# Nido Project – GitHub Connector Access Instructions for ChatGPT

This document is intended for **new ChatGPT sessions** interacting with the fwillem/Nido repository via the GitHub Connector. Upload this file at the start of your session to avoid re‑explaining the setup every time. It provides guidelines on how to fetch source files using release tags or interim savepoints created by the user.

## Connector scope and restrictions

* The GitHub Connector is enabled in ChatGPT Web with access to the repository fwillem/Nido. It is not available in the Desktop app.
* The connector can only access repositories explicitly authorised by the user. No other repositories or private files are visible.
* Do **not** request files using raw.githubusercontent.com URLs. Use the connector by specifying the file path and the version (branch, tag or commit) as described below.

## Requesting a file

To access a file, follow these steps:

1. **Ask the user for the relative file path** within the repository (for example, app/src/main/kotlin/com/example/nido/ui/dialogs/BlueScreenOfDeath.kt).
2. **Ask for the version** the user wants to inspect. They should provide either a release tag (e.g. v1.50), a savepoint tag (a timestamp like 2025-08-18\_17-20-25), or a full commit SHA. Without a specified version you risk retrieving outdated content.
3. Use the connector syntax to open the file. For example:

Open app/src/main/kotlin/com/example/nido/ui/dialogs/BlueScreenOfDeath.kt at tag v1.50  
Open app/src/main/kotlin/com/example/nido/ui/dialogs/BlueScreenOfDeath.kt at tag 2025-08-18\_17-20-25  
Open app/src/main/kotlin/com/example/nido/ui/dialogs/BlueScreenOfDeath.kt at commit ebf78973814f533d52ad8d154cda928f1ab488b8

The connector will load the file from the specified snapshot. If the version is not yet available on GitHub, inform the user to push their branch/tag or wait a few minutes for indexing.

## Versions: releases vs. savepoints

The user manages versions of the code using two Git aliases. Understanding these aliases helps you know what to ask for:

### Releases

The command git release <version> creates a new branch and tag with the given version (for example, v1.50), commits the current changes, and pushes both the branch and the tag to the remote repository. When the user refers to a release (e.g. v1.50), you should reference the file at tag v1.50.

### Savepoints

The command git savepoint creates an interim auto‑save commit. It does the following:

1. Stages all changes and commits them with the message “Auto‑save”.
2. Creates a timestamped tag (format YYYY‑MM‑DD\_HH‑MM‑SS).
3. Pushes both the current branch and the timestamp tag to the remote repository.
4. Prints a message of the form:

Created savepoint — Commit: <full SHA> Tag: <timestamp> Branch: <branch>

When the user runs git savepoint, ask them for either the **tag** (the timestamp) or the **commit SHA**. Use whichever reference they provide to open the file. If the connector cannot find the tag or commit, remind the user to push their branch or provide the full hash.

## Handling indexing delays and errors

GitHub’s API may take a short time to index newly pushed commits and tags. In practice, referencing a specific tag or commit often bypasses long delays. If a file does not reflect recent changes:

* Confirm that the user pushed the branch/tag. If not, ask them to run git push --set-upstream origin <branch> for new branches.
* Wait a few minutes and retry the request.
* If the file is urgently needed, ask the user to upload or paste the content directly.

Avoid telling the user to open raw.githubusercontent.com links or to rely on project tree files; using the connector with version references is the recommended approach.

## Summary of commands

* **Release a version:** git release v1.50 → Use tag v1.50 when requesting files.
* **Create a savepoint:** git savepoint → Use the printed Tag or Commit when requesting files (e.g. tag 2025-08-18\_17-20-25).
* **Open a file:** Open <relative/path> at tag <version> or Open <relative/path> at commit <SHA>.

By following these guidelines, new ChatGPT sessions can immediately access the correct version of files in fwillem/Nido without re‑explaining the workflow.

*(Last updated: 18 August 2025)*