

1. Nextcloud Talk Dashboard Widget: Chat Room View & Interaction

Goal: Build a Dashboard widget that displays the content of a selected Nextcloud Talk room and allows users to react or reply directly from the Dashboard.

Why it's relevant:

- Nextcloud's Dashboard now supports interactive widgets, and Talk's API allows fetching and sending messages programmatically.
- This would streamline communication by reducing the need to switch between apps.

Technical Requirements:

- Use the [Nextcloud Dashboard widget API](#) to create a custom widget.
- Integrate with the [Nextcloud Talk API](#) to fetch and post messages.
- Support real-time updates (e.g., using server-sent events or polling).

Bonus:

- Add message filtering (e.g., by mention, keyword).
 - Support rich message previews (links, images, files).
-

2. External App Launcher Widget

Goal: Create a simple Dashboard widget that lets users add external apps (URL + icon) to their Dashboard, opening them in an iframe or new tab.

Why it's relevant:

- The community has already experimented with [iFrame widgets](#) for embedding external sites.
- This would make Nextcloud a true “digital workspace” hub.

Technical Requirements:

- Develop a widget using Vue.js (Nextcloud's recommended framework).
- Allow users to add/remove/edit app entries via a settings panel.
- Ensure security by validating URLs and using sandboxed iframes.

Bonus:

- Add support for OAuth or API key management for private apps.
-

3. Nextcloud Talk Bots

A. FAQ Bot

Goal: Build a bot that answers FAQs based on a markdown file (structured as # Question, ## Variants, ### Answer).

Why it's relevant:

- Nextcloud Talk supports bots that can send and receive messages, and the community is actively developing bot use cases.
- Useful for onboarding, IT support, or community management.

Technical Requirements:

- Parse a markdown file for Q&A pairs.
- Use the [Talk bot API](#) to listen for commands (e.g., /faq wifi-berlin) and respond with the answer.
- Support fuzzy matching for question variants.

Bonus:

- Allow admins to update the FAQ via a Nextcloud file or admin panel.

B. File Expiry Reminder Bot

Goal: Monitor shared files with expiration dates and post reminders in the relevant Talk room before they expire.

Why it's relevant:

- Nextcloud already supports file sharing with expiry dates, but there's no built-in reminder system.
- This bot would improve data hygiene and compliance.

Technical Requirements:

- Use the [Nextcloud Files API](#) to scan for expiring shares.
- Post reminders in Talk rooms linked to the file or user.

Bonus:

- Allow customizable reminder schedules (e.g., 7 days, 1 day before expiry).

C. Quote-of-the-Day Bot

Goal: Post a daily motivational/funny/custom quote to a designated Talk room.

Why it's relevant:

- Simple but effective for team morale or community engagement.
- Can be extended to pull quotes from an external API or a Nextcloud file.

Technical Requirements:

- Schedule daily messages using a cron job or background task.
- Use the Talk bot API to post messages.

Bonus:

- Let users submit quotes via a Talk command (e.g., `/addquote`).

D. Clean-up Bot

Goal: Post weekly reminders about inactive Talk rooms, old pinned messages, or stale files, suggesting archiving.

Why it's relevant:

- Helps keep workspaces tidy and relevant.
- Can integrate with Nextcloud's existing archiving features.

Technical Requirements:

- Use the Talk and Files APIs to detect inactive rooms/files.
- Post summary reports and action suggestions in Talk.

Bonus:

- Add commands to archive rooms/files directly from the bot (e.g., `/archive-room`).
-

4. Nextcloud ExApp: Enhanced Deck Features

Goal: Extend Nextcloud Deck (Kanban) with new features like dependency tracking and Gantt chart views.

Why it's relevant:

- Deck is a popular project management app, but lacks advanced features like dependencies and Gantt charts.
- ExApps allow development in any language and can integrate deeply with Nextcloud.

Technical Requirements:

- Use the [Nextcloud ExApp API](#) to build a backend service.
- Develop a frontend plugin for Deck to display dependencies and Gantt charts.
- Sync data with Deck's existing API.

Bonus:

- Add critical path analysis or resource allocation features.
-

5. Nextcloud Talk Bridge Bot

Goal: Build a bot that bridges Nextcloud Talk with other chat platforms (e.g., Matrix, IRC, Slack) in a single room.

Why it's relevant:

- Nextcloud already supports [bridging](#), but a user-friendly bot could simplify setup and management.

Technical Requirements:

- Use the Talk bot API to relay messages between platforms.
- Support multiple bridges per room and message formatting conversion.

Bonus:

- Add moderation tools (e.g., filter messages, block users).
-

Resources & Getting Started

- **Nextcloud Talk API Docs:** nextcloud-talk.readthedocs.io
 - **Dashboard Widget Tutorial:** nextcloud.com/developer
 - **ExApp Tutorial:** help.nextcloud.com/t/new-tutorial-develop-your-first-exapp-in-python/205108
 - **Community Examples:** r/NextCloud
-