HonoRequest

August 18, 2026

HonoRequest

Kind: Class

Source: `src/request.ts`

`HonoRequest` represents the incoming request exposed through Hono's context. It reads decoded route parameters with `param()` and URL query values with `query()`, while keeping request parsing behavior inside the framework.

Methods

Method	Signature	Returns
`param`	`param(key: string extends P ? never : P2 extends ${infer _}? ? never : P2)`	`string`
`param`	`param(key: P2)`	`string
`param`	`param(key: string)`	`string
`param`	`param()`	`Simplify<UnionToIntersection<ParamKeyToRecord<ParamKeys<P2>>>>`
`param`	`param(key: string)`	`unknown`
`#getDecodedParam`	`#getDecodedParam(key: string)`	`string
`#getAllDecodedParams`	`#getAllDecodedParams()`	`Record<string, string>`
`#getParamValue`	`#getParamValue(paramKey: any)`	`string
`query`	`query(key: string)`	`string
`query`	`query()`	`Record<string, string>`
`query`	`query(key: string)`	`void`
`queries`	`queries(key: string)`	`string[]
`queries`	`queries()`	`Record<string, string[]>`
`queries`	`queries(key: string)`	`void`
`header`	`header(name: RequestHeader)`	`string
`header`	`header(name: string)`	`string
`header`	`header()`	`Record<RequestHeader

Method	Signature	Returns
`header`	`header(name: string)`	`void`
`parseBody`	`parseBody(options: Options)`	`Promise<T>`
`parseBody`	`parseBody(options: Partial<ParseBodyOptions>)`	`Promise<T>`
`parseBody`	`parseBody(options: Partial<ParseBodyOptions>)`	`void`
`json`	`json()`	`Promise<T>`
`text`	`text()`	`Promise<string>`
`arrayBuffer`	`arrayBuffer()`	`Promise<ArrayBuffer>`
`bytes`	`bytes()`	`Promise<Uint8Array>`
`blob`	`blob()`	`Promise<Blob>`
`formData`	`formData()`	`Promise<FormData>`
`addValidatedData`	`addValidatedData(target: keyof ValidationTargets, data: {})`	`void`
`valid`	`valid(target: T)`	`InputToDataByTarget<I, T>`
`valid`	`valid(target: keyof ValidationTargets)`	`void`

Properties

Property	Type
`raw`	`Request`
`#validatedData`	`{ [K in keyof ValidationTargets]?: {} }`
`#matchResult`	`Result<[unknown, RouterRoute]>`
`routeIndex`	`number`
`path`	`string`
`bodyCache`	`BodyCache`
`#cachedBody`	`any`

Where it refuses work

- `HonoRequest` stops the work with an early return when `name`.
- `HonoRequest` stops the work with an early return when `cachedBody`.

Diagram

```
graph LR
  Request[Incoming Request] --> HonoRequest
```

```
  Route[Matched Route] --> HonoRequest
  HonoRequest --> Param[param()]
  HonoRequest --> Query[query()]
  Param --> Handler[Route Handler]
  Query --> Handler
```

Usage

```
import { Hono } from 'hono'

const app = new Hono()

app.get('/users/:id', (c) => {
  const id = c.req.param('id')
  const filter = c.req.query('filter')
  const query = c.req.query()

  return c.json({
    id,
    filter,
    query,
  })
})

export default app
```

AI Coding Instructions

- Access `HonoRequest` through `c.req` in route handlers rather than constructing it directly.
- Use `param('name')` when reading a matched route parameter; account for `undefined` when the parameter may not exist.
- Use `param()` without an argument when the route's typed parameter record is needed.
- Use `query('name')` for a single query value and `query()` for the full query record.
- Keep parameter decoding in the request class; do not call private `#getDecodedParam`, `#getAllDecodedParams`, or `#getParamValue` outside this class.

Relationships

- IMPORTS → `HTTPException`
- IMPORTS → `GET_MATCH_RESULT`
- IMPORTS → `parseBody`
- IMPORTS → `getQueryParam`
- IMPORTS → `getQueryParams`
- IMPORTS → `tryDecodeURIComponent`

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `Data` — `src/context.ts`:26