Zod

https://nathankrasney.com/

What is zod

- Validation library
- TypeScript-first schema declaration and validation library.
- The goal is to eliminate duplicative type declarations - how is this achieved ??
- With Zod, you declare a validator *once* and Zod will automatically infer the static TypeScript type

https://nathankrasney.com/

Motivation for zod

- Simple
- Designed to be as developer-friendly as possible
- Popular - from 600K weekly downloads in june 2022 to 3.5M in june 2023
- Works in node and browsers
- Works also in javascript not just typescript
- The error thrown are super informative

https://nathankrasney.com/

Motivation for zod for me

- Use all ready defined interface from types directory to define the zod schema - is it possible ??
- Simple validation library

https://nathankrasney.com/

Use cases

Client

- Validation of form input
- Dynamic type check of json received from a server (check my video for motivation but validation here with json schema)

Server

- Validate info from client via http request
- Validate data into schema based database SQL

Simple string validation - safeParse

```
function validateStringSafeParse(
  val: any
): SafeParseReturnType<string, string> {
  const schema = z.string();
  return schema.safeParse(val);
}
…..
console.log(validateStringSafeParse("111")); //  { success: true, data: "111" }
console.log(validateStringSafeParse(111)); // { success: false, error: Getter }
```

Define the schema

Check value against the schema
safeParse **return an object** with
success : true\false and data | error

string-validators.ts (ver 0.1)

https://nathankrasney.com/

Simple string validation - parse

```
function validateStringParse(val: any): void {
  const schema = z.string();
  schema.parse(val);
}
……
validateStringParse("111"); // ok
validateStringParse(111); // throw
```

Define the schema

Check value against the schema
Throw if not valid

```
ZodError: [
  {
    "code": "invalid_type",
    "expected": "string",
    "received": "number",
    "path": [],
    "message": "Expected string, received number"
  }
]
```

Very informative info thrown

string-validators.ts (ver 0.1)

https://nathankrasney.com/

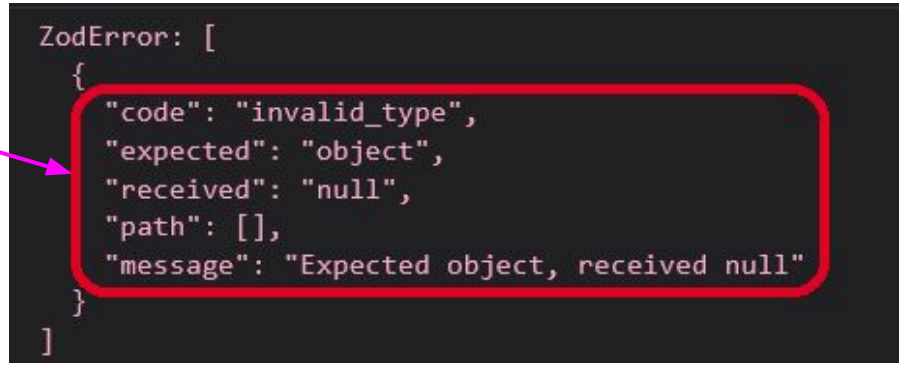
Validate object - sample 1

```
function validatePerson(person: any): void {
  const schemaUser = z.object({name: z.string(),age: z.number(),});
  schemaUser.parse(person);
}
```

object-validators.ts (ver 0.3)

Check value against the schema
Throw if not valid

```
……
const personOk: IPerson = {name: "John Doe",age: 10,};
validatePerson(personOk); // do not throw
validatePerson(null); // throw
```



https://nathankrasney.com/

Validate object - sample 2

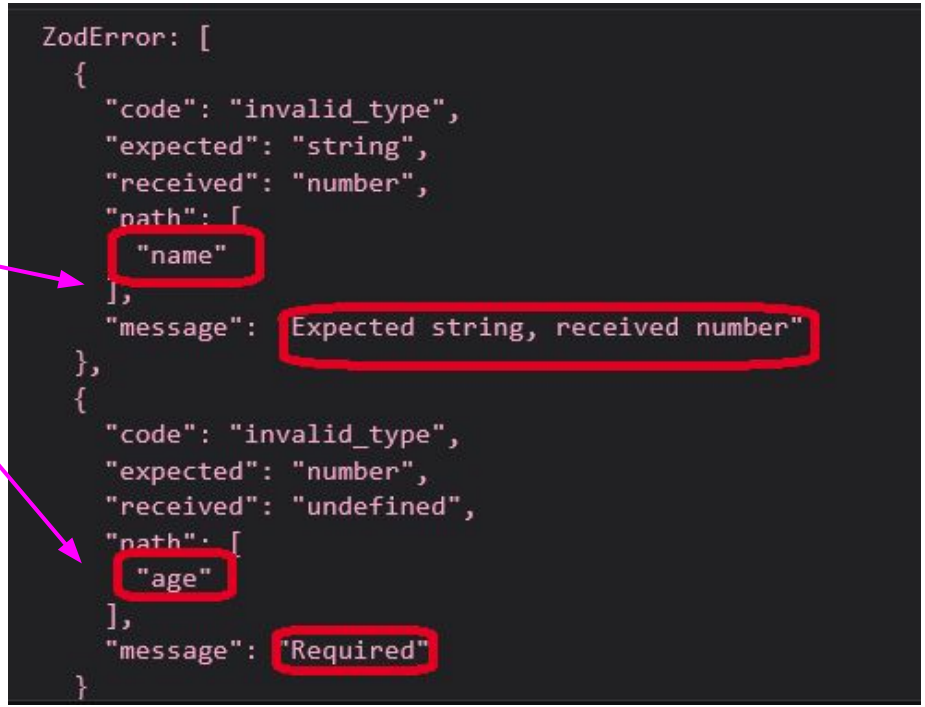
validatePerson({name : 11}); // should throw

Same function from last slide

……

The error are super informative !!!!!!!

Path allow to refer to the object properties



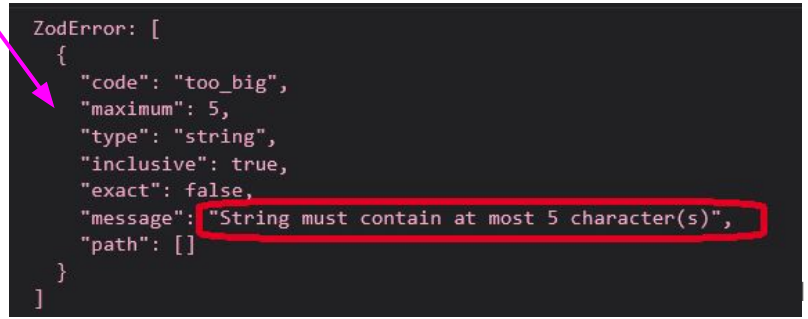
https://nathankrasney.com/

Parse string with limits

```
function validateStringMinMax(val: any): void {
  const schema = z.string().min(3).max(5);
  schema.parse(val);
}
....
validateStringMinMax("ab12"); // ok , not throw
validateStringMinMax("ab1222");; // throw
```

Define the schema

Check value against the schema
Throw if not valid



https://nathankrasney.com/

display errors concepts

You can use the `.format()` method to convert this error into a nested object.

```
const result = z
  .object({
    name: z.string(),
  })
  .safeParse({ name: 12 });

if (!result.success) {
  const formatted = result.error.format();
  /* {
    name: { _errors: [ 'Expected string, received number' ] }
  } */

  formatted.name?._errors;
  // => ["Expected string, received number"]
}
```

name is a key in the schema !!

https://nathankrasney.com/

Display errors in a form sample 1/2

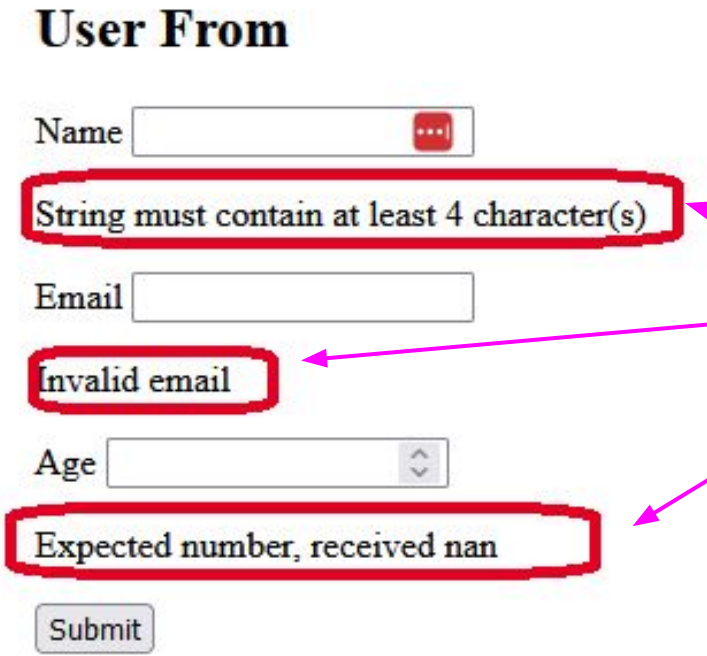
```
const validationResult = formZodSchema.safeParse({
    name: getNameVal(),
    email: getEmailVal(),
    age: getAgeVal(),
  });

  if (validationResult.success) {
    getNameError().innerText = getEmailError().innerText = getAgeError().innerText ="";
  } else {
    const formatted = validationResult.error.format();
    getNameError().innerText = formatted.name?._errors.join(" , ") ?? "";
    getEmailError().innerText = formatted.email?._errors.join(" , ") ?? "";
    getAgeError().innerText = formatted.age?._errors.join(" , ") ?? "";
  }
```

Check validateFormWithZod
form-validation.ts (ver 0.3)

https://nathankrasney.com/

Display errors in a form sample 2/2



Following the code from prev slide - these are the errors we get for the schema

```
const formZodSchema = z.object({
  name: z.string().min(4).max(7),
  email: z.string().email(),
  age: z.number().min(18).max(65),
});
```

https://nathankrasney.com/

My repository

zod-validation-playground

https://nathankrasney.com/

questions

- Can i create schema using already existing interface?? This is the most important question for me - check here , check also ts-to-zod
- How the inferred interface can be useful - does it help if i all ready have the interface defined in types directory ??

https://nathankrasney.com/

references

Official docs

Zod Makes TypeScript Even Better - nov 2022

Fixing TypeScript's Blindspot: Runtime Typechecking - sep 2021