

Data Positioning Shared Library

OWASP passed

npm v0.3.319 License MIT

A library containing common constants, types and utilities used across all Data Positioning projects.

Installation

Install as a production dependency:

```
npm install @dataposition/dataposition-shared
```

See the Data Positioning security documentation for additional initialization requirements.

Declarations

This package provides constants, errors, types/interfaces and utilities used by Data Positioning modules.

Modules

The Data Positioning solution consists of the following modules. All modules, except `App`, extend the base type `Module`.

Type	Dynamic	Notes
App		Implements the data positioning web application.
Engine	✓	Implements the data positioning engine.
Connector	✓	Implements a connector which handles one or more connections.
Context	✓	Implements a context which defines one or more models.
Presenter	✓	Implements a presenter which renders one or more presentations.
Tool	✓	Implements...

Components

Each module implements a set of components. All module component types extend the base component types.

Types	Notes
Component	The Component type serves as a base type for all components.
ComponentReference	

ComponentStatus	
ComponentStatusId	
ComponentTypeId	
StatusColorId	

Connector Module Components

Item	Notes
Connector Types	Connector types. The Connector type extends the Component type.
Connection Types	Connection types. The Connection type extends the Component type.

Context Module Components

Item	Notes
Context Types	Context types. The Context type extends the Component type.
Data View Types	DataView types. The DataView type extends the Component type.
Dimension Types	Dimension types. The Dimension type extends the Component type.
Engine Types	Engine types.
Event Query Types	Event Query types. The Event Query type extends the Component type.

Engine Module Components

Item	Notes
Engine Types	Engine types.

Presenter Module Components

Item	Notes
Presenter Types	Presenter types. The Presenter type extends the Component type.
Presentation Types	Presentation types. The Presentation type extends the Component type.

Composables

Constants

Errors

Utilities

Usage

Import the library in your TypeScript project:

```
import { type ConnectorConfig, getComponentStatus } from '@datapost/datapost-shared';

// Example type usage.
let connectorConfig: ConnectorConfig;

// Example function usage.
getComponentStatus('alpha');
```

Implements the common Data Positioning repository management command set. For more information see [@datapost/datapost-development](#).

Bundle Analysis Reports

The Bundle Analysis Report provides a detailed breakdown of the bundle's composition and module sizes, helping to identify which modules contribute most to the final build. It is generated automatically on each release using the npm package `rollup-plugin-visualizer`.

[View the Bundle Analysis Report](#)

Dependency Check Report

The OWASP Dependency Check Report identifies known vulnerabilities in project dependencies. It is generated automatically on each release using the npm package [owasp-dependency-check](#). We also rely on GitHub Dependabot to continuously check for vulnerabilities across all dependencies.

[View the OWASP Dependency Check Report](#)

Dependency Licenses

The following table lists top-level production and peer dependencies. All these dependencies (including transitive ones) have been recursively verified to use Apache-2.0, BSD-2-Clause, CC0-1.0, or MIT—commercially friendly licenses with minimal restrictions. Developers cloning this repository should independently verify dev and optional dependencies; users of the uploaded library are covered by these checks. We do not include unlicensed dependencies. Used to support development activity and not released as part of the production release. Check if you clone. We use the npm packages [license-report](#), [license-report-check](#) and [license-report-recursive](#) to identify dependency licenses.

The following table lists top-level production and peer dependencies. All these dependencies (including transitive ones) have been recursively verified to use Apache-2.0, BSD-2-Clause, CC0-1.0, or MIT—commercially friendly licenses with minimal restrictions. Developers cloning this repository should independently verify dev and optional dependencies; users of the published library are covered by these checks. We do not include unlicensed dependencies. Used to support development activity and not released as part of the production release. Check if you clone. We use the npm packages [license-report](#), [license-report-check](#) and [license-report-recursive](#) to identify dependency licenses.

Name	Type	Installed	Latest	Latest Released	Deps	Document
csv-parse	MIT	6.1.0	6.1.0	5 months ago: 2025-07-16	0	⚠ No license file

date-fns	MIT	4.1.0	4.1.0	4 months ago: 2025-08-03	0	LICENSE.md
nanoid	MIT	5.1.6	5.1.6	2 months ago: 2025-09-22	0	LICENSE

Installed dependencies are kept up-to-date with latest releases.

License

[MIT](#) © 2026 Data Positioning Pty Ltd