; ⋮ → **History** → **Search What You Read…**; the **Remember what you read (Memory)** switch is also in Settings → **AI & automations**. On the phone: ⋮ sheet → **Page** → **Search what you read**.

> **What it does:** Memory is the same history, queried the other way round: not “where have I been” but “that page about API prices I read two weeks ago”. **It is off until you switch it on**, and while it is off the view *is* the consent screen, which states in plain words what is kept, what is not, what leaves the machine and how to undo it all, with **Turn on Browser Memory** and **Not now**.

56. Memory: what is captured, and what never is

> **Where to find it:** it works on its own; the rules are stated on the consent screen. ⋮ → **History** → **Remember This Page Now** captures the current page on request (on the phone: ⋮ sheet → **Page** → **Remember this page now**). On a result's card, **Forget this page** and **Never remember this site**.

> **What it does:** a page is read four seconds after it has settled, and only if the tab is still there: this keeps out redirects, interstitials and pages you bounced straight off. Ciotto **never keeps** any page that has a password field (bank, webmail, admin panel, health portal), any sign-in or payment address, any page under 200 characters, anything from a private window or a “stranger” tab, or anything while history is paused, not even on request. A site you are already signed in to does not show a password field: that is why **Never remember this site** exists.

57. Memory: search with an editable plan

> **Where to find it:** the search field in the Memory view; the plan's chips appear below it.

> **What it does:** the search runs entirely on this machine and looks only at the pages read in the workspace in use. What you type is read into a **plan** — the words that matter, a price limit, a period, the type of page — and the plan is shown as chips that can be removed one at a time. “Black shoes under 300 euros” becomes shoes · black · under 300 €, and the price is a real filter,

not a similarity. From a result's card, **Open in a background tab** reopens the page in the background.

58. Clear browsing data

Where to find it: Settings → **Privacy & security** → **Clear browsing data**, above the catalog of protections; `Ctrl/Cmd+Shift+Delete` opens that page. Also available on Android and iOS.

What it does: a procedure in three steps. First, the scope: **Current workspace** (history, download list, decisions on site permissions) or **Current profile** (cookies and sessions, site storage, cached files, saved passwords). Second, the period: last hour, last 24 hours, last 7 days, last 4 weeks or all time; only history and the download list can be limited to a period, the other categories are cleared in full. Third, the categories, each with the number of items that would be deleted. Nothing is preselected. **It never deletes files already downloaded**, only their entry in the list.

G. Working tools

59. Scheduled research

> **Where to find it:** ⋮ → **History** → **Scheduled Research…**; can be pinned to the toolbar. The toolbar button appears anyway as soon as at least one saved search exists. On the phone: ⋮ sheet → **Page** → **Scheduled research**.

> **What it does:** a question you keep asking — a competitor, a price, an article you are waiting for — is run again by Ciotto at intervals: every hour, every six hours, every day or every week. Each search names **one or more engines**, and the results are merged: a page found by both is a single node. Only engines that can be read without a human at the wheel are offered.

60. Scheduled research even with the window closed

> **Where to find it:** the **Scheduled Research** panel, the **Keep searching after the window is closed** option. Desktop only.

> **What it does:** by default, searches stop when Ciotto closes. With this option the application stays in the system tray and keeps to the schedule; the tray icon is both the sign that it is still running and the way to close it. It is off until you ask for it.

61. Opening a list of addresses

> **Where to find it:** ⋮ → **Tools** → **Open a List of URLs…**; can be pinned to the toolbar. On the phone: ⋮ sheet → **Tools** → **Open a list of URLs**.

> **What it does:** takes twenty addresses in one go instead of twenty trips to the address bar. They open as **background tabs, one every so often** (from one second to one minute, as you choose, remembered for next time). A batch stops at a hundred addresses: a longer list is cut off, and it says so. While the batch is running, the toolbar button lets you stop it (**Hold**, **Resume**, **Stop**).

62. Collecting addresses from the page or the selection

> **Where to find it:** the same panel, the **Links on this page** and **In your selection** buttons. Also in the page's context menu: **Open the Links in This Selection…** (with some text selected) and **Open Every Link on This Page…**.

> **What it does:** collects the links of the page behind the panel, or those contained in what you have selected. In a selection it also recognizes addresses written as plain text, because the most common list of URLs in the world is a chat message or a block of code with no links in it. The two buttons **replace** each other's results instead of adding up; whatever you typed by hand stays.

63. Asking all open tabs

> **Where to find it:** `Ctrl/Cmd+Shift+A`; the page's context menu → **Ask Every Open Tab…**, and with some text selected also **Ask Your Tabs About This…**; ⋮ → **Tools** → **Ask Your Tabs…**; menu bar → **View** → **Ask Your Tabs…**; can be pinned to the toolbar. On the phone: ⋮ sheet → **Tools** → **Ask your tabs**.

> **What it does:** a single question put to all the tabs of the window. The panel lists them all, each with a checkbox and with what including it would cost. Pages with nothing to read are listed in gray instead of being left out. The answer arrives in the AI sidebar as a normal turn, citing `[pagina 3]` according to the panel's numbering.

64. Size meter before sending

> **Where to find it:** above the send button of the **Ask your tabs** panel.

> **What it does:** before anything goes out, it estimates the size against what the chosen model can hold, and **refuses to start** when it would not fit: it suggests, cheapest first, removing the images, unticking the heaviest pages, sending less text per page, or switching to a model with more room. The numbers say they are estimates.

65. Bring this tab back later

> **Where to find it:** `Ctrl/Cmd+Shift+S`; right-click a tab → **Bring This Tab Back Later…**; menu bar → **File** → **Bring This Tab Back Later…**; the list of what is waiting is in ⋮ → **Tools** → **Tabs Coming Back…**. The toolbar button appears when something is waiting. On the phone: ⋮ sheet → **Tools** → **Tabs coming back**.

> **What it does:** the opposite of a favorite. The tab **closes now** and comes back when what it was waiting for happens: a moment (**In three hours**, **This evening**, **Tomorrow morning**, **This weekend**, **Next week**, or a time you choose), a rule (every Monday at nine, the first of every month), or **the next time you visit a given site**. It comes back as a background tab, with a dot on the button and a notification you can turn off. You cannot put off a locked, private or “stranger” tab, nor an internal Ciotto page: the refusal is explained.

66. Parked tabs with the window closed

> **Where to find it:** the **Tabs Coming Back** panel: **Keep waiting after the window is closed** and the choice of what to do when there is no window.

> **What it does:** keeps Ciotto in the system tray so that a tab can come back even with no windows open. It does not add itself to the system's startup programs: restarting the machine interrupts the wait. When the time comes and there are no windows, you can choose between keeping the tab ready for the next window, opening a window specially for it, or just being notified.

67. Favorites and reading list

> **Where to find it:** the star in the address bar (one click saves the page to the workspace's favorites; a second click opens **Folder**, **Manage favorites…** and **Remove**); `Ctrl/Cmd+D` (**Bookmark this page**); ⋮ → **Favorites and reading list**, with **Open saved items**, **Save as favorite** and **Add to reading list** (each with **Current workspace** or **Personal**); the bookmark button in the toolbar and `Ctrl/Cmd+Shift+O` open the **Saved items** page. On the phone: ⋮ sheet → **Page** → **Favorites and reading list**, and **Page** → **Save page…** to save the current page.

> **What it does:** a single ordered archive, which can be searched and organized into folders, for two kinds of item: **favorites** and the **reading list**. Each item has a scope: **personal** (it follows you everywhere) or **workspace** (it stays in that workspace), and the **Personal** / **Workspace** button on the row changes it. Folders nest with a `/` in the name (Work/Clients). Items are reordered by dragging them or with the arrows; reading list items are marked as read with **Read**.

68. Offline copy of a saved item

> **Where to find it:** the **Saved items** view → **Reading list**, the **Offline** command on the row; the copy opens in a reader with **Delete copy** and **Close**.

> **What it does:** keeps a readable copy of a reading list page, cleaned up as in reader mode (no scripts, forms or remote images), encrypted and bound to the profile, so the item stays readable even if the site disappears. To create it, the page must be open in a tab; it cannot be done from a “stranger” tab. Offline copies are **never** synced to other devices.

69. Reader mode

> **Where to find it:** ⋮ → **Page** → **Reader mode** (it becomes **Exit Reader mode** when active); can be assigned to a shortcut. On the phone: ⋮ sheet → **Page** → **Reader mode**, and it is one of the default quick actions.

> **What it does:** strips the article of the clutter around it and shows it in a readable form. It removes executable content and forms, keeps your reading position and does not capture anything into Memory on its own initiative.

70. Media controls and Picture in Picture

> **Where to find it:** the ♪ button in the toolbar, present only while a tab is playing sound or is muted; ⋮ → **Tools** → **Media controls**; right-click a tab → **Picture in Picture**. On the phone: long-press the tab → **Picture in Picture** and **Mute tab**.

> **What it does:** gathers in a single panel (**Media**) whatever is playing in all the tabs, with **Play** and **Pause**, ten-second skips forward and back, **Mute** and **PiP**, which pops the video out into a small always-on-top window when the platform supports it. On desktop, while Ciotto is in the foreground and something is playing, the keyboard's Play/Pause and Stop media keys respond

too. Ciotto does not read the media content: it only uses the controls the page exposes. On the phone the panel is not there: playback and skipping are left to the system controls.

71. Notes tied to the page

Where to find it: the **Notes for this page** button in the toolbar, always present; or ⋮ → **Notes** → **Notes for This Page** / **All Notes**; can be pinned to the toolbar and assigned to a shortcut, though none is set by default. On the phone: ⋮ sheet → **Page** → **Notes for this page** / **All notes**.

What it does: opens a **notes sheet next to the page**, on the right or below as you prefer (the **Dock notes to the side** and **Dock notes below** buttons are in the panel's header, and the choice is remembered). The sheet **narrows the page instead of covering it**: the page stays live and navigable while you take notes on it, unlike the other panels, which freeze pages into snapshots so they can sit on top of them. It saves itself as you write.

The sheet is tied to the **address**, not to the tab: it comes back when you return to that page, even in another window a week later. The address is cleaned up before it is stored — the `#ancora` goes, credentials go, tracking codes such as `utm_source` go — so the same page reached from a newsletter and from a bookmark shares a single sheet, while `?id=99` and `?id=100` remain the two different pages they really are.

At the bottom of the panel, under **Also noted on this site**, are **the other pages of the same site you have taken notes on**: from `pippo.com/prodotti` you can see that you also annotated `/servizi` and `/contatti`, and you get there with one click. **All Notes** opens a page that gathers them all, **grouped by site**, with a search that looks inside the text. **On the phone** the sheet always docks below: a narrow screen has no width to give a side panel without taking it away from the page.

Sync: notes have a category of their own, **Page notes**, among the sync categories (entry 127), **off by default** like history. The reason: a note is your own, but the list of pages you wrote notes on is a reading list that nobody agreed to publish. It is worth carrying between your own devices, but never without your having asked for it.

H. Macros and automation

72. Macro Studio

> **Where to find it:** ⋮ → **Macros** → **Manage Macros…**; Settings → **AI & automations** → **Open Macro Studio…**; can be assigned to a shortcut. It opens as an internal Ciotto tab. On the phone: ⋮ sheet → **Tools** → **Macros**.

> **What it does:** turns work done in the browser into a local flowchart, which can be inspected, branched and run again. On the left, the list of macros (**All** and **Scheduled here** filters), with the **AI** label for those that contain an AI step, **needs fixing** for those that have errors, and the outcome of the last run. In the center, the flowchart, with **Add a step** (a menu with search and categories: Page, Browser, Control, AI, Data, Finish), undo and redo, **?** for an explanation of each node on mouse-over, and the **Guide**. On the right, the properties of the selected node, or the macro's settings, folded into sections. Changes are saved automatically (**Saving…** / **Saved**). Whether a macro uses a model, and whether it is ready to start, is not something you choose: the steps it contains decide. Macros are saved in the application's local data folder; the format allows no arbitrary JavaScript and no unbounded loops.

Platforms: desktop, Android and iOS for building and editing; recording (entry 73) is desktop only, and on the phone only some kinds of step can be run (entry 74).

73. Recording a macro

> **Where to find it:** ⋮ → **Macros** → **New Recording**, then **Pause Recording** / **Resume Recording** / **Stop and Save Recording**; inside Macro Studio, the ● **Record** button, next to the choice between **Compact record** and **Faithful record**. During recording, a box in the toolbar shows the name, the level of detail (**Compact** / **Faithful**, clickable), **Pause** / **Resume** and **Stop**.

> **What it does:** records what you do. In **compact** mode it keeps the meaningful clicks and groups the changes made in fields; in **faithful** mode it also keeps key combinations and significant scrolling. The mode can be changed while recording. Only one recording or run can be active in the whole application. Sensitive fields (passwords, one-time codes, card numbers, security codes, IBANs) are not recorded in plain text: they become named inputs (entry 76), and not even the keys pressed inside them end up in the macro. Nothing is recorded from private windows or from “stranger” tabs. The addresses opened are saved as the browser resolved them, including the search on the chosen engine.

Platforms: desktop only.

74. Running a macro: preflight check and status bar

> **Where to find it:** the **Run** button inside Macro Studio, with the ▾ arrow (**Run options…**); the macro's name in ⋮ → **Macros**; macros pinned to the toolbar; a shortcut assigned to the macro

(entry 152). During the run, the status appears in the toolbar and in a strip above the page.

What it does: Run starts right away when nothing is missing; if a value is missing, or with **Run options...**, the **preflight check** opens, which states where the macro will run (**Current browser** or **Dedicated window**), with what level of safety (**Confirm each action**, **Confirm risky actions**, **Pre-authorized actions**), on which session and profile, with which credential set and which model, and which values have to be provided. The same applies to menus, pinned buttons and shortcuts: if the check is clean the macro starts, otherwise Macro Studio opens on the problem. During the run you can pause, resume or stop it; a step that needs confirmation shows a bar with **Allow once** and **Stop** and the type of risk. At the end the strip says how it went, which step failed and, if the macro wrote to a data sheet, how many cells and where, with **Open the sheet**. Macros without AI steps run without any network traffic to any AI provider.

Platforms: desktop. On Android and iOS the only macros that run are those made of steps the phone can perform (opening addresses, reading from the page, writing to sheets, opening, choosing and closing tabs, variables, conditions, AI extraction and decision on text only); for the others, the preflight check says which steps are missing instead of skipping them.

75. Reading from the page, and AI nodes

Where to find it: Macro Studio → **Add a step**: **Read from page** in the Page category; **AI extract**, **AI decision** and **AI mini-agent** in the AI category.

What it does: Read from page reads an element of the page without any model and keeps its text, the value of a field, the address of a link or an attribute in a variable: you choose the element by pointing to it on the open page, and the macro keeps a stable reference, not coordinates. If the reading might fail, **Use AI if reading fails** adds, on the error branch, an AI extraction that writes the same variable. Three kinds of node involve a model: **AI extract** (pulls out a value), **AI decision** (chooses among declared outcomes) and **AI mini-agent** (acts on the page inside a fence). Each has its own context (text, screenshot or both), its own provider and model, and, for the mini-agent, the list of allowed tools, the list of allowed domains, a cap on turns and actions, and a time limit. A saved secret can only go into a sign-in field: if the mini-agent tries to write it anywhere else, the macro stops.

76. Sensitive values and credential sets for macros

Where to find it: the properties of the fill node in Macro Studio; the **Required data** section in the macro's settings; the **Credential set** choice and **Save encrypted set** in the preflight check.

What it does: a recorded sensitive field does not keep the value that was typed: it becomes a **named input**, asked for at run time or set beforehand in **Required data** (a secret stays encrypted on this device). Optional credential sets are encrypted through the operating system's keyring; where that encryption is not available, only values valid for a single run are accepted. On Linux, a keyring without real protection (`basic_text`) does not count as encryption. The confirmations during the run say which field and which variable are about to be used.

77. Macros that run on their own

> **Where to find it:** Macro Studio, the macro's settings → **Run on a schedule**; the **Scheduled here** filter in the list shows the ones armed on this device.

> **What it does:** runs a saved macro without anyone starting it: **Every so often** (at intervals of 5 minutes or longer) or **At a time of day**, on chosen days. The schedule travels with the macro to your other devices, but running does not: each device it should run on has to be armed separately with **Save and activate**, so that two paired machines do not do the same job twice; **Pause schedule** stops it. Before arming it, Ciotto checks that the macro can really run with nobody at the screen: no steps that ask for confirmation, no new window, no panel assigned to an identity, every risky step already authorized; if something is wrong, the list of reasons appears next to the button. On desktop the run takes place in a window you cannot see; with **Keep running with no window open**, Ciotto stays in the system tray after the last window is closed. **Tell me** chooses whether to notify you **Never**, **When one fails** or after every run (**Every run**). A macro set to use your own profile runs with your sign-ins only if you tick **Allow it to run as the real you**; otherwise it starts on a throwaway session. **Run it now** tries it straight away. Runs missed while the computer was off become a single run.

Platforms: desktop. On Android and iOS a scheduled macro cannot run in the background: it starts when Ciotto is open, at launch, on returning to the foreground or at the periodic check, and the run remains visible.

78. Data sheets

> **Where to find it:** ⋮ → **Macros** → **Data sheets**; the **Write to sheet** step in Macro Studio (Data category); ⋮ → **Macros** → **Open macro results folder**. On the phone: ⋮ sheet → **Tools** → **Data sheets**.

> **What it does:** a place where a macro writes what it has read, cell by cell. The **Write to sheet** step chooses the sheet (or creates one with **New sheet…**) and the cell: **One fixed cell**, **Next free row** of a column, or **One row per day**, which puts the date next to the value and corrects today's row instead of adding another one. Secrets are never written. The **Data sheets** page shows the sheets; each cell remembers which macro wrote it and when, and it can be corrected by hand without touching the macro. A sheet can be renamed, emptied (**Empty it**), deleted and exported as CSV (**Export as CSV**), with any text that looks like a formula treated as text. On desktop, **Also keep it as a file…** also keeps a copy of the sheet in a file, updated at every run and every correction, and **Open macro results folder** opens the results folder. The sync category for sheets is off by default.

Platforms: desktop, Android and iOS. On the phone there is no file copy and no results folder: CSV export delivers the file as a download, and **Data sheets** takes the place of the folder. Data sheets appear among the sync categories only on Android and iOS (entry 127).

I. AI assistant

79. AI sidebar

> **Where to find it:** `Ctrl/Cmd+J`; menu bar → **View** → **Toggle AI Sidebar**; ⋮ → **Tools** → **AI chat**. On the phone: ⋮ sheet → **Tools** → **AI chat**.

> **What it does:** a conversation beside the page (**AI assistant**). A switch at the bottom decides whether the page you are looking at goes with the question (**Page context on**) or not (**Page context off — the page you're on is not sent**). The answer arrives a piece at a time as it is generated, can be interrupted with **Stop**, and **Clear** starts the conversation over. Enter sends, Shift+Enter starts a new line. The keys stay in the main process: the interface never sees them.

80. AI providers and models

> **Where to find it:** the sparkle button in the toolbar → the **AI provider** panel; ⋮ → **Tools** → **AI provider**; the provider in use can also be chosen in Settings → **AI & automations** → **AI provider**. On the phone: ⋮ sheet → **Tools** → **AI provider**.

> **What it does:** the panel has a **Provider** menu with all the providers, configured or not: Anthropic (Claude), OpenAI, Google Gemini, xAI (Grok), DeepSeek, Mistral, Groq, OpenRouter and Ollama (local, on your own machine). For the chosen one it shows the status, the key, the **Model** menu and **Use this one**; at the bottom it says which model the assistant is using and offers **Open the assistant**.

81. API keys: entering, checking and removing them

> **Where to find it:** the **AI provider** panel, after choosing the provider in the **Provider** menu.

> **What it does:** the key is pasted into the **API key** field, and with **Connect** it is checked against the provider **before** it is stored, so a wrong key is found out right away and not at the first question. For OpenRouter there is a guided sign-in (**Sign in with OpenRouter**, desktop only); for Ollama, **Look for it on this machine** looks for the local service and **How do I install it?** explains how to get it. **Where do I get a key?** leads to the provider's console, and **Remove** removes the key. Keys stay in the main process; if the machine offers no keyring, the panel says that the key is stored in plain text.

82. Page translation

> **Where to find it:** ⋮ → **Tools** → **Translate page** and **Show original page**; the banner offering to translate appears only after you switch on ⋮ → **Tools** → **Offer to translate pages**, which is off by default. On the phone: ⋮ sheet → **Tools**, with the same three items.

> **What it does:** recognizes the page's language and translates it in place: the text is replaced where it stands, starting with what you are looking at. The translation goes through the AI provider already configured — preferably **Ollama on your own machine**, which costs nothing and lets nothing leave the device. With an online provider, Ciotto asks for consent first, because in that case the page's text really does leave, and it asks for every translation. Fifteen languages are recognized: all the interface languages except Klingon. With the offer switched on, a page in another language brings up the banner **This page is in … — translate it?**, with **Translate**, **Never for this site** and **Dismiss**. With no provider at all, the language is still recognized and the banner offers **Show me how**, a guide to installing Ollama, instead of staying silent; this offer appears once per session. The sites you mark stay on this device and can be reviewed in Settings → **Appearance & accessibility** → **Per-site translation**.

Platforms: desktop, Android and iOS.

J. Voice

83. Voice commands

> **Where to find it:** `Ctrl/Cmd+Shift+Space` (one press to listen, another to stop); the microphone button in the toolbar; ⋮ → **Tools** → **Speak a command**; menu bar → **View** → **Speak a command**. Settings → **AI & automations** → **Voice**. On the phone: ⋮ sheet → **Tools** → **Speak a command**.

> **What it does:** turns a sentence spoken aloud into an instruction for the browser: opening a site or searching for something, switching to a tab by saying its title, any action in the list of shortcuts (“close the tab”, “reload”), a question for the assistant (the full answer goes to the sidebar), a piece of text to type into a field on the page. The recording goes to the configured AI provider that can transcribe — OpenAI, Groq or Google — with your own key; the first time, Ciotto asks for consent for that provider (**Send your voice to be transcribed?**) and keeps no recording. The words are matched against the labels of all of Ciotto's languages at once, so a command works even in a language other than the interface's. Closing the window, closing the other tabs and burning everything always ask for confirmation, by voice or with **Yes** and **No**; after ten seconds without an answer, it counts as a no. A command reconstructed by a model, rather than recognized word for word, must also be confirmed, unless all it does is open a panel. Voice is on by default (**Speak to Ciotto**); turning it off removes the button and prevents the microphone from being opened. In Settings you also choose the transcriber (**Transcribed by**) and change the key combination; listening for a wake word is shown but not yet available.

Platforms: desktop, Android and iOS. On the phone there is no toolbar button: you use the ⋮ sheet.

84. Spoken answers

> **Where to find it:** Settings → **AI & automations** → **Voice**: **Answer out loud**, **Speaking speed** and **Voice**.

> **What it does:** after a voice command, Ciotto says what it heard and what it did (“Opening wikipedia.org”, “Done — …”), so you do not need to look at the screen; when it has not understood, it says so. It uses the voices already installed on the device: it downloads nothing and sends the text nowhere to be spoken. Which languages it can speak depends on the voices the system has. It is on by default, together with voice. While it is speaking it does not listen, so it cannot answer its own question.

Platforms: desktop, Android and iOS. On Android and iOS it speaks with the system's speech synthesis engine.

85. Spoken tasks on the page

> **Where to find it:** the same voice command, saying what to do on the page instead of the name of a command (“fill in this form with my address”).

> **What it does:** hands the spoken task to the page agent. Before starting, Ciotto asks for confirmation. The run is a macro that is never saved, tied to the site of the tab in the foreground so that it cannot move on to another one, with tighter limits than a macro written by hand (few turns, few actions, two minutes), and it asks for permission before every click and every keystroke. Saying no stops it there.

Platforms: desktop. On Android and iOS the task goes to the assistant in the sidebar instead.

K. Privacy and security

86. The shield and the “What was blocked here” panel

> **Where to find it:** the shield icon in the toolbar, with the count of what has been blocked, opens Settings → **Privacy & security** (entry 87). The quick panel opens from ⋮ → **Privacy and identity** → **What was blocked here**, and can be pinned to the toolbar under the name **Privacy and security**. On the phone: ⋮ sheet → **Privacy and identity** → **What was blocked here**, which is also one of the default quick actions.

> **What it does:** the number on the shield adds up the ads, trackers and dangerous pages stopped since launch; the counters live only in memory and start again from zero at every launch. The quick panel gathers the switches for the four core protections, the **Show ads on …** switch for the site in the foreground (entry 89), the encrypted DNS choice (entry 93), the counters (**Ads**, **Trackers**, **Malware**, **Banners**) and the list of exceptions, each with **Remove**. The **All privacy features** → link leads to the full catalog. Shield, panel and catalog read the same state, so they cannot contradict each other.

87. Full catalog of protections

> **Where to find it:** ⋮ → **Privacy and identity** → **Privacy and Security…**; Settings → **Privacy & security**; the shield icon in the toolbar. On the phone: ⋮ sheet → **Privacy and identity** → **All privacy features**.

> **What it does:** at the top, the session counters (**Ads blocked**, **Trackers blocked**, **Malware/phishing stopped**, **Cookie banners rejected**), the line on the status of the protection lists, and Clear browsing data (entry 58). Below, a card for each protection, divided into groups (Network, Identity, Data, Downloads), with a title, a one-line summary and, on request (**What situation does this protect me from?**), a paragraph describing the **real situation** it answers, not the mechanism. Protections not yet built have no card promising them, and those the platform cannot make work are not shown. All the protections beyond the four core ones (**Core**) are **off** until you switch them on.

88. Signed protection lists

> **Where to find it:** there is nothing to set; the status can be read in Settings → **Privacy & security**, in the line below the counters.

> **What it does:** the lists that decide what to block (EasyList and EasyPrivacy for ads and trackers, URLhaus and OpenPhish for malware and phishing) come from Ciotto's site in a signed bundle: a list of the files with their hashes, signed with an Ed25519 key written into the program. Before using a single rule, Ciotto checks the signature and every hash, and refuses a bundle that is not newer than the one it already has, so an old copy cannot be put back into circulation. If the check

fails, nothing changes: the last verified copy stays in use and the line says why. A bundle more than a week old is still used, but the line points this out. Distributed versions never fall back on unsigned lists.

Platforms: desktop, Android and iOS.

89. Ad blocking and anti-tracking

Where to find it: the privacy catalog, first card; in the **What was blocked here** panel (entry 86), the **Show ads on …** switch for the site in the foreground, and the list of exceptions.

What it does: filters advertising and tracking requests using the EasyList and EasyPrivacy lists, which in distributed versions come from the signed bundle. It is **on** by default. **Show ads on …** turns blocking off for that one site, permanently (including its third-party ads), without touching other sites; the exception is removed with the same switch or with **Remove** in the list.

90. HTTPS-only

Where to find it: the privacy catalog; an exception for a single site from the shield.

What it does: turns an `http://` navigation into `https://` before it leaves the machine. A site that really has no `https://` shows a block page that explains itself (**Not secure**), not a silent connection error, with three choices: **Go back**, **Continue anyway** (that site stays on `http://` for the rest of the session) and **Don't block this site again** (a permanent exception for that site alone). Exceptions are reviewed and removed in the **What was blocked here** panel. The list of dangerous sites (entry 91) looks at the address before the upgrade, so a phishing address on `http://` is stopped even if the site answers on `https://`.

91. Local malware and phishing list

Where to find it: the privacy catalog.

What it does: the list is built from URLhaus and OpenPhish, arrives in the signed bundle of protection lists and sits **on the machine**: it is not Google Safe Browsing, so nothing of the address you are about to open is sent anywhere to be checked. The address is compared before any rewriting (upgrade to `https://`, link cleaning), also with the other scheme and without the trailing dot. A match blocks the navigation with the **Dangerous site** page; **Continue anyway** is a one-time permission for that exact address and never becomes a permanent exception. If the list cannot be downloaded, Ciotto tries again after 30 seconds, then 2, 10 and 30 minutes, instead of staying unprotected until the next round, and the error appears in the list status.

92. Download warnings

Where to find it: the privacy catalog; the warning appears below the download's row, in the download shelf.

> **What it does:** for every download it looks at three things, in order of severity: whether the source, or one of the intermediate hops, is on the local malware list (**Matches a known malware/phishing list**); whether the file is of a type that can run code — installers, scripts, shortcuts and the like, from `.exe`, `.msi` and `.ps1` to `.url`, `.chm` and `.vhdx` (**This file type can run code — only open it if you trust the source**); and whether it arrived over an unencrypted connection (**Downloaded over an insecure connection**). The most serious warning is the one shown. The download is not blocked: the file reaches the disk, but **Open** on a file flagged as dangerous or listed asks for confirmation before opening it. Turning the warnings off turns all three off together.

93. Encrypted DNS

> **Where to find it:** ⋮ → **Privacy and identity** → **What was blocked here** → **Encrypted DNS**: a choice between **Off (system)**, **Cloudflare** and **Quad9**.

> **What it does:** routes name resolution through DNS-over-HTTPS with the chosen service, instead of the system's unencrypted resolver. It is off by default, and the choice takes effect at once, without restarting.

Platforms: desktop only.

94. Site permissions

> **Where to find it:** the bar that appears above the page when a site asks for the camera, microphone, location, notifications, clipboard, MIDI devices or the local network, with **Remember for this site** (ticked by default), **Block** and **Allow**; the answers already given can be reviewed from the site information popover (entry 35).

> **What it does:** nothing is granted without asking, and the answer is remembered for that site. Permissions belong to the workspace: the same question in another workspace starts from scratch. Notifications show as denied until the site asks for them and you answer **Allow**: a page cannot show one without having gone through the question. For camera and location, on the other hand, a site that merely checks the state of the permission sees it as granted, so it does not give up on asking; actual access still goes through the bar. Several identical requests from the same page become a single question.

Platforms: desktop, Android and iOS. On the phone only camera, microphone and location are asked for; the other permissions are refused. On iOS the location request is made by the system, site by site, without going through Ciotto.

95. Open as a stranger

> **Where to find it:** the privacy catalog, to switch it on (**Open as a stranger**); then right-click a link → **Open Link as a Stranger**.

What it does: opens a link on a throwaway session held in memory: no cookies and no history follow it in, and when the tab is closed its storage is wiped immediately, not when the application quits. It is different from the private context, which is instead shared among several tabs. In the "stranger" tab saved passwords are not offered, and permissions granted there stay only in memory, without touching those of the workspace; if the profile goes out through a proxy (entry 103), the tab uses the same proxy.

96. No requests before the filters are ready

Where to find it: the privacy catalog.

What it does: closes the gap of one or two seconds after launch during which the filter lists are not yet loaded. With this protection on, the very first requests wait — five seconds at most — for ad blocking and the malware list to finish loading. If they are not ready after five seconds, the request is canceled instead of going out unprotected: it is tried again when the lists are ready.

Platforms: desktop only.

97. Sites cannot knock on your home network

Where to find it: the privacy catalog; the request appears as a permission.

What it does: asks for consent before a public page reaches a private address (`192.168.*`, `10.*`, loopback, link-local, `100.64.0.0/10`, `0.0.0.0`), whether through a background request or by navigating there directly. It also applies to names that lead into the home network: Ciotto resolves the name before deciding, and recognizes local suffixes (`.local`, `.lan`, `.home.arpa`…) and typical router names. The request says that the site wants to reach a device on the local network; the permission is called *local-network* and is remembered per origin, exactly like camera or location. Many identical requests from a page become a single question.

Platforms: desktop only.

98. Clean links

Where to find it: the privacy catalog.

What it does: removes known tracking parameters (`utm_*`, `fbclid`, `gclid` and the others) from a navigation before it loads. It acts before HTTPS-only, so the cleaned address goes through the whole chain from the start.

99. Permissions that expire

Where to find it: the privacy catalog; the new options appear in the permission request, next to **Remember for this site**.

> **What it does:** adds **Just for today** (24 hours from when you answer) and **For a week** (seven days) next to the usual **Always**. An expired permission is treated as if it had never been asked for.

100. Cookie banners: rejected, not hidden

> **Where to find it:** the privacy catalog.

> **What it does:** recognizes the page's consent manager and presses the real reject button, instead of hiding the banner with a visual trick while the site silently records a consent. Where the banner offers no reject button, Ciotto removes it from view without clicking anything, so the question stays unanswered. Rejected banners are counted in **Cookie banners rejected**.

Platforms: desktop only.

101. Forget me when I close this site

> **Where to find it:** the privacy catalog, to switch it on; sites are added by typing their name in the **Sites to forget on close** list that appears below the protection's card, with **Add** and **Remove**.

> **What it does:** deletes the cookies and storage of the marked site, subdomains included, the moment its last tab in that window closes, and also when the window is closed or Ciotto quits (on quitting it waits up to three seconds for the cleanup to finish). A stated limit: a copy of the same site left open elsewhere is a known and accepted gap, rather than a deletion that would tear the session away from another window.

102. Readable privacy report

> **Where to find it:** the privacy catalog, to switch it on; the report (**Recent activity**) appears below the protection's card.

> **What it does:** turns the counters into sentences. A circular log — never written to if the feature is off, and valid only for the current session — notes the site of each block and the tracker's host, resolved against a small domain→company table built into the program: “On nytimes.com Ciotto stopped 9 requests to Google, Criteo and Taboola” instead of a bare number.

Platforms: desktop only.

103. A different exit for each profile (proxy)

> **Where to find it:** the privacy catalog, to switch it on; the proxy is set on the profile's row in the profiles panel (⋮ → **Profiles** → **Manage Profiles…**), in a field that accepts `socks5://host:porta` or `http://host:porta`, with a **Test** button.

> **What it does:** applies a different proxy to each profile's session, so that two identities do not appear to come from the same network. **Ciotto does not provide the proxy**, and the card says so openly. The Test button queries a real endpoint through that profile's session to confirm that

it really works (**Works** or **Failed**). The proxy also applies to the pages' WebRTC calls, to “stranger” tabs opened from that profile and to the site icons in the strip, so the real address does not slip out through a side door. When the proxy is active, the profile's chip in the toolbar has a dot.

Platforms: desktop only.

104. Extension updates that ask for more are stopped

Where to find it: the privacy catalog, to switch it on; the approval bar appears at the top of the window when an update is held back.

What it does: compares the manifest of every extension load with the last one accepted. A reload of the same version goes through without fuss; a new version that asks for a permission or a host it could not reach before is **unloaded from the session** right after loading, and held back until the bar that states exactly what has changed (“… updated from 1.2 to 1.3 and now also asks to:”) is approved with **Allow this update**. The block survives a restart.

Platforms: desktop only.

105. Embedded content loads when you say so

Where to find it: the privacy catalog; per-site exceptions are written in the **Sites where embeds always load** list that appears below the protection's card.

What it does: holds a third-party embed (YouTube, X, Instagram, Google Maps) behind a placeholder (**Load this … embed?**) until you click it. The request is diverted to an internal page and never reaches the real host; clicking the placeholder makes the same frame navigate to the real address.

Platforms: desktop only.

106. Photos without coordinates (experimental)

Where to find it: the privacy catalog, to switch it on; then it has to be enabled **explicitly for each site**, by writing the site in the **Sites to strip photo data on** list below the protection's card.

What it does: re-encodes an eligible photo (JPEG, WebP) before it leaves the page — the same transformation you would get by opening it and saving it again — and with it the EXIF data and any GPS coordinates disappear. If decoding fails, it does not block the upload: it sends the original.

Platforms: desktop only.

107. Macros run as strangers

Where to find it: the privacy catalog, to switch it on; the choice for an individual macro is in Macro Studio → the macro's settings → **Safety & session** → **Session** (**Sandbox — runs as a**

stranger or **Your profile — runs as you**), and the preflight check shows it for every run.

What it does: with the protection on, a macro run starts on a throwaway session instead of taking over the tab that is already open. All the tabs opened by the run share the same partition, so a sign-in made in one of them is visible to the others in the same run and to nothing outside it. The partition is wiped when the last tab sharing it closes. A macro that really needs your own identity is set to **Your profile — runs as you**, and the preflight check flags it. With the protection off, every run acts with the real profile, and the preflight check says so. Stated limits: no extensions, and downloads are attributed to the default profile.

Platforms: desktop only. On the phone the switch is not there: there, macros always act with the tab's profile.

108. Believable time zone and anti-fingerprinting noise, per profile (experimental)

Where to find it: the privacy catalog, to switch it on; below the card appear the four channels **Canvas noise**, **WebGL noise**, **Audio noise** and **Font measurement noise**, each with **On** and **Off**. The time zone is chosen on the profile's row in the profiles panel (the **Machine's own timezone — e.g. Europe/Berlin** field).

What it does: changes the time zone that a profile's tabs declare, keeping it consistent with a language and a platform that are real for this machine; leaving the field empty keeps the machine's own time zone, and a name that is not a time zone is rejected (**Not a timezone**). In addition, it adds a little noise to what canvas, WebGL, audio and font measurement report, the surfaces a site uses to build a fingerprint that survives the deletion of cookies: the noise stays the same for a given site during a session, so pages do not break, but it changes from one session to the next and from one profile to another. The four channels switch on together with the protection, and each one can be switched off separately. **It is not complete anti-fingerprinting**: screen size, installed fonts and dozens of other signals remain the real ones. If it cannot apply itself, it leaves the tab exactly as it would have been: never a broken page.

Platforms: desktop only.

109. Encrypted cookies per profile (experimental)

Where to find it: the privacy catalog, to switch it on; it is activated on the profile's row in the profiles panel, with **Enable encrypted vault** and a passphrase typed twice. At launch, the **Unlock your vaults** window appears.

What it does: moves a profile's cookies out of Chromium's plain-text store and keeps them as a single file encrypted with the chosen passphrase. At launch every protected profile starts **locked**: the unlock window asks for each one's passphrase (**Unlock**, then **Continue to Ciotto**), and skipping one simply means starting signed out, not exposed; the profile's row says **Vault locked — logins not restored.** and offers **Unlock** at any time. It protects cookies and sign-ins, not the sites' local storage. **A lost passphrase takes with it the cookies it protected**: that is how it is built. **Disable vault** asks you to type **disable**; if there is a locked master password, it first asks

> you to unlock it, because turning off the vault also moves the profile's passwords, cards and addresses.

Platforms: desktop only.

110. Burn everything

> **Where to find it:** ⋮ → **Privacy and identity** → **Burn everything**; can be pinned to the toolbar (entry 154); can be assigned to a shortcut (entry 152), but has no default combination. On the phone: ⋮ sheet → **Privacy and identity** → **Burn everything**.

> **What it does:** closes all the tabs of the workspace you are in — **except the locked ones** — and goes back to an empty start page. On desktop it asks first (**Close every tab in this environment?**), and the question says two things: that the closed tabs can still be recovered with `Ctrl/Cmd+Shift+T`, and how many locked tabs will stay open. In the same question it offers to **clear the data too** (**Close and clear this profile's data**): the profile's cookies, storage and cache, plus the workspace's history and downloads. **The Don't ask again box applies only to the recoverable half:** clearing the data always asks. The switch that makes it ask again is in Settings → **Tabs & performance** → **Ask before burning everything**. The command is never recorded into a macro, and when spoken it always asks for confirmation.

Platforms: desktop, Android and iOS. On the phone, tapping the item closes the tabs at once, without the question; data is cleared from Clear browsing data (entry 58).

L. Passwords and credentials

111. Saved passwords

> **Where to find it:** Settings → **Passwords & autofill**; ⋮ → **Privacy and identity** → **Passwords and autofill**, which lists the logins saved for the open site and leads to the full list with **Manage passwords…**. On the phone: ⋮ sheet → **Privacy and identity** → **Passwords and autofill**.

> **What it does:** keeps credentials **per profile**, encrypted with a key protected by the operating system's keyring. They are added with **Save a login** (HTTPS or localhost addresses only), searched by site, username or label, changed with **Edit** and removed with **Remove**. The list shows only the metadata (site, username, label), never the password. On Linux, if the available keyring is not secure, saving is refused and you are asked to turn on the protection of the profile's vault.

112. Autofill

> **Where to find it:** the key in the address bar (**Fill a saved password**), which appears when a credential exists for that site; ⋮ → **Privacy and identity** → **Passwords and autofill**, which lists that site's credentials. On the phone: ⋮ sheet → **Privacy and identity** → **Passwords and autofill**.

> **What it does:** fills in once, after checking that the origin of the main frame exactly matches the saved one. **It does not submit the form**: confirming is left to you. In a “stranger” tab, credentials are not offered.

113. Addresses and payment cards

> **Where to find it:** Settings → **Passwords & autofill**, the **Save address** and **Save card** boxes; filling is requested from the person icon in the address bar (**Addresses**), which appears when the profile has at least one saved address or card.

> **What it does:** keeps addresses and contact details, and payment cards, encrypted with the same key as the passwords and closed by the same lock: the master password, if set, locks both. An address is filled in only after you choose it, never on its own. The list of cards shows only the last four digits, never the whole number. Showing the whole number (**Show number**) or filling in a card requires entering the master password **every time**; without a master password, Ciotto asks you to set one instead of going ahead. The security code is never asked for or stored.

Platforms: desktop, Android and iOS.

114. Offer to save after signing in

> **Where to find it:** a bar that appears at the top of the window right after you sign in: **Save your**

password for … as …?, with **Save** and **Not now**.

What it does: when a sign-in form is submitted, Ciotto offers to save the credential. The form is submitted by the page, not by Ciotto. The password stays in the main process, indexed by an opaque token. The offer does **not** appear in the private context or in a "stranger" tab, and it does not come up again for a credential already saved. On the phone the offer is accepted only from the main page, and the origin is taken from the tab, not from what the page declares: a third-party frame cannot offer to save a credential for another site.

115. Password generator

Where to find it: Settings → **Passwords & autofill** → **Save a login**, the **Generate** button next to the password field.

What it does: produces a random twenty-character password, ready to paste into the site and to save in the same operation.

116. Master password and auto-lock

Where to find it: Settings → **Passwords & autofill**, the **Master password** box: **Set master password** (at least eight characters, typed twice), then **Lock now**, **Change**, **Remove** and **Auto-lock**.

What it does: seals the credentials key, already protected by the system keyring, a second time; the same lock closes addresses and cards. It adds an automatic lock after a period of inactivity (**1 minute**, **5 minutes**, **15 minutes**, **1 hour** or **Never**). It is also the re-authentication Ciotto asks for before showing or copying a password and before using a card (entries 117 and 113): without a master password those operations are not available. It does not touch cookie encryption, which is a separate feature. **There is no recovery path.**

117. Showing and copying a password

Where to find it: the credential's row in Settings → **Passwords & autofill**, the **Reveal** and **Copy** commands.

What it does: a password is shown or copied only after the master password has been entered, every time. Without a master password the operation does not start, and Ciotto explains that you need to set one. Entering it also unlocks the vault, if it was locked. A password that is shown disappears after twenty seconds; copying is done by the main process, which after twenty seconds empties the clipboard if it still contains that password, even if the Settings window has been closed. There is no channel that returns all the passwords in plain text at once.

118. Password health report

> **Where to find it:** Settings → **Passwords & autofill**, above the list of credentials (**Password health: … weak · … reused**) and on the individual rows (**Weak**, **Reused**).

> **What it does:** a local check, with no network, that flags passwords reused across several entries of the same profile, and weak ones. The comparison happens in memory; the report contains only identifiers and flags, never the passwords or their hashes.

119. Six-digit TOTP codes

> **Where to find it:** the **TOTP secret or otpauth:// (optional)** field in the credential form; the code appears on the credential's row, with the seconds remaining.

> **What it does:** optionally keeps a TOTP secret (in plain base32 or as an `otpauth://` address) inside the credential's encrypted payload, and generates the six-digit codes on the machine. The secret is also imported from the columns of password managers' CSV files.

120. Checking against known breaches

> **Where to find it:** Settings → **Passwords & autofill**, the **Check for breaches** button above the list.

> **What it does:** an **optional, on-demand** check against Have I Been Pwned, using k-anonymity: only a five-character prefix of the SHA-1 hash leaves the device, never the password. The entries found get the **Breached** label, and a line sums up how many there are. The result contains only identifiers and counts.

121. System passkeys

> **Where to find it:** they appear on their own when a site offers sign-in with a passkey.

> **What it does:** passkey ceremonies stay within Chromium, Android Credential Manager or Apple's authentication services. **The private keys of passkeys never enter Ciotto's storage or its sync.**

M. Lightning identity

122. Creating the Lightning identity

> **Where to find it:** the lightning-bolt button in the toolbar; ⋮ → **Privacy and identity** → **Lightning Identity**; Settings → **Sync, Lightning & devices** → **Open Lightning…**. On the phone: ⋮ sheet → **Privacy and identity** → **Lightning identity**.

> **What it does:** the identity is a key that Ciotto creates on this machine and sends nowhere. Sites that support it can recognize you without an email address and without a password, and for signing in and for WebLN **each site sees a different key**, so sites cannot compare notes with one another; the Nostr name (npub), on the other hand, is a single one, because in Nostr it is your public identity. Here Lightning serves only to have you recognized and to link your own devices: there is no account, no funds, nothing to top up, and no payment. The panel opens with **Create my identity** and **I already have a recovery phrase**; once the identity has been created, it shows the key and the Nostr name.

123. Backing up and restoring the twelve words

> **Where to find it:** the **Lightning identity** panel: **Write down my recovery words** (then **Show my recovery words again**), **I have written them down**, **I already have a recovery phrase** with **Restore**, **Remove this identity**.

> **What it does:** shows the twelve words, which are the only way back to that identity, with a prompt to write them down on paper, in order. Until you confirm with **I have written them down**, the panel warns that the identity has never been written down, and the toolbar button keeps a dot. The words can be seen again at any time with **Show my recovery words again**, without further questions. Restoring rebuilds the identity from an existing phrase, counting the words as you type them. Deleting is irreversible: without the twelve words, every site that recognized you will no longer recognize you, and to confirm it you have to type **remove**.

124. Signing in to a site with the identity

> **Where to find it:** a confirmation bar that appears at the top of the window when a site offers Lightning sign-in (LNURL-auth): **Sign in to … with your Lightning identity?**, with **Cancel** and **Sign in**.

> **What it does:** the site presents a challenge; Ciotto signs it with the key dedicated to **that** site, and only after explicit consent. Nothing is signed until the bar gets an answer, and the request expires on its own after two minutes. If the sign-in is started by a page other than the site you are about to sign in to, the bar says so (**Careful: … opened this login for …**). If the identity does not exist yet, accepting creates it. Sign-in is not offered in private windows.

125. WebLN and Nostr for pages

Where to find it: available to pages that use them; every call goes through a permission, asked for with the same bar as site permissions.

What it does: exposes two well-known interfaces to pages — `window.webln` and `window.nostr` — through a single controlled channel. WebLN offers only the site's key and the signing of a message, not payments. **Seeing who you are and signing as you are separate permissions**, because they carry different risks: a site authorized to see your identity is not, by that alone, authorized to sign with it. Signing is asked for **on every request**, is never remembered, and the bar shows what is being signed (the message, or the type and content of the Nostr event); an event dated in the future is refused. If the identity does not exist yet, the first consent creates it. In private windows and “stranger” tabs these interfaces are not offered.

N. Encrypted sync and tab transfer

> **Note for testers:** the tests in this section require two up-to-date Ciotto devices. Tests involving the Web app or the extensions also require the published companion service; until it is published, they must be marked as not executable. No test requires funds: Ciotto has no wallet, no channels and no payments.

126. Sync status

> **Where to find it:** the sync button in the toolbar → the **Encrypted sync** panel; ⋮ → **Privacy and identity** → **Encrypted sync**; Settings → **Sync, Lightning & devices**. On the phone: ⋮ sheet → **Privacy and identity** → **Encrypted sync**.

> **What it does:** the panel gathers, in sections that open on request, pairing (entry 129), **Send this tab**, **What follows you**, **Inbox**, **Tabs from other devices**, **Devices and contacts** and **Connection details**, which shows the transport in use for tabs and for bulky data, and the address at which the device can be reached. In Settings the same section has the categories, the list of devices with **Revoke**, and the **Sync now** command. No Ciotto server keeps tabs or browser data.

127. Data categories that can be synced

> **Where to find it:** the **Encrypted sync** panel, the **What follows you** box; the same switches in Settings → **Sync, Lightning & devices**.

> **What it does:** each category is switched on or off separately. On by default: **Remote tabs**, **Workspaces**, **Profile descriptors** (non-secret descriptors only), **Projects**, **Favorites and reading list**, **Portable preferences**, **Sanitised macros**. **Off** by default: **Page notes**, **History**, **Memory**, **Passwords (authorized devices only)**, and on Android and iOS **Data sheets**. The categories that are off are switched on with an ordinary checkbox; passwords also need each individual device to be authorized (entry 137). A device shows only the categories it can really handle: on desktop there is no data sheets category.

128. What is never synced

> **Where to find it:** a rule of sync, the same on every device; the Web app restates it in the **Portable data** box.

> **What it does:** sync **never includes**: recovery phrases, private keys, cookies, site storage, passkeys, AI provider tokens, proxy credentials, macro credential values, the contents of cookie vaults, offline copies of saved items, downloaded files, the download list and private sessions. Passwords travel only under their own category, with a separate authorization for each device.

> The one intended exception, and only on request: the session handoff with a QR (the entry after pairing) carries the cookies and storage of the shared tab's site, once, to a phone of your own.

129. Pairing a device

> **Where to find it:** the **Encrypted sync** panel → **Pair a device or contact** → **Invite / QR**, choose **My device** and **Create 10-minute invite**; after the first pairing, **Devices and contacts** → **Pair a new device**. On the other device: **Receive / Scan** on the phone, or **Receive**, paste the `ciotto://pair` link and **Verify**.

> **What it does:** generates a **single-use, signed** invite, valid for ten minutes, as a QR code and as text to copy with **Copy invite**. From the desktop to a phone, a single scan is enough (**Scan this QR once on the new phone, then approve the request here.**): the phone waits (**Waiting for approval on the desktop…**) and the desktop asks **Pairing request** — **Allow … as a device?**, with **Allow** and **Reject**. In the other cases pairing is mutual: whoever receives the invite approves it and in turn shows a return invite, to be read on the first device. The device being added keeps its own keys but does not receive the recovery phrase. Only the device that holds the Lightning identity can add devices; an added device can only invite contacts. Devices and contacts cannot be renamed.

Platforms: reading the QR with the camera is available only on Android and iOS; on the desktop you paste the link.

130. Handing the session over to your phone with a QR

> **Where to find it:** on the desktop, the **Encrypted sync** panel → **Invite / QR** (with a device already paired, the section is titled **Receive or share a Ciotto session**); on the phone, the same panel → **Receive / Scan**.

> **What it does:** brings the page open on the computer, signed in, to your already paired phone, together with the other open tabs. **Independent Ciotto session** shows the active tab, the **Duration** (5, 15, 30 or 60 minutes; 15 by default) and a QR valid for two minutes, which can be made again with **Regenerate QR**. A single scan transfers, encrypted, the cookies and storage of that page's site: the phone loads them into a throwaway profile and opens the page as a normal Ciotto tab; the other tabs arrive without their sign-ins. On the desktop, the red **Shared session** control appears in the toolbar, with the time remaining and **stop**: when you stop it or the time runs out, the phone closes that tab and deletes the profile. **Close everything on the phone** asks the paired phones to close the other tabs that arrived this way as well. No images of the page and no remote commands are passed.

Platforms: you share from the desktop (Windows, macOS, Linux) and receive on Android and iOS; on iOS the handoff must be completed with the app in the foreground. Not available in the Web app or in the extensions.

131. Pairing a contact

> **Where to find it:** the **Encrypted sync** panel → **Devices and contacts** → **Invite a contact**; before the first pairing, also **Invite / QR**, choose **Contact**. On the other Ciotto: **Receive / Scan** or **Receive**.

> **What it does:** establishes a relationship with another person, who has a Ciotto identity of their own. Pairing is mutual: whoever receives the invite approves it and in turn shows a return invite, to be read on the first device. A contact can send and receive tabs, which stay in the **Inbox** until you open them; a contact gets no access to your personal sync or to any data category. A contact invite cannot bring a device in among your own.

132. Sending a tab to a device or a contact

> **Where to find it:** the **Encrypted sync** panel → **Send this tab**: the recipient (**Choose a device or contact**), **Send** or **Move**, and the **Optional message** field.

> **What it does: Send** sends a copy of the current tab, which also stays here; **Move** sends it and closes it here once it is opened on the other side. To one of your own devices, the reading position, reader mode and the workspace it came from travel too; to a contact, only the address and title. Only ordinary HTTP(S) tabs are allowed: private, Stranger and internal tabs are refused (**Private, Stranger and internal tabs cannot be transferred.**). If the recipient is switched off, the item stays encrypted in the local queue and is retried automatically for seven days. The optional message (up to 500 characters) travels with the tab, but for now the recipient's **Inbox** does not show it.

133. Inbox

> **Where to find it:** the **Encrypted sync** panel → **Inbox**, with the number waiting next to the title. The toolbar button shows a dot while there is something to open, and its tooltip says **Encrypted sync — N waiting**.

> **What it does:** tabs sent from your **own** devices open on their own, on desktop and on the phone, in the workspace they came from if it exists here too. Those from **contacts**, and those from your own devices that could not be opened, wait here (**Tabs from contacts, or trusted-device tabs that could not open automatically, wait here.**): for each one you choose the workspace and press **Open**, or **Dismiss**. An item waits for seven days at most. There is no command to retry a delivery: Ciotto retries it on its own, and also with **Sync now** in Settings.

134. Tabs from other devices

> **Where to find it:** the **Encrypted sync** panel → **Tabs from other devices**; on the New tab page, the **Tabs from your devices** block, switched on in Settings → **New tab & startup** → **New tab: remote tabs**.

What it does: publishes to your **own** devices (never to contacts) an encrypted snapshot of the safe tabs you have open, up to 500, so that you can pick up elsewhere what you were reading. On desktop you browse the list and open an item with a click. A phone, on the other hand, opens those tabs on its own the first time it receives a device's snapshot (**A newly paired phone opens its first snapshot automatically.**); after that, the list remains as a fallback. The **Remote tabs** category must be on.

135. Companion bridge for the Web app and extensions

Where to find it: the Web app and the extension popups: the **Lightning companion** selector.

What it does: the Web app and the extensions do not speak Lightning directly: they go through a native Ciotto device chosen explicitly. The message is already encrypted and gets a second envelope before passing through a Cloudflare rendezvous point with a random token, which keeps nothing and does not log the contents. Once paired as a device, the Web app and the extensions are devices in every respect and receive the categories they can handle: the Web app all of them except notes, data sheets and passwords, the extensions only tabs and saved items.

136. Revoking a device

Where to find it: the **Encrypted sync** panel → **Devices and contacts** → **My devices**, the **Revoke** command; the same list in Settings → **Sync, Lightning & devices**.

What it does: the device stays in the list marked **revoked**, and this Ciotto stops sending it tabs and data and stops accepting what it sends; if a session handoff with that device was in progress, it ends. The key of the passwords category is rotated only if the revoked device was authorized for passwords. **What the device has already received cannot be deleted remotely.** Current limit: the revocation applies only on the device you make it from; your other devices are not informed, and the revoked device can keep syncing through them until you revoke it there too.

137. Password sync with double consent

Where to find it: the **Passwords (authorized devices only)** category in **What follows you**, plus the **Allow password sync** / **Stop password sync** command under each device in **My devices**.

What it does: passwords do not follow you by default. **Two consents** are needed: switching on the category, and authorizing that particular trusted device (**Password sync needs both the local category switch and explicit permission for each trusted device.**). The payload uses a dedicated category key, which is rotated when an authorized device loses its authorization or is revoked.

O. Chrome extensions

138. Installing from the Chrome Web Store

> **Where to find it:** menu bar → **Extensions** → **Open Chrome Web Store**; ⋮ → **Extensions** → **Open Chrome Web Store**.

> **What it does:** installing from the store works directly: you open the store, press the install button (inside Ciotto the label “Add to Chrome” becomes “Add to Ciotto”) and Ciotto asks **Add "..." to Ciotto?**, reminding you that extensions can read and change the data of the sites you visit, with **Add extension** and **Cancel**. Installed extensions live in the application's data folder and update on their own; with the feature in entry 104 switched on, an update that asks for more permissions stays on hold until you approve it. A newly installed extension does not appear in the toolbar by itself: it has to be pinned (entry 140). Ciotto has no command of its own for removing an extension: you delete its folder (entry 139) and restart.

Platforms: desktop only (Windows, macOS, Linux): the web views of Android and iOS cannot host extensions. In the European Union the store shows a cookie consent wall that you have to get past by hand.

139. Unpacked extensions

> **Where to find it:** menu bar → **Extensions** → **Open Extensions Folder**.

> **What it does:** opens the folder where extensions live: put a folder with a `manifest.json` in it and restart, and the extension is loaded. This is how you develop an extension for Ciotto, and it is also the way to remove one.

140. Extension action bar

> **Where to find it:** icons in the toolbar, only for the extensions pinned from ⋮ → **Customise Toolbar** → **Extensions**; all the others in ⋮ → **Extensions**.

> **What it does:** shows in the toolbar the icons of the pinned extensions, which share the toolbar's five slots with commands and macros (**Pinned to the toolbar — N of 5**), and opens their popups. Extensions that are not pinned are reached from ⋮ → **Extensions**, which lists the actions available for the active tab and ends with **Open Chrome Web Store**; if no extension has a button, it says **No extensions with a toolbar button**. The icons are served from inside the application, not from the internet.

141. Extensions in other profiles

> **Where to find it:** ⋮ → **Profiles** → **Manage Profiles...**, the **Run extensions in this profile** checkbox under each named profile; the workspaces panel states the status for each workspace.

> **What it does:** extensions always run in the default profile. For another profile they are switched on with the checkbox, which starts unticked for new profiles (profiles created before the checkbox existed find it ticked): when you tick it, Ciotto loads the extensions into that profile's session; when you untick it, Ciotto unloads them. Current inconsistency: the row of extension icons disappears from the toolbar anyway while a page from a profile other than the default one is in the foreground, even with the checkbox ticked.

P. Downloads, files and printing

142. Downloads and the download shelf

> **Where to find it:** `Ctrl/Cmd+Shift+J`; menu bar → **View** → **Downloads**; ⋮ → **Downloads**; the ⋮ button shows a dot while a download is in progress. On the phone: ⋮ sheet → **Page** → **Downloads**.

> **What it does:** downloaded files appear on a shelf at the bottom of the window, with their progress. From there you can pause and resume them (**Pause**, **Resume**, desktop only), cancel them or remove them from the list with ×, open them with **Open** and show them in their folder with **Show in folder**; **Clear** removes the finished items of the current workspace only. A file that can run code, that was downloaded without HTTPS or that is on a malware list is flagged on its row; for files that can run code and for listed ones, **Open** asks for confirmation before opening them. Downloads belong to the workspace and the profile that started them, and use that identity's signed-in context. **Removing an item from the list never deletes the file from the disk**, on any platform; the list survives a restart, except for private downloads.

143. Download folder and asking where to save

> **Where to find it:** Settings → **Downloads & files**: **Ask where to save each file** and **Download folder**.

> **What it does:** without the prompt switched on, files go straight to the folder you set (or to the system one if left empty). With the prompt on, you are asked for the destination every time. It is a preference of the device: it does not follow you to your other devices.

144. Opening a local file

> **Where to find it:** menu bar → **File** → **Open local file…**; ⋮ → **Page** → **Open local file…**; `Ctrl/Cmd+O`; or drag files or folders onto the tab row or the toolbar. On the phone: ⋮ sheet → **Page** → **Open local file**.

> **What it does:** opens in a tab a file chosen with the system picker (HTML and MHTML pages, PDF, text). On the desktop each dragged file opens in a tab, and a folder opens as a list you can browse. A page opened from the disk can load images, style sheets, scripts and fonts from the web, and from the disk only from its own folder and its subfolders, where a saved page keeps its files. It cannot read responses with `fetch`, XHR or WebSocket, nor embed a site or another folder in a frame: this way a downloaded HTML file cannot read the disk or send its contents elsewhere.

Platforms: dragging and the limits on local pages are desktop features; on Android and iOS you use the system picker.

145. Saving the page

> **Where to find it:** menu bar → **File** → **Save page** → **Web archive (MHTML)...** or **HTML only...**; ⋮ → **Page** → **Save page**; `Ctrl/Cmd+S` saves as an MHTML archive.

> **What it does:** saves the page as a complete web archive (MHTML) or as HTML only. The file is written only after you have explicitly chosen the destination. From a private or Stranger tab, Ciotto warns you first (**This action leaves private browsing.**).

Platforms: desktop only. On the phone, **Save page...** in the ⋮ sheet saves the page to the favorites or the reading list (entry 67).

146. Printing

> **Where to find it:** `Ctrl/Cmd+P`; menu bar → **File** → **Print page...**; ⋮ → **Page** → **Print page...**. On the phone: ⋮ sheet → **Page** → **Print page**.

> **What it does:** on the desktop it first opens a Ciotto print preview: a window with the page already laid out in the PDF viewer, whose buttons let you print, through the system print dialog, or download the PDF. On the phone it opens the system print dialog directly.

147. Exporting the page as a PDF

> **Where to find it:** menu bar → **File** → **Export page as PDF...**; ⋮ → **Page** → **Export page as PDF...**. On the phone: ⋮ sheet → **Page** → **Save or share PDF**.

> **What it does:** produces a PDF of the page directly, without going through the print dialog. On the phone the file is handed to the system share sheet. From a private or Stranger tab, Ciotto warns you first.

148. Built-in PDF viewer

> **Where to find it:** automatic, when you open a link to a PDF or a PDF from the disk.

> **What it does:** on the desktop a PDF opens in a tab with Chromium's viewer, instead of ending up on the download shelf like any other file. From there you can scroll, zoom, search the text, print and save.

Platforms: on iOS the PDF is shown inside the page, with the viewer of the system's web view. On Android Ciotto has no viewer of its own: the PDF is downloaded and opens in the system viewer when you tap **Open** on the download shelf.

149. Capturing the page as an image

> **Where to find it:** menu bar → **File** → **Capture page**, or ⋮ → **Page** → **Capture page**: **Visible area...**, **Full page...**, **Select area...**, **Copy visible area**, **Copy selected area**. On the phone: ⋮

sheet → **Page** → **Capture visible area**.

What it does: takes a picture of the page: the visible area, the whole page or a portion you choose. The result can be saved to a PNG file or copied to the clipboard. Above 50 megapixels the full page is not captured: Ciotto says so (**The full page is larger than the 50 megapixel capture limit.**) and offers **Capture visible area** or **Export PDF**. On Android and iOS only the visible area is offered, handed to the share sheet, because the phone's web view cannot reliably crop the others: the items that are not available are not shown.

150. Installing a site as an application

Where to find it: ⋮ → **Page** → **Install this site as an app…**; managed in Settings → **Web apps** (**Open**, **Uninstall**). On the phone: ⋮ sheet → **Page** → **Install this site as an app**, with **Manage Ciotto apps**.

What it does: starting from the site's manifest, it offers **Create Ciotto app** or **Install in system** (as well as **Cancel**). **Create Ciotto app** creates a local Ciotto application that stays bound to the chosen profile: on the desktop it opens without the browser's bars, on the phone in a Ciotto tab. Any navigation outside the manifest's scope goes back to a normal browser tab. Installing into the operating system is currently not available on any platform: Ciotto says so (**No system web-app installer is available here. Create a Ciotto app instead.**) instead of pretending. Nothing can be installed from a private or Stranger context.

Q. Settings, updates and distribution

151. The fourteen sections of Settings

> **Where to find it:** ⋮ → **Settings…**; menu bar → **File** → **Settings…**; `Ctrl/Cmd+,`. On the phone: ⋮ sheet → **Settings**, full screen, closed with **Done**.

> **What it does:** one central place for all preferences, in fourteen sections gathered into four groups. **You and Ciotto**: *General*, *Appearance & accessibility*, *New tab & startup*, *Search*. **Browsing**: *Tabs & performance*, *Downloads & files*, *Keyboard shortcuts*. **Privacy and identity**: *Privacy & security*, *Passwords & autofill*, *Profiles & workspaces*, *Sync, Lightning & devices*. **Advanced**: *Web apps*, *AI & automations*, *About & updates*. At the top, **Search settings** finds a section even by words that are not in its name (for example, "ollama" leads to AI & automations). Some items say what they belong to, such as the language (**Portable across your trusted devices.**) and the download folder (**Device-local…**).

152. Customizable keyboard shortcuts

> **Where to find it:** Settings → **Keyboard shortcuts**.

> **What it does:** the page is divided in two: on the left you compose the combination by pressing the keys (**Build the shortcut**), on the right you choose from the drop-down what it should do (**Choose the action**). Before anything is saved, the page says what it found: **Free — nothing uses it yet.**, **Already assigned to …** or **The system keeps this one for …**. The target can be a browser action or a saved macro (the **Macros** group); a deleted macro gives its combination back. Below is the complete list, divided into **Tabs and windows**, **Navigation**, **This page**, **Privacy and data**, **Panels**, **Layout** and **Application**, and searchable with **Search actions**; each row has ↺ (**Reset to default**) and × (**Disable**), and at the bottom are **Reset all to defaults** and the combinations **Kept by the system**, each with what it does next to it. While the field is listening for keys, the menu shortcuts are suspended, so pressing Ctrl+T to assign it does not also open a tab in the window behind.

Platforms: desktop only; on the phone the section says so (**Keyboard shortcuts are a desktop feature…**).

153. Importing and exporting browser data

> **Where to find it:** Settings → **General**, the **Import and export browser data** box; exporting passwords is in Settings → **Passwords & autofill** → **Export passwords**.

> **What it does:** imports from an HTML, JSON or CSV file (**Choose HTML, JSON or CSV**) or, on the desktop, directly from **Chrome**, **Edge**, **Firefox** and **Safari** installed on the same computer: from these it reads favorites and history (from Safari, on macOS only, the favorites), while passwords and search engines need an exported file, so no locked store is ever copied behind your back.

> The categories are **Favorites and reading list**, **History**, **Passwords** and **Custom search engines**; you choose the scope of the saved items (**Saved-item scope**: **Current workspace** or **Personal**) and the profile that receives the passwords (**Password destination profile**), then **Import selected**. **It never imports cookies, active sessions, extensions or private data. Export non-sensitive data** exports favorites, reading list, history and search engines. Passwords are exported separately, after entering the master password, with **Export unencrypted CSV**: the file is in plain text, and the page says so.

Platforms: reading installed browsers is desktop only; on Android and iOS you import from a file chosen with the system picker, and exporting passwords is not available.

154. Commands pinned to the toolbar

> **Where to find it:** ⋮ → **Customise Toolbar**; for a macro, also **Add to header** in Macro Studio.

> **What it does:** lets you pin up to five items to the toolbar, chosen from **Browser commands** (twenty-five commands, from new tab to **Burn everything** and **Page DevTools**), enabled **Macros** and **Extensions**, so that repeated operations are one click away. At the top, the menu says how many slots are in use (**Pinned to the toolbar — N of 5**).

Platforms: desktop only; on the phone, the quick actions of the ⋮ sheet play the same role.

155. Default browser

> **Where to find it:** the **Make Ciotto my default browser** button on the start page.

> **What it does:** asks the operating system to open links with Ciotto. On Windows it opens the default apps settings, where the choice remains yours; on Android it uses the system request for the browser role; on iOS it opens the system Settings. The button disappears when Ciotto is already the default browser; on iOS, which gives no way of knowing this, it stays visible. A link opened from another application opens in a new tab, both at launch and when Ciotto is already open.

156. Updates and release channel

> **Where to find it:** menu bar → **Help** → **Check for Updates…**; ⋮ → **Check for Updates…**; Settings → **About & updates** → **Check for updates**. The channel (⋮ → **Release Channel**: **Stable** or **Beta (early releases)**, and **Release channel** in Settings) appears only where automatic updates exist. On the phone: Settings → **About & updates**.

> **What it does:** checks whether a newer version exists. In **automatic** mode (the public desktop distribution) the update is downloaded in the background and a strip says **Ciotto … is ready. Restart to update.**, with **Restart Now** and **Later**: installation happens only when you ask for it. In **manual** mode — the one used by private test builds, and always the one on Android and iPhone — nothing is downloaded in the background: Ciotto asks the site what the current version is (on the desktop shortly after launch and then every four hours, on the phone at launch and every time you return to the app) and, if there is a newer one, says so with a strip at the top

> of the window (**Ciotto … is available.**, with **Download** and **Later**). **Download** opens the download page, which for test builds requires signing in; installing remains something you do yourself. If the site declares that the installed version is no longer supported, the desktop says so (**This version of Ciotto (…) is no longer supported.**, with **Download the Latest Version** and **Quit**) and the phone shows a screen with **Update Ciotto**.

157. Authenticated private distribution

> **Where to find it:** it is not a command in the application: it is the way test builds are delivered.

> **What it does:** the private channel produces exactly these artifacts: an NSIS installer for Windows x64, a DMG for macOS ARM64, an AppImage and a .deb for Linux x64, and a signed APK for Android. For iOS, the Web app and the extensions, no downloadable artifact is produced; the Web app and the extensions show no update commands, while on iOS Settings → **About & updates** still has **Check for updates** (entry 161). A release is rejected if the APK's signature cannot be verified.

158. Interface inspection blocked in packaged builds

> **Where to find it:** menu bar → **View** → **Toggle Interface DevTools** (present only in development builds). The **page** tools are always available: `F12` on Windows and Linux, `Alt+Cmd+I` on macOS (reassignable, entry 152), menu bar → **View** → **Toggle Page DevTools**, ⋮ → **Tools** → **Page DevTools**.

> **What it does:** a packaged build refuses live inspection of Ciotto's interface, so the work done to protect the code at rest is not undone by two clicks in the running program. Inspecting a page you visit is left intact: that is what a browser is for.

Platforms: on Android, inspecting the web view is possible only in debug builds; on iOS it is not enabled; on the phone there are no page developer tools.

159. Help menu: information, source code and crash reports

> **Where to find it:** menu bar → **Help**.

> **What it does:** brings together **Discover Ciotto**, **About Ciotto** (the version, the sentence “The browser that works for you, with macros, AI and sign-in over Lightning.”, the GPL-3.0 license and the address of the source code), **Check for Updates…**, **Ciotto Website**, **Get the Source Code** and the **Report Crashes** checkbox. Crash reports stay off until you switch them on; once on, Ciotto records crashes on the computer and would send them only to a configured collection service, which no build has today. Unticking the checkbox takes effect from the next launch.

Platforms: desktop only; on macOS **About Ciotto** is also in the application menu. On the phone, the version and updates are in Settings → **About & updates**.

R. Platform differences

The features listed so far are described in their desktop form, which is the most complete. The entries in this section sum up what changes elsewhere: they are not additional features, they are the way the same features appear on each target, including the phone interface.

160. Ciotto on Android

> **Where to find it:** the Ciotto application, installed from a signed APK (direct channel) or from the Play channel.

> **What it does:** the same browser in a compact shell (see the entry on the phone interface, further on). The ⋮ button opens a pop-up sheet that replaces the native menus: at the top the row of five quick actions, then **New tab**, **New private tab**, **Reopen closed tab** and four groups — **Page**, **Workspaces and layout**, **Tools**, **Privacy and identity** — and finally **Discover Ciotto** and **Settings**. **Only** the items that really work appear there: a feature not yet ported is absent, rather than present and inert. Left out by choice are the media controls, the page developer tools, the new window, the preset layouts and layout drawing.

161. Ciotto on iOS

> **Where to find it:** the Ciotto application on iPhone and iPad.

> **What it does:** the same compact shell as on Android, with the adaptations WebKit imposes: suspending processes is left to the system, native Picture in Picture is available, and sync is received while the application is active; while the application is suspended, the sender keeps its own encrypted queue. As on Android, a link that asks for a new window opens in the same tab; page dialogs (`alert`, `confirm`, `prompt`) show the site's name and buttons in Ciotto's language; `ciotto://pair` links arrive even when the app is closed. A page that asks for the microphone does not trigger a camera request as well. The location request remains WebKit's own: on iOS, Ciotto does not intercept it.

Distribution note: no iOS package is produced for the private test channel. Settings → **About & updates** still shows the version and **Check for updates**, which reads the current version from the site.

162. Phone interface

> **Where to find it:** Settings → **Appearance & accessibility** → **Mobile interface**; at the top of the ⋮ sheet, the row of quick actions; in the bar, the button with the number of tabs.

> **What it does:** adjusts the compact shell of Android and iOS. **Toolbar position** puts the bar and the tabs at the bottom (**Bottom**, the default) or at the top (**Top**); the reload button is inside the address field. With **Swipe gestures on the address bar**, on by default, a horizontal swipe on the address bar moves to the next or previous tab (the other way round in languages written from

right to left), and a swipe toward the center of the screen — up with the bar at the bottom, down with the bar at the top — opens the tab list. **Quick actions** chooses the five quick actions at the top of the ⋮ sheet (by default **Ask your tabs**, **Notes for this page**, **Reader mode**, **Workspaces** and **What was blocked here**), with arrows to set the order and **Restore defaults**; in the sheet, the "+" next to an item adds it (**Add … to quick actions**), and holding an action down for two seconds lets you drag it to the trash. The tab list has **Search tabs**, new tab, private tab and reopen, and divides tabs into **Open** and **Inactive** (asleep or discarded), with the **Pinned**, **Locked**, **Private**, **Stranger**, **Sleeping** and **Discarded** badges; a tab is closed with × or by swiping it sideways, except for locked ones. After a tab is closed, **Tab closed** appears for five seconds with **Undo**.

Platforms: Android and iOS only. On the phone a window has at most two panels, and the profile is chosen per workspace, not per panel.

163. Companion Web app

> **Where to find it:** an authenticated Web page, opened in any browser.

> **What it does: it is not a browser**: it is a companion for sync. It shows the status and the categories, and lets you pair devices and contacts (**Invite my device**, **Invite a contact**, **Verify and pair**), read the inbox, see the tabs of your other devices, browse synced saved items and change portable preferences (theme, language, Home button, favorites bar). **Send a copy** sends a paired peer the address of the application's own page, only as a copy and without a message. Credential keys never arrive here, and neither do offline copies.

164. Ciotto extensions for Chromium, Firefox and Safari

> **Where to find it:** the popup of the Ciotto extension installed in another browser.

> **What it does:** a limited popup for bringing a tab into the Ciotto world: it sends the current tab to a device or a contact through a native companion (**Send this tab**), saves it to your personal favorites or reading list (**Save favorite**, **Read later**), shows the inbox, the tabs of your other devices and synced saved items, and lets you pair by invitation. It does **not** replace the host browser's address bar, tab groups, downloads or security controls, and it asks for no password-related permission. It asks for the `tabs`, `storage` and `alarms` permissions (on Chromium also `offscreen`) and for access to all http and https addresses.

165. Distribution website

> **Where to find it:** Ciotto's public website.

> **What it does:** presents the product and hosts the download links and the private channel's authenticated page. **It never becomes a runtime environment for the browser**: it is not a version of Ciotto that runs inside the website.

You can download Ciotto from **ciotto.org**, where you will also find this guide in the other languages.