

# startTime

August 19, 2026

## startTime

**Kind:** Function

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

**Part of:** [Middleware](#)

Start a timer for the timing middleware.

`startTime` starts the timer used by the timing middleware. Call it when request handling begins, then retain the returned value so later middleware code can calculate elapsed time.

## Signature

```
function startTime(c: Context, name: string, description: string)
```

## Parameters

| Name                     | Type                 |
|--------------------------|----------------------|
| <code>c</code>           | <code>Context</code> |
| <code>name</code>        | <code>string</code>  |
| <code>description</code> | <code>string</code>  |

## Diagram

```
graph LR
  Request[Incoming request] --> Start[startTime]
  Start --> Timer[Stored timer value]
  Timer --> Handler[Request handler]
  Handler --> End[Timing middleware records elapsed time]
```

## Usage

```
import { startTime } from "../src/middleware/timing/timing";

function timingMiddleware(request: Request) {
  const timer = startTime();

  return handleRequest(request).finally(() => {
    recordTiming(timer);
  });
}
```

## AI Coding Instructions

- Call `startTime` at the beginning of the middleware execution path.
- Keep the returned timer value available until elapsed time is recorded.
- Use the same timer value for all timing calculations associated with a request.
- Keep timer creation separate from response logging or metric submission.