

---

# **MyPureScriptLibraries Documentation**

**Justin Woo**

**Jul 01, 2020**



---

## Contents

---

<b>1</b>	<b>Pages</b>	<b>3</b>
1.1	For Everyone . . . . .	3
1.2	Cool things . . . . .	3
1.3	Node-related . . . . .	5
1.4	Type level demos . . . . .	6
1.5	Type level utils . . . . .	6
1.6	Codegen-related . . . . .	7
1.7	Outdated and stream related . . . . .	8
1.8	Related to Redux Interop . . . . .	8
1.9	Other Demos . . . . .	8



DEPRECATED; this content will be moved to GitHub for ease of maintenance.

---

This is a glorified list of libraries that I have written for PureScript.

If any of these don't build, let me know on Twitter and I can help you out. It's probably just a small tweak away.

---

**Note:** Suggest improvements on descriptions and report missing links on the Github issues

---



## 1.1 For Everyone

These libraries are going to be useful for basically anyone who touches PureScript.

### 1.1.1 Spacchetti

Dhall-based configuration for package-sets. An alternative to using [package-sets](#) (which I also maintain).

### 1.1.2 Simple-Json

Library for serializing JSON automatically for type aliases, especially of record types using row type information. An alternative to the [Argonaut](#) libraries.

### 1.1.3 Milkis

Library for giving you a basically Union-typed interface for `fetch` on Browser and on Node. A simple to use alternative to the [Affjax](#) library.

### 1.1.4 Chirashi

An easy way to work with Errors by inserting a Variant, and reading it out later.

## 1.2 Cool things

Cool libraries using types to accomplish very practical things

### 1.2.1 Choco-Pie

Library for creating cycles of PureScript-Behaviors Events, where you can make individual “drivers” for handling queries for effects and work with a pure “main” function that takes the driver events in as “sources” and outputs “sinks” of queries to the drivers. Similar to Cycle.js

### 1.2.2 Kishimen

Use sum types with Generic instances to get Variants out. Useful for JSON deserialization among other things.

### 1.2.3 Mochi

Create record constructor functions from record types.

### 1.2.4 Tortellini

Library for parsing and decoding INI files to a record type directly.

### 1.2.5 Lenient-Html-Parser

A lenient HTML parser, so you can work with a List of Tags parsed. Kind of like TagSoup.

### 1.2.6 Sijidou

A library for matching a string into a record of functions of statically known Symbol proxies.

### 1.2.7 Naporitan

A library to reflect a record of proxies, so you don’t have to repeat it and all.

### 1.2.8 Xiaomian

A library to easily use Naporitan with a record of `SProxy` of keys of a row type.

### 1.2.9 Bibimbap

The flip side of `Chirashi`, simply storing sum type values and retrieving them by using the constructor name as a test value, by using `Generic.Rep` to get constructor names from a sum type definition.

### 1.2.10 Kushiya

Library (for 0.12) for parsing a `Symbol` template and getting a function to parse a `String` into a record.

### 1.2.11 Kushikatsu

A simple `Variant (Unmatched String) (routes :: # Type)` routing library built on Kushiya.

### 1.2.12 Home-Run-Ball

Library for refinements on values using a row type to track validations and automatically apply validations in the row type parameter.

### 1.2.13 Jajanmen

Cool type-safe Symbol query parameterized helper for Node-SQLite3.

### 1.2.14 Gomtang-Basic

Echarts with Union constraints.

### 1.2.15 Record-Extra

Extra generic Record functions. More useful as a reference for how to implement some specific things in your projects.

## 1.3 Node-related

Things that I use on Node.js.

### 1.3.1 Toppokki

Bindings for [Puppeteer](#), for when you need to do things with headless Chrome.

### 1.3.2 Node-He

A wrapper for he to work with HTML entities.

### 1.3.3 Node-Telegram-Bot-API

A wrapper and partial typing for Telegram api.

### 1.3.4 Makkori

Wrapper for Express, with almost no assumptions, using Union constraints for options.

### 1.3.5 Node-Sqlite3

A sqlite3 wrapper.

### 1.3.6 Sunde

Aff interface for spawning child processes.

## 1.4 Type level demos

These are largely just demonstrations of what you can do with types in PureScript and not used for things.

### 1.4.1 Hotteok

For handling JS Union values. Basically Advanced TypeScript with more truthful qualifiers as a library.

### 1.4.2 Dango

Sized lists using Symbols for keeping track of size using “.” Symbols, e.g. “...” for size-3 lists.

### 1.4.3 Kazunoko

Symbol-literal arithmetic for small numbers, e.g. Add “one” “two” “three”

### 1.4.4 Chahan

Generic Sum and Product into type-level List.

### 1.4.5 Roskis

IxMonad for managing a mutated record. Probably you will always want to use Record.Builder or ST-something instead, but this might be interesting for some.

### 1.4.6 La-Galbi

Type-level path string formatting. You don’t really need this with Symbol.Cons now giving you much better ways to do with Record-Format, but you might find this interesting.

## 1.5 Type level utils

Type level programming utilities and such.

### 1.5.1 Expect-Inferred

Library for checking inferred types of values.

### 1.5.2 Biang

A library for converting generic reps of Sum and Product to row types, for whatever convenient usage needs at the type level.

### 1.5.3 AlphaSucc

Alphabetical successions for providing labels for type-level programming.

### 1.5.4 Type-IsEqual

Type equality result as Boolean kind for use in type-level programming

### 1.5.5 Shoronpo

A library for type-level Symbol formatting with other Symbols, e.g. intercalated record labels

### 1.5.6 Tanghulu

A library for working with unlimited type-level naturals using an encoding of `.` in Symbol.

## 1.6 Codegen-related

These are projects for generating code and types for other languages from PureScript. Again, you can use some of these directly, but you should mostly use them for inspiration.

Many of these are probably out of date slightly. Let me know if you want to use some of these and I can help you upgrade them and learn how they work.

### 1.6.1 Has-JS-Rep

For testing that something has a JS representation with constraints.

### 1.6.2 Ohyes

Library for generating TypeScript types of constrained types.

### 1.6.3 Bismuth

Library for generating Flow types of constrained types.

### 1.6.4 Kancho

Library for generating Elm types of constrained types.

### 1.6.5 Bundaegi

Library for generating TypeScript types and io-ts schemas.

## 1.7 Outdated and stream related

These are outdated and related to streams. Please see the Event/Behaviors libraries and ChocoPie if you want to see some stream-related libraries.

### 1.7.1 Cycle-Run

Cycle.js FFI

### 1.7.2 Xstream

Xstream FFI

### 1.7.3 Observable-Lift

Function for lifting effectful operations to PureScript-Observable

## 1.8 Related to Redux Interop

I'm sorry you clicked on this link. Use the projects as references to what you might do in your own project, but don't expect them to be that useful by themselves.

None of these are made for beginners, so expect to have to refer to <https://github.com/purescript/documentation/> and learn a lot about PureScript before making use of these.

### 1.8.1 Redux-DevTools

Some bindings to work with Redux DevTools.

### 1.8.2 Gorebab

Redux-Observable typed more correctly.

### 1.8.3 Chapagetti

React-Redux with row Unions.

### 1.8.4 Jolly-Pong

Well-typed Redux duct taping.

## 1.9 Other Demos

Examples and demos of doing some practical things in PureScript that I don't use for anything personally. Might be interesting to use for some problems you run into.

### 1.9.1 Puchitomato

A small module for a string literal type that guarantees a String value the same as its Symbol parameter. This actually is something that has come up multiple times for me, so maybe using this with Simple-JSON would be good.

### 1.9.2 Chanpon

Record-based query building.

### 1.9.3 Ffi-Props

Library to work with FFI props with row type parameters.

### 1.9.4 Soondubu

Chart.js with Union constraints.

### 1.9.5 Dotorimuk

A Chart.js POC library. Less useful but with some food for thought.