

readme.md for

@foss.global/codefeed

Generate an activity feed from a Gitea instance. Scans orgs and repos, retrieves commits since a configurable timestamp, enriches with tags, optional npm publish detection, and CHANGELOG snippets.

Install

```
pnpm add @foss.global/codefeed  
# or  
npm i @foss.global/codefeed
```

Requires Node.js 18+ (global fetch/Request/Response) and ESM.

Quick Start

```
import { CodeFeed } from '@foss.global/codefeed';  
  
// Fetch commits since one week ago (default), no caching  
const feed = new CodeFeed('https://code.example.com', 'gitea_token');  
const commits = await feed.fetchAllCommitsFromInstance();  
console.log(commits);
```

With options

```
const thirtyDays = 30 * 24 * 60 * 60 * 1000;  
const since = new Date(Date.now() - thirtyDays).toISOString();  
  
const feed = new CodeFeed('https://code.example.com', 'gitea_token', since, {  
  enableCache: true,          // keep results in memory
```

```

cacheWindowMs: thirtyDays, // trim cache to this window
enableNpmCheck: true,      // check npm for published versions
taggedOnly: false,        // return all commits (or only tagged)
orgAllowlist: ['myorg'],   // only scan these orgs
orgDenylist: ['archive'], // skip these orgs
repoAllowlist: ['myorg/app1', 'myorg/app2'], // only these repos
repoDenylist: ['myorg/old-repo'], // skip these repos
untilTimestamp: new Date().toISOString(), // optional upper bound
verbose: true,            // print a short metrics summary
});

const commits = await feed.fetchAllCommitsFromInstance();

```

Each returned item follows this shape:

```

interface ICommitResult {
  baseUrl: string;
  org: string;
  repo: string;
  timestamp: string; // ISO date
  hash: string; // commit SHA
  commitMessage: string;
  tagged: boolean; // commit is pointed to by a tag
  publishedOnNpm: boolean; // only when npm check enabled and tag matches
  prettyAgoTime: string; // human-readable diff
  changelog: string | undefined; // snippet for matching tag version
}

```

Features

- Pagination for orgs, repos, commits, and tags (no missing pages)
- Retries with exponential backoff for 429/5xx and network errors
- CHANGELOG discovery with case variants (`CHANGELOG.md`, `changelog.md`, `docs/CHANGELOG.md`)
- Tag-to-version mapping based on tag names (`vX.Y.Z` → `X.Y.Z`)
- Optional npm publish detection via `@org/repo` package versions
- In-memory caching with window trimming and stable sorting
- Allow/deny filters for orgs and repos, optional time upper bound
- One-line metrics summary when `verbose: true`

Environment

- Gitea base URL and an optional token with read access
- Node.js 18+ (global fetch)

Testing

The repo contains:

- An integration test using a `GITEA_TOKEN` from `.nogit/` via `@push.rocks/qenv`.
- A mocked pagination test that does not require network.

Run tests:

```
pnpm test
```

For the integration test, ensure `GITEA_TOKEN` is provided (e.g., via `.nogit/` as used in `test/test.ts`).

Notes

- When `taggedOnly` is enabled, the feed includes only commits associated with tags.
- `publishedOnNpm` is computed by matching the tag-derived version against the npm registry for `@org/repo`.
- For very large instances, consider using allowlists/denylists and enabling caching for incremental runs.

Revision #5

Created 2026-03-28 10:49:25 UTC by foss.global Team

Updated 2026-03-28 12:15:21 UTC by foss.global Team