

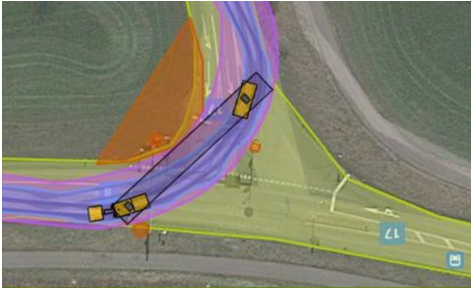
Runtypes

Who: Erik

Product: HeavyGoods.net

Lang: Typescript

Simulation vs. Reality



We use Typescript

Isn't that wildly unsafe still?

when developing:

Typescript

~~~~~

Javascript

when running:

Javascript

```
const t: Truck = JSON.parse(data)    // 😊
```

```
const t = JSON.parse(data)           // 😞
```

# Runtypes Idea

We need to implement the check ourselves:

```
function truck(x: unknown) ⇒ Truck
```

```
const t = truck(JSON.parse(data)) // it's a Truck
```

# Runtypes Schema

And we make it easy to write these runtime type definitions, e.g. by using fp style combinators:

```
const truck = record({  
  id: string(),  
  weight: integer(),  
  model: string(),  
})  
  
const t = truck(JSON.parse(data)) // it's a Truck
```

# Build or Buy

Tons good of libraries have been already written:

- functional / with exceptions / with explicit workflow / integration rich / fat or small API / 0-deps / ...
- @ decoration based (like Java, if you like that)
- heavily performance optimized with eval
- AOT compilers going full 🦾 berserk mode to optimize your checks as if you wrote them yourselves

# Benchmark

Fortunately, there is a nice little benchmark for many of them:

[github.com/molnar/typescript-runtime-type-benchmarks](https://github.com/molnar/typescript-runtime-type-benchmarks)

## Runtype Benchmarks

[Github Repository](#)

Benchmark Comparison of Packages with Runtime Validation and TypeScript Support

Benchmarks:

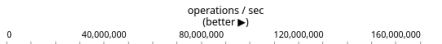
- ☒ Safe Parsing
- ☒ Strict Parsing
- ☒ Loose Assertion
- ☒ Strict Assertion

Node.js Versions:

- ☒ v20.11.0
- ☐ v19.9.0
- ☐ v18.19.0
- ☐ v16.20.2

Sort:

Fastest



# Recap

Always validate external data in Typescript.

Use runtypes for this and only write your schema once.

Visit [github.com/moltar/typescript-runtime-type-benchmarks](https://github.com/moltar/typescript-runtime-type-benchmarks)  
to find a nice library for your needs.

Or build your own. Happy Coding.