wrapTime

August 17, 2026

wrapTime

Kind: Function

Source: `src/middleware/timing/timing.ts`

Part of: Middleware

Wrap a Promise to capture its duration.

`wrapTime` wraps a Promise and records how long the asynchronous operation takes to settle. It is used by timing middleware to collect duration data without changing the operation's normal resolution or rejection flow.

Signature

```
async function wrapTime(c: Context, name: string, callable: Promise<T>, description: string, pr
```

Parameters

Name	Type
`c`	`Context`
`name`	`string`
`callable`	`Promise<T>`
`description`	`string`
`precision`	`number`

Returns: `Promise<T>`

Diagram

```mermaid
graph LR
  A[Async operation Promise] --> B[wrapTime]
  B --> C[Record start time]
  C --> D[Wait for Promise settlement]
  D --> E[Capture duration]
  E --> F[Timing middleware data]
```

Usage

```ts
import { wrapTime } from "./middleware/timing/timing";

async function loadUser() {
  const response = await wrapTime(fetch("/api/users/current"));

  if (!response.ok) {
    throw new Error("Unable to load user");
  }

  return response.json();
}
```

AI Coding Instructions

- Pass the original Promise to `wrapTime`; do not await the operation before wrapping it.
- Await or return the wrapped Promise so callers keep the original async control flow.
- Apply `wrapTime` at middleware or request boundaries where duration data is collected.
- Preserve existing error handling; rejected Promises should continue through the normal error path.