

Next-gen Devtools for Effect



Ego Slide

- Push buttons
- Type things
- Break stuff
- Wear sweatshirt
- This picture is not stack safe
- Does anyone care about stack safety?

It all starts with... code?

Argument of type '() => Generator<YieldWrap<Effect<Order, OrderNotFoundError, never>> | Effect<number, ProductNotFoundError | PricingNotFoundError, never>, number, any>' is not assignable to parameter of type '(resume: Adapter) => Generator<YieldWrap<Effect<any, any, any>>, number, never>'.

Call signature return types 'Generator<YieldWrap<Effect<Order, OrderNotFoundError, never>> | Effect<number, ProductNotFoundError | PricingNotFoundError, never>, number, any>' and 'Generator<YieldWrap<Effect<any, any, any>>, number, never>' are incompatible.

The types returned by 'next(...)' are incompatible between these types.

Type 'IteratorResult<YieldWrap<Effect<Order, OrderNotFoundError, never>> | Effect<number, ProductNotFoundError | PricingNotFoundError, never>, number>' is not assignable to type 'IteratorResult<YieldWrap<Effect<any, any, any>>, number>'.

Type 'IteratorYieldResult<YieldWrap<Effect<Order, OrderNotFoundError, never>> | Effect<number, ProductNotFoundError | PricingNotFoundError, never>>' is not assignable to type 'IteratorResult<YieldWrap<Effect<any, any, any>>, number>'.



@effect/language-service (already available!)

- Works with any editor that supports TypeScript Language Service Protocol
- Provides additional Effect diagnostics
- Provides refactors for Effect code

TS sample-01.ts 1, U

▷ ⌂ ⏪ ⏴ ...

examples > `TS sample-01.ts` > `getOrderSubtotal` > `Effect.gen() callbacks`

```
1 import { Effect } from "effect"
2 import { getOrderById, getPriceOfProduct } from "./order"
3
4 export function getOrderSubtotal(orderId: string) {
5   return Effect.gen(function* () { Argument of type '() => Generator<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFound>
6     const order = yield* getOrderById(orderId)
7     let subtotal = 0;
8
9     for(const item of order.cartItems) {
10       const price = yield getPriceOfProduct(item.productId)
11       subtotal += price * item.quantity
12     }
13
14     return subtotal
15   })
16 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Filter (e.g. text, **/*.ts, !**/node_modules/**)

1 / 1

◀ TS sample-01.ts examples 1

Argument of type '() => Generator<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>, number, ... ts(2345) [Ln 5, Col 21] ^
Call signature return types 'Generator<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>, number, any>' and 'Generator<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>, number, any>'
The types returned by 'next(...)' are incompatible between these types.
Type 'IteratorResult<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>, number>' is not assignable to type 'IteratorResult<...>'
Type 'IteratorYieldResult<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>>' is not assignable to type 'IteratorResult<...>'
Type 'IteratorYieldResult<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>>' is not assignable to type 'IteratorYieldResult<...>'
Type 'YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFoundError, never>' is not assignable to type 'YieldWrap<Effect<any, any, any>>'
Type 'Effect<number, ProductNotFoundError, never>' is missing the following properties from type 'YieldWrap<Effect<any, any, any>>': #private, [YieldWra...

tsconfig.json U X

examples > tsconfig.json > {} compilerOptions > moduleResolution

```
1  {
2    "compilerOptions": {
3      "strict": true,
4      "downlevelIteration": true,
5      "target": "es2015",
6      "module": "NodeNext",
7      "moduleResolution": "nodenext"
8    },
9    "include": ["**/*.ts"],
10 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

...

Filter

Tasks

▼

TS sample-01.ts 1, U X

▷ ⌂ ⏪ ⏴ ...

```
examples > ts sample-01.ts > ⌂ getOrderBySubtotal
1 import { Effect } from "effect"
2 import { getOrderById, getPriceOfProduct } from "./order"
3
4 export function getOrderBySubtotal(orderId: string) {
5   return Effect.gen(function* () { Argument of type '() => Generator<YieldWrap<Effect<Order, never, never>>' | Effect<number, ProductNotFound>
6     const order = yield* getOrderById(orderId)
7     let subtotal = 0;
8
9     for(const item of order.cartItems) {
10       const price = yield* getPriceOfProduct(item.productId)
11       subtotal += price * item.quantity
12     }
13
14     return subtotal
15   })
16 }
```

PROB

OUTPUT

DEBUG CONSOLE

Filter

Tasks



examples > TS sample-02.ts > ...

```
1 import { Effect } from "effect";
2 import {
3   getOrderById,
4   getPriceOfProduct,
5   OrderNotFoundError,
6   ProductNotFoundError,
7 } from "./order";
8
9 export const getOrderSubtotal = (
10   orderId: string
11 ): Effect.Effect<number, OrderNotFoundError | ProductNotFoundError> =>
12   Effect.gen(function* () {
13     const order = yield* getOrderById(orderId);
14     let subtotal = 0;
15
16     for (const item of order.cartItems) {
17       const price = yield* getPriceOfProduct(item.productId);
18       subtotal += price * item.quantity;
19     }
20
21     return subtotal;
22   });
23
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filter (e.g. text, **/*.ts, !**/node_modules/**)

□ ◻ ≡ ^ ×

- ✗ TS sample-02.ts examples 1

✖ Type 'Effect<number, OrderNotFoundError | ProductNotFoundError | PricingNotFoundError, never>' is not assignable to type 'Effect<nu... ts(2322) [Ln 12, Col 3] ^
Type 'OrderNotFoundError | ProductNotFoundError | PricingNotFoundError' is not assignable to type 'OrderNotFoundError | ProductNotFoundError'.
Type 'PricingNotFoundError' is not assignable to type 'OrderNotFoundError | ProductNotFoundError'.
Property 'order' is missing in type 'PricingNotFoundError' but required in type 'ProductNotFoundError'.
orders.ts [Ln 16, Col 2]: Labeled is declared here.

TS sample-02.ts 2, U X

▷ ⌂ ⌂ ...

```
examples > TS sample-02.ts > [e] getOrderSubtotal > ⓘ Effect.gen() callback
1 import { Effect } from "effect";
2 import {
3   getOrderId,
4   getPriceOfProduct,
5   OrderNotFoundError,
6   ProductNotFoundError,
7 } from "./order";
8
9 export const getOrderSubtotal = (
10   orderId: string
11 ): Effect
```

PROBLEMS 2

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Filter (e.g. text, **/*.ts, !**/node_modules/**)

Y ☰ ≡ ^ ×

▼ TS sample-02.ts examples 2

✖ Missing 'PricingNotFoundError' in the expected Effect errors. effect(2) [Ln 12, Col 3]

✖ Type 'Effect<number, OrderNotFoundError | ProductNotFoundError | PricingNotFoundError, never>' is not assignable to type 'Effect<nu... ts(2322) [Ln 12, Col 3] ^

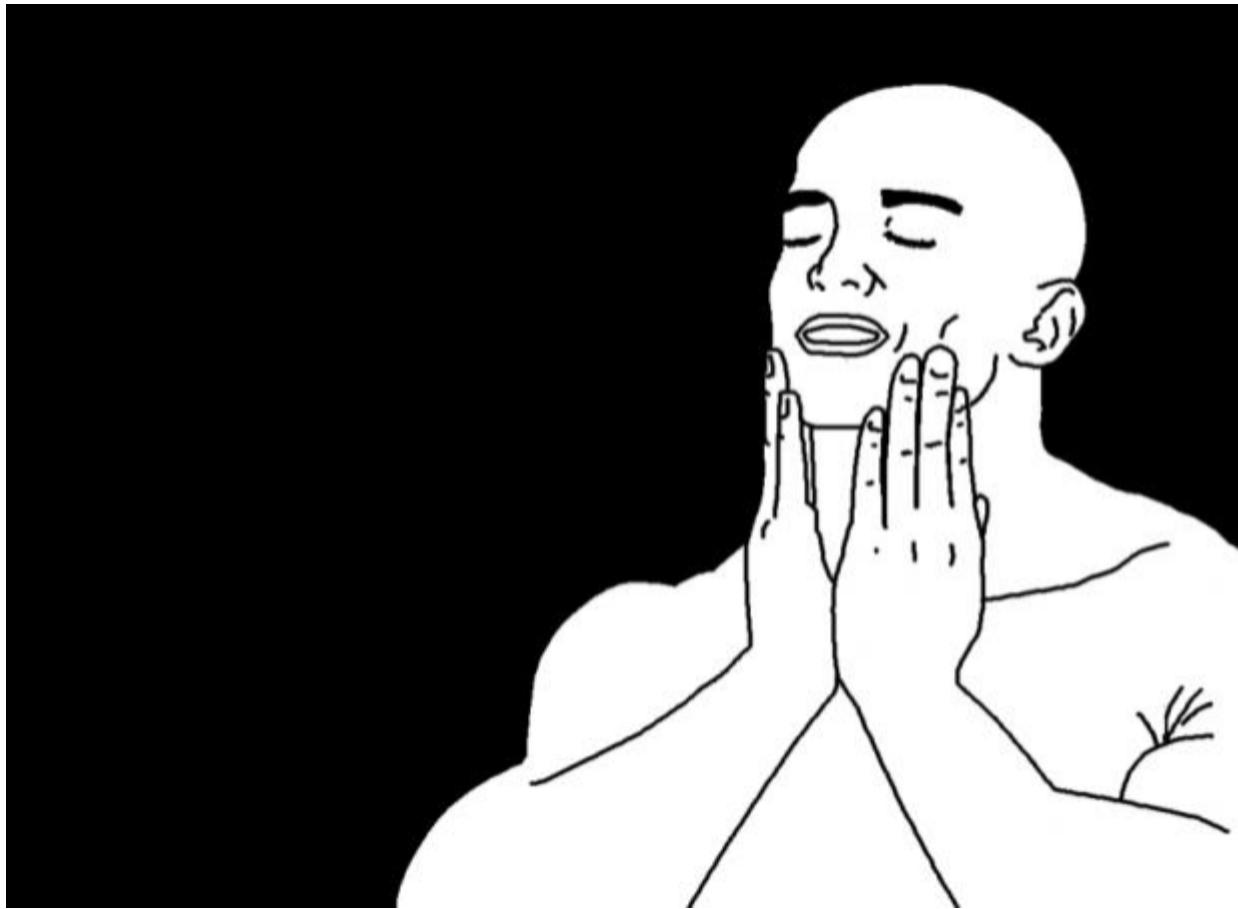
Type 'OrderNotFoundError | ProductNotFoundError | PricingNotFoundError' is not assignable to type 'OrderNotFoundError | ProductNotFoundError'.

Type 'PricingNotFoundError' is not assignable to type 'OrderNotFoundError | ProductNotFoundError'.

Property 'orderId' is missing in type 'IPricingNotFoundError' but required in type 'IPricingNotFoundError'

```
const longErrorType: Effect<.Effect<void, ErrorWithVeryLongName1 | ErrorWithVeryLongName2 | ErrorWithVeryLongName3 | ErrorWithVeryLongName4 | ... 5 more ... | ErrorWithVeryLongName10, never>

/* Effect Type Parameters */
type Success = void
type Failure = ErrorWithVeryLongName1 | ErrorWithVeryLongName2 | ErrorWithVeryLongName3 | ErrorWithVeryLongName4 | ErrorWithVeryLongName5 | ErrorWithVeryLongName6 | ErrorWithVeryLongName7 | ErrorWithVeryLongName8 | ErrorWithVeryLongName9 | ErrorWithVeryLongName10
type Requirements = never
```



TS sample-03.ts U X



```
examples > TS sample-03.ts > ⚡ getOrderSubtotal
1  import { Effect } from "effect";
2  import { Order } from "./order";
3
4  declare function getOrderById(orderId: string): Promise<Order>;
5  declare function getPriceOfProduct(productId: string): Promise<number>;
6
7  export async function getOrderSubtotal(orderId: string) {
8    const order = await getOrderById(orderId);
9    let subtotal = 0;
10
11   for (const item of order.cartItems) {
12     const price = await getPriceOfProduct(item.productId);
13     subtotal += price * item.quantity;
14   }
15
16   return subtotal;
17 }
18
```

PROBLEMS **OUTPUT** **DEBUG CONSOLE** **TERMINAL** **PORTS**

Filter (e.g. text, **/*.ts, !**/node_modules/**)



No problems have been detected in the workspace.

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filter (e.g. text, **/*.ts, !**/node_modules/**)

▽ ◀ ≡ ^ ×

✓ TS sample-floating.ts examples 1

 Effect must be yielded or assigned to a variable. effect(3) [Ln 4, Col 3]

@effect/eslint-plugin (future)

- Allows to prevent mistakes like
- Will provide an in-editor way to teach Effect best practices
- ...we'll also look into other linters as well!

DevTools for interactive learning experiences

The screenshot shows the Visual Studio Code interface with the Extensions sidebar open. The sidebar lists several extensions, each with a small icon, name, description, developer name, and a status indicator.

- EXTENSIONS**
- INSTALLED** (10 items)

 - Bun for Visual Studio Code** by The Visual Studio Code exten... (99ms)
 - Oven**
 - Error Lens** by Alexander (7ms)
Improve highlighting of error...
 - esbuild Problem Matchers** by Connor Peet
Problem matchers for esbuil...
 - ESLint** by Microsoft (22ms)
Integrates ESLint JavaScript...
 - Extension Test Runner** by Microsoft
Runs tests in VS Code exten...

- RECOMMENDED** (8 items)

 - Docker** by Microsoft (42.9M, 4.5 stars)
Makes it easy to create, man...
 - Dev Container** by Microsoft (29.8M, 4.5 stars)
Open any folder or repositor...

TS sample-04.ts

```
examples > TS sample-04.ts > ...
1 import { NodeRuntime, NodeSocket } from "@effect/platform-node";
2 import { Effect, Metric, Layer } from "effect";
3 import { DevTools } from "@effect/experimental"
4
5 const hiCounterMetric = Metric.counter("hi_counter");
6
7 const program = Effect.gen(function* () {
8     yield* Effect.log("Hello!");
9     yield* Metric.increment(hiCounterMetric);
10    yield* Effect.sleep(2000);
11 }).pipe(Effect.withSpan("Hi", { attributes: { foo: "bar" } }), Effect.forever);
12
13 const DevToolsLive = DevTools.layerWebSocket().pipe(
14     Layer.provide(NodeSocket.layerWebSocketConstructor),
15 )
16
17 program.pipe(Effect.provide(DevToolsLive), NodeRuntime.runMain);
18
```

TS sample-04.ts M ●

▷ ⌂ ⌂ ...

```
examples > TS sample-04.ts > ...
1 import { NodeRuntime } from "@effect/platform-node";
2 import { Effect, Metric, Layer } from "effect";
3
4 const hiCounterMetric = Metric.counter("hi_counter");
5
6 const program = Effect.gen(function* () {
7   yield* Effect.log("Hello!");
8   yield* Metric.increment(hiCounterMetric);
9   yield* Effect.sleep(2000);
10 }).pipe(Effect.withSpan("Hi", { attributes: { foo: "bar" } }), Effect.forever);
11
12 program.pipe(NodeRuntime.runMain);
13
```

EFFECT DEV TOOLS

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

...

Filter (e.g. text, **/*.ts, !**/node_modules/**)

✖️ ⌂ ⌂ ^ ×

No problems have been detected in the workspace.

The screenshot shows the Effect Dev Tools extension in VS Code. The sidebar on the left contains icons for EFFECT DEV TOOLS, CLIENTS (with a message about the server being stopped and a 'Start the server' button), TRACER, and METRICS. The main area has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is active and shows a terminal session), and PORTS. The terminal shows the command `bun sample-04.ts` being run.

EFFECT DEV TOOLS ...

CLIENTS

The Effect Dev Tools server is currently stopped.

Start the server

TRACER

METRICS

examples > TS sample-04.ts > ...

```
1 import { NodeRuntime, NodeSocket } from "@effect/platform-node";
2 import { Effect, Metric, Layer } from "effect";
3 import { DevTools } from "@effect/experimental"
4
5 const hiCounterMetric = Metric.counter("hi_counter");
6
7 const program = Effect.gen(function* () {
8     yield* Effect.log("Hello!");
9     yield* Metric.increment(hiCounterMetric);
10    yield* Effect.sleep(2000);
11 }).pipe(Effect.withSpan("Hi", { attributes: { foo: "bar" } }), Effect.forever);
12
13 const DevToolsLive = DevTools.layerWebSocket().pipe(
14     Layer.provide(NodeSocket.layerWebSocketConstructor),
15 )
16
17 program.pipe(Effect.provide(DevToolsLive), NodeRuntime.runMain);
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

mattiamanzati@MacBook-Pro-di-Mattia-2 examples % bun sample-04.ts

RUN AND DEBUG ...

examples > TS sample-05.ts [U] M JS core.js

VARIABLES

WATCH

CALL STACK

BREAKPOINTS

- All Exceptions
- Uncaught Exceptions
- Debugger Statements
- Assertion Failures
- Microtasks

```
3  getOrderById,
4  getPriceOfProduct,
5  OrderNotFoundError,
6  PricingNotFoundError,
7  ProductNotFoundError,
8 } from "./order";
9
10 export const getOrderBySubtotal = (
11   orderId: string
12 ): Effect<number, OrderNotFoundError | ProductNotFoundError | PricingNotFoundError> =>
13   Effect.gen(function* () {
14     const order = yield* getOrderBySubtotal(orderId);
15     let subtotal = 0;
16
17     for (const item of order.cartItems) {
18       const price = yield* getPriceOfProduct(item.productId);
19       subtotal += price * item.quantity;
20     }
21
22     return subtotal;
23   }):
```

EFFECT DEV TOOLS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

mattiamanzati@MacBook-Pro-di-Mattia-2 examples % bun sample-05.ts

zsh

JavaScript Debug Ter...

DevTools for every environment (future)

- Chrome extension
- VSCode extension
- Standalone application

```
▶ livestore-worker-default-dj5kl paus
eventlog-worker
▶ Watch
▶ Breakpoints
as
  □ Pause on uncaught exceptions
yf
  □ Pause on caught exceptions
  ▼ Scope
    ▼ Local
      this: undefined
      ▶ args: []
        msg: "Expected to be applying-syncstate-advance but got in-sync"
    ▶ Module
    ▶ Global
    ▶ Call Stack
      □ Show ignore-listed frames
      ▶ shouldNeverHappen index.ts:94
        (anonymous)
        @livestore/common:leader-thread:syncing:applyMutationItems leader-sync-processor.ts:324
        (anonymous)
        @livestore/common:leader-thread:syncing:local-push make-leader-worker.ts:77
        (anonymous)
        @livestore/web:worker:PushToLeader make-leader-worker.ts:77
        (anonymous)
        @livestore/web:worker:wrapper:InitialMessage:innerFiber make-leader-worker.ts:77
        (anonymous)
        @livestore/web:worker:wrapper:InitialMessage make-leader-worker.ts:77
        (anonymous)
        Promise.then make-leader-worker.ts:77
        (anonymous)
      ▶ XHR/fetch Breakpoints
      ▶ DOM Breakpoints
      ▶ Global Listeners
      ▶ Event Listener Breakpoints
      ▶ CSP Violation Breakpoints
DedicatedWorkerGlobalScope
```

Console AI assistance Performance monitor What's new Memory inspector Network conditions Issues Rendering Search Lighthouse Network request blocking

livestore-wo... Filter All levels No Issues

15:29:48.822 [15:29:48.822] ERROR (#449): Error on main-thread: LiveStore.UnexpectedError: { "cause": TimeoutException: Operation timed out before the specified duration of '10s' elapsed, "note": undefined, "payload": undefined } logger.ts:457
15:29:48.822 [15:29:48.822] DEBUG (#449): Fiber terminated with an unhandled error logger.ts:457
15:29:48.953 [15:29:48.953] ERROR (#472): Error on main-thread: LiveStore.UnexpectedError: { "cause": TimeoutException: Operation timed out before the specified duration of '10s' elapsed, "note": undefined, "payload": undefined } logger.ts:457
15:29:48.955 [15:29:48.954] DEBUG (#472): Fiber terminated with an unhandled error logger.ts:457
15:29:48.956 [15:29:48.956] ERROR (#477): Error on main-thread: LiveStore.UnexpectedError: { "cause": TimeoutException: Operation timed out before the specified duration of '10s' elapsed, "note": undefined, "payload": undefined } logger.ts:457
15:29:48.957 [15:29:48.957] DEBUG (#477): Fiber terminated with an unhandled error logger.ts:457
15:29:48.959 [15:29:48.959] ERROR (#482): Error on main-thread: LiveStore.UnexpectedError: { "cause": TimeoutException: Operation timed out before the specified duration of '10s' elapsed, "note": undefined, "payload": undefined } logger.ts:457
15:29:48.960 [15:29:48.960] DEBUG (#482): Fiber terminated with an unhandled error logger.ts:457

Observability

What to say...it's great!

What's next?

Let's shape it together!

Thank you!

x.com/mattiamanzati



