TECH STYLE LTD

TECH-STYLE.CO

DEVELOPER PACK

Open Source Pack

The packages we publish on npm, PyPI, NuGet and pkg.go.dev, their licences and how to install them.

WHAT IS INSIDE

- Install and go
- pgrecon
- json-compress
- secure-keystore-js
- strapi-content-sync-pro On Strapi Marketplace
- strapi-provider-upload-webdav
- strapi-remote-backup-pro In development
- Not just utilities — whole products
- Also on GitHub
- Because the alternative is rewriting it

Document	Open Source Pack
Generated	2 September 2026
Online	https://www.tech-style.co/docs/open-source-pack

When we solve a problem that isn't specific to our business, we publish it. Seven packages across four registries — including json-compress, one wire format with six interoperable implementations, three of them published so far on npm, NuGet and pkg.go.dev — plus platforms, plugins and tools on GitHub under MIT, Apache 2.0 or AGPL 3.0. Commercial licensing is available for the AGPL products.

PUBLISHED ON NPM, NUGET, PYPI & PKG.GO.DEV

Install and go

Seven packages across four registries. json-compress ships today on npm, NuGet and pkg.go.dev — three of the six implementations of one wire format that all write the same bytes; the PHP, Python and Rust ports build from the repository and are queued for release. Alongside them, pgrecon on PyPI under Apache 2.0 and three more on npm, one of those also carried on the official Strapi Marketplace, plus an agentless Strapi backup tool in development. All extracted from problems we hit doing the work.

pgrecon

Migration reconnaissance for PostgreSQL. Oracle-to-PostgreSQL projects overrun because nobody measures the PL/SQL before committing to a date — pgrecon is the measurement. It parses your DDL and program units with a real grammar, applies 76 rules, reports every incompatibility with evidence and a remedy, turns the result into an effort estimate in person-days that shows its own arithmetic, and converts the schema it can prove into PostgreSQL DDL.

It never connects to your database. It generates a read-only SQL*Plus script your DBA reads and runs; analysis happens locally on the output files. That single design decision is why security teams approve it in minutes.

```
$ pip install pgrecon
```

Offline Analysis | Grammar-based PL/SQL Parsing | 76 Rules | Severity & Remedies | Effort Estimation | JSON Export | Oracle 9.2 > 21c | Python 3.11+ | Apache 2.0

json-compress

Lossless JSON compression that is still JSON. It writes every key name once instead of once per record, turns arrays of like-shaped objects into columns and rows, hoists out the columns whose value never changes, and de-duplicates the values that keep coming back. Record-shaped documents come out 45–75% smaller, and decompress hands back exactly what went in — key order included.

It began as an internal helper for moving large result sets between a database layer and a set of front ends, where the payloads were dominated by repeated field names. It kept being copied from project to project, so we wrote it properly, specified the wire format so it could be ported, and then ported it — six times, into JavaScript, .NET, Python, PHP, Go and Rust.

```
$ npm install @tech-style/json-compress
```

```
$ dotnet add package TechStyle.JsonCompress
```

```
$ go get github.com/eharain/JSON-Compress/go
```

6 Interoperable Implementations · Key Catalogue · Columnar Tables · Constant & Dictionary Columns · Value De-duplication · Byte-perfect Round Trip · Zero Dependencies · npm · NuGet · pkg.go.dev · TypeScript Types · CLI Included · Documented Wire Format · MIT

secure-keystore-js

A secure key-value store with two-layer encryption. A single user key unlocks a per-store master key, and that master key unlocks each value individually — so secrets sit encrypted on disk and are only decrypted, in memory, at the moment you ask for one.

It was built because credentials kept ending up in plaintext config files across our own services. It now backs credential storage in strapi-content-sync-pro.

```
$ npm install secure-keystore-js
```

Two-layer Encryption · AES-256-GCM · Per-key Encryption · Multiple Stores · Import / Export · CLI Tool · Licence pending

strapi-content-sync-pro On Strapi Marketplace

Copy, migrate and live-sync content, media and data between Strapi environments. It handles the awkward parts of moving a CMS between staging and production: relations that must land in order, media that shouldn't be re-uploaded, and fields that should never cross the boundary at all.

It is published on the official Strapi Marketplace, listed under Content, Data, Database & Migrations and Import & Export — so it installs and updates alongside the rest of a Strapi project's plugins.

Paired & Single-side Modes · Bi-directional Sync · Media Sync (HTTP & rsync) · Field-level Policies · On-demand · Scheduled · Live · Large Dataset Pagination · Dependency Analytics · Bulk Transfer · Pause / Resume / Cancel · Persisted History · Retention Controls · Alerts & Logging · API Tokens & HMAC-SHA256 · MIT

```
$ npm install strapi-content-sync-pro
```

strapi-provider-upload-webdav

A WebDAV upload provider for Strapi v5. Point the media library at any WebDAV-capable storage — a NAS, a shared host, a self-managed box — instead of an object-store bucket you'd rather not pay for or depend on.

```
$ npm install strapi-provider-upload-webdav
```

Strapi v5 · WebDAV · Self-hosted Storage · Licence pending

strapi-remote-backup-pro In development

Back up and restore any Strapi v4 or v5 instance remotely — without installing a plugin in it. Every other Strapi backup tool is a plugin, which means adding a dependency to a live CMS and redeploying it. This one authenticates against the same admin API the Strapi admin panel itself uses, so there is nothing to install, nothing to redeploy, and nothing running inside your production process.

It ships as a desktop app for Windows, macOS and Linux and as a CLI for developers, CI and cron — both driving one engine, so anything the app can do a terminal can do too. Restore is selective: pick content types, pick records, choose how many relation hops to follow, and review a diff before anything is written to a live instance.

No Plugin Installed · Strapi v4 & v5 · Desktop App & CLI · Content · Media · Schemas · Relation-depth Restore · Reviewable Restore Diff · Local · S3 · Azure · Drive · Dropbox · OneDrive · SFTP · FTP · Scheduling & Retention · AES-256-GCM Archives · Documented Format · MIT

```
$ npx strapi-remote-backup-pro backup
```

OPEN-SOURCE PLATFORMS

Not just utilities — whole products

Five of our platforms are open source in full, licence and all — including the whole Rutba estate, published across seven repositories.

ACTIVELY DEVELOPED

Rutba Suite

The next generation of our business software, and it is open too: a new Koa + knex core engine, every app group, the background workers, the control plane and the Electron desktop shells — seven repositories under github.com/eharain. AGPL 3.0, with commercial licensing available. Actively developed

ACTIVELY DEVELOPED

Rutba ERP

The first generation, and still maintained — 22 Next.js applications on a shared API covering point of sale, inventory, manufacturing, orders, marketplace, helpdesk, HR, payroll and accounting. MIT, and yours to run yourself. Actively developed

ACTIVELY DEVELOPED

Rutba Media FileServer

A masters-only media origin with on-demand resizing and LRU caching — plus an optional digital asset platform: index, versions, collections, MFA, WebDAV and AI enrichment. AGPL 3.0, with commercial licensing available. Actively developed

ACTIVELY DEVELOPED

Rutba MTA

Multi-tenant email relay middleware with reputation-based pacing, suppression management, bounce capture and signed webhooks. AGPL 3.0, with commercial licensing available. Actively developed

ACTIVELY DEVELOPED

pgrecon

Oracle-to-PostgreSQL migration reconnaissance — a read-only extraction script, grammar-based PL/SQL parsing, 76 rules with evidence and remedies, an effort estimate in person-days, and a converter that emits PostgreSQL DDL for what it can prove. Apache 2.0. Actively developed

MORE REPOSITORIES

Also on GitHub

The estate repositories, plus the smaller tools and plugins — some published, some still repository-only. All open to read, fork and use.

The Rutba estate — seven repositories

rutba

The workspace root — the design record, the repo map and the guardrail hooks

rutba-suite

The consumer tier — the core engine, the shared packages and every app group · AGPL 3.0 + commercial

Rutba-Workers

Background processors — publishing, marketplace sync, interactions and video render

Rutba-Management

The control plane — directory authentication, organizations, licensing and billing

Rutba-Native-Apps

Windows desktop shells for POS, Mail and Studio, over the offline sync framework · AGPL 3.0 + commercial

Rutba-MTA & Rutba-Media-FileServer

The two workers that are whole products in their own right, each with its own repository and history

Tools & plugins

strapi-permission-manager-pro

Strapi permission management plugin · MIT

strapi-api-guard-pro

Strapi API protection plugin

strapi-to-strapi-data-sync

The earlier Strapi-to-Strapi sync plugin · MIT

Strapi-Remote-Backup-Pro

Agentless scheduled backup and restore for Strapi · in development

Because the alternative is rewriting it

Every one of these packages started as internal code. A keystore because credentials were sitting in config files. A sync plugin because staging and production kept drifting. A media server because pre-generating six variants per image was costing more disk than the business justified. A migration assessor because every Oracle project we joined had already committed to a date before anyone read the PL/SQL. A remote backup tool because clients kept needing their Strapi instance backed up on hosting we had no deploy access to.

Publishing them costs us documentation time and gains us three things: the discipline of writing an interface someone else can use, the scrutiny of people who don't already know how it works, and the certainty that we can still use our own tools if our priorities change.

We use permissive licences — MIT, or Apache 2.0 where patent grants matter — because a restrictive licence on a small utility helps nobody.

Using our packages?

Free for commercial use under MIT / Apache 2.0 · Issues and pull requests welcome on GitHub · Commercial support and custom work available · Happy to talk through an integration before you commit

CONTACT

Talk to us

Tech Style Ltd is a UK technology company building enterprise software, procurement intelligence and AI platforms. Tell us what you are trying to do and we will tell you honestly whether we are the right people for it.

This document online	www.tech-style.co/open-source
Website	www.tech-style.co
Enquiries	hello@tech-style.co
Products	www.tech-style.co/products
Partner programme	www.tech-style.co/partners

This document was generated on request.

It is built from the live content of tech-style.co at the moment you downloaded it, so it can never quote a price, a version or a status the site has since moved on from. Request a fresh copy any time at www.tech-style.co/downloads.