



MIDDLEWARE

LanguageVariables

Kind: Interface

VERSION
5

UPDATED
12 August 2026

DOCUMENT
reference-subsystem-src-middleware/interface-languagevariables

CONTENTS

1	Properties	2	4	AI Coding Instructions	3
2	Diagram	2	5	How it works	3
3	Usage	2			

Kind: Interface

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

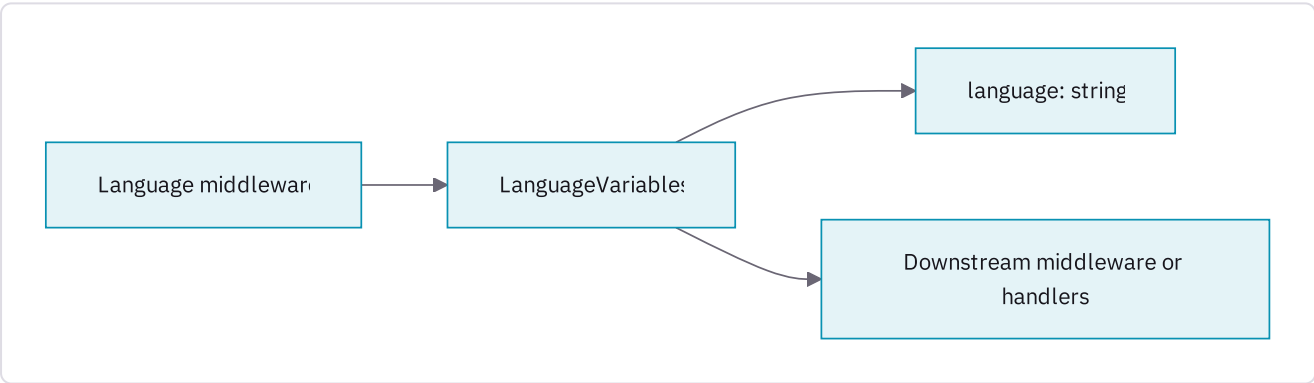
Part of: Middleware

`LanguageVariables` defines the language value used by language middleware. It carries a `language` string so downstream middleware or handlers can read the active language.

1 Properties

Property	Type
<code>language</code>	<code>string</code>

2 Diagram



3 Usage

```
import type { LanguageVariables } from './src/middleware/language/language';

const variables: LanguageVariables = {
  language: 'en',
};

function getLanguage({ language }: LanguageVariables): string {
  return language;
}

const activeLanguage = getLanguage(variables);
```

4 AI Coding Instructions

- Keep the `language` field as a string when creating `LanguageVariables` values.
- Pass `LanguageVariables` through middleware context or handler parameters where language state is needed.
- Do not rename the `language` property without updating middleware and downstream consumers.
- Validate or normalize language input before assigning it to this interface when input comes from requests.

5 How it works

`LanguageVariables` is an exported TypeScript interface with one context-variable entry:

- `language: string` — the selected language value is typed as a string.
[src/middleware/language/language.ts:47-49]

It is intended for the `Variables` type of a Hono application: the in-file example aliases

`LanguageVariables` and passes it as `Variables` when constructing `Hono`.

[src/middleware/language/language.ts:274-278] With that typing, `c.get('language')` refers to the

`language` variable. [src/middleware/language/language.ts:286-289] Context variable getters and setters map keys in `E['Variables']` to their declared value types. [src/context.ts:95-108]

`LanguageVariables` itself has no methods, validation, errors, or runtime side effects. The related

`languageDetector` middleware detects a language, stores it as `ctx.set('language', lang)`, then calls the next middleware. [src/middleware/language/language.ts:302-308] The stored value is either the first detected language from the configured detector order or `fallbackLanguage`.

[src/middleware/language/language.ts:238-266]