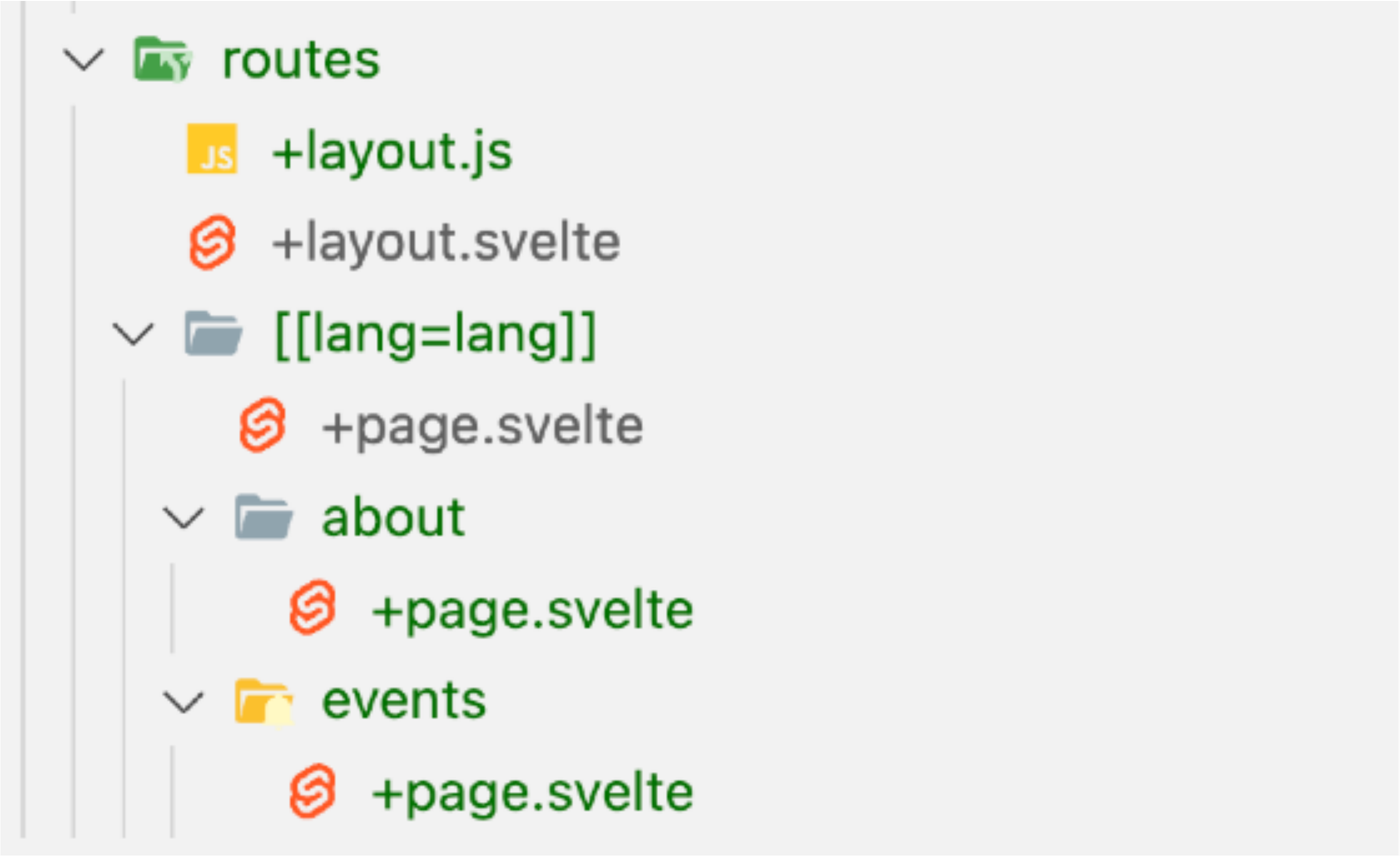
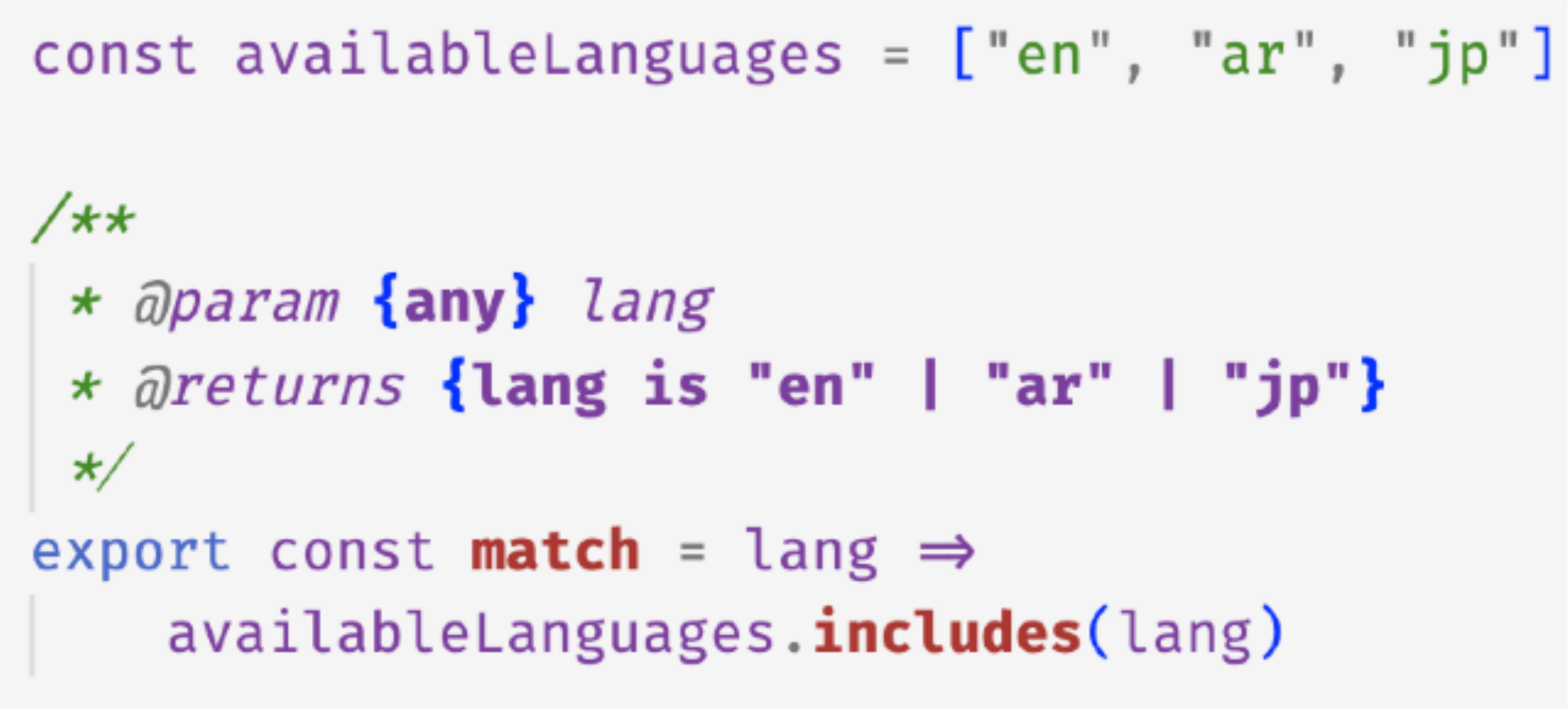
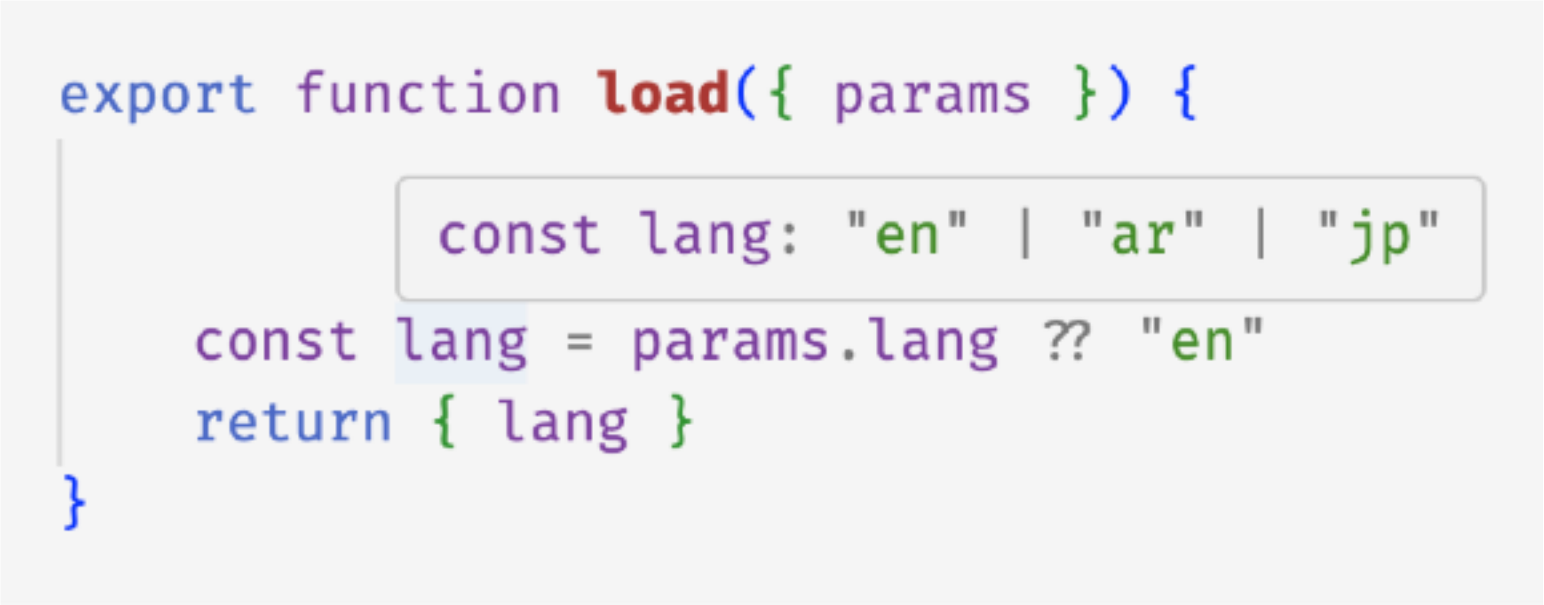


```
routes
  +layout.js
  +layout.svelte
  [[lang=lang]]
    +page.svelte
    about
      +page.svelte
    events
      +page.svelte
```



```
const availableLanguages = ["en", "ar", "jp"]

/**
 * @param {any} lang
 * @returns {lang is "en" | "ar" | "jp"}
 */
export const match = lang ⇒
    availableLanguages.includes(lang)
```



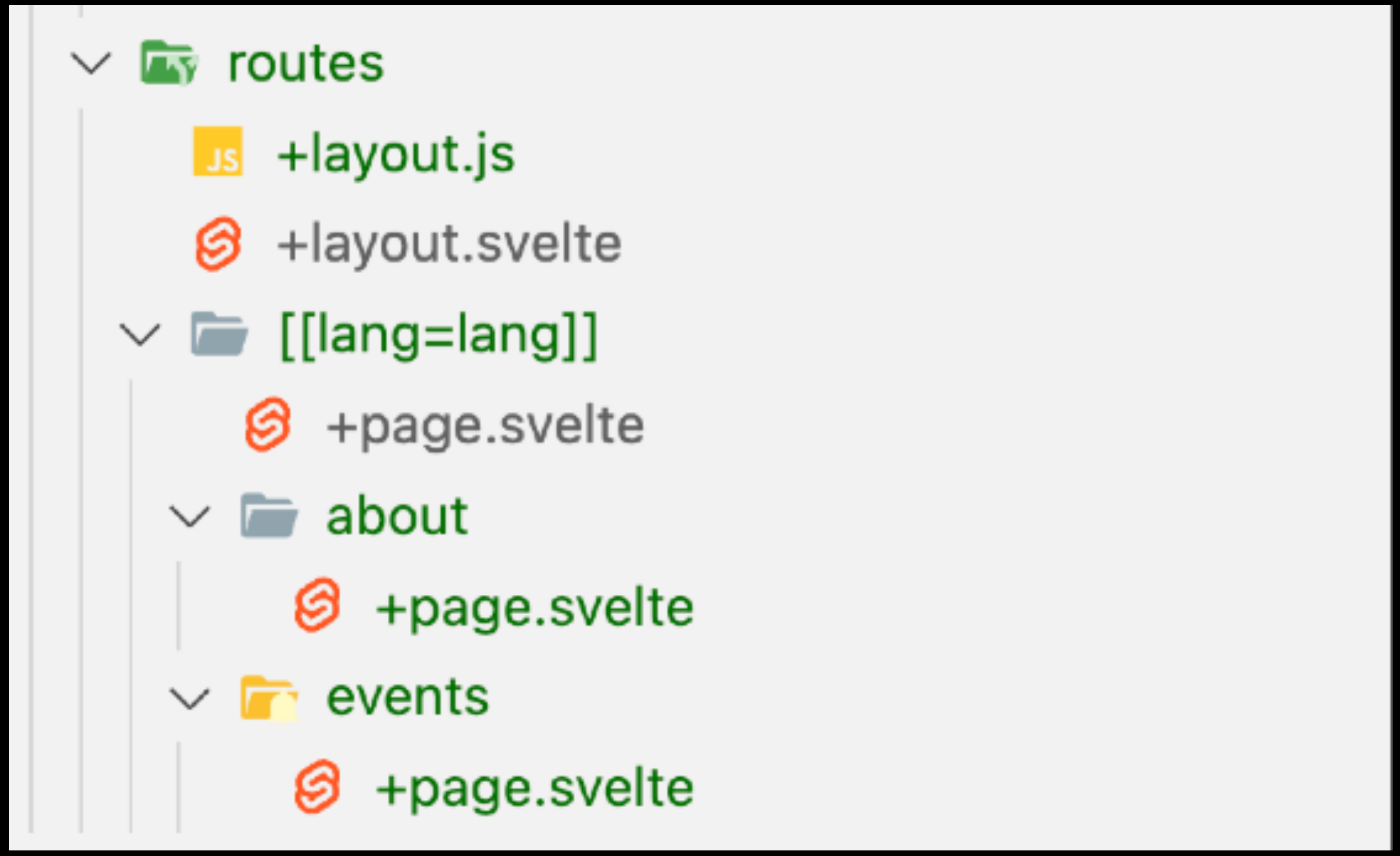
```
export function load({ params }) {
                 const lang: "en" | "ar" | "jp"
    const lang = params.lang ?? "en"
    return { lang }
}
```

Internationalised Routing in SvelteKit

Storing the language in the URL

/en/about	example.com/about	example.com/about
/de/about	example.de/about	de.example.com/about
/fr/about	example.fr/about	fr.example.com/about

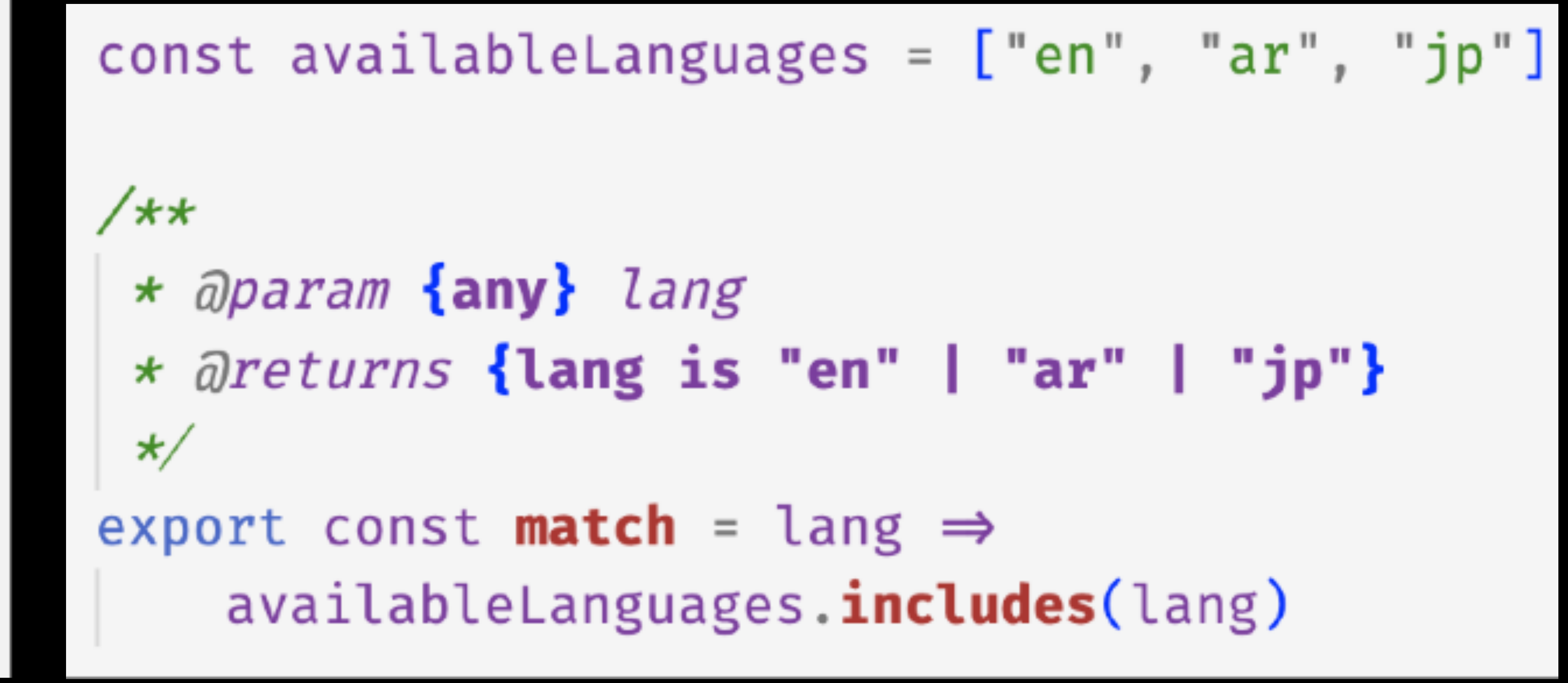
Add a language Parameter



```
routes
  +layout.js
  +layout.svelte
  [[lang=lang]]
    +page.svelte
    about
      +page.svelte
    events
      +page.svelte
```

Add a matcher

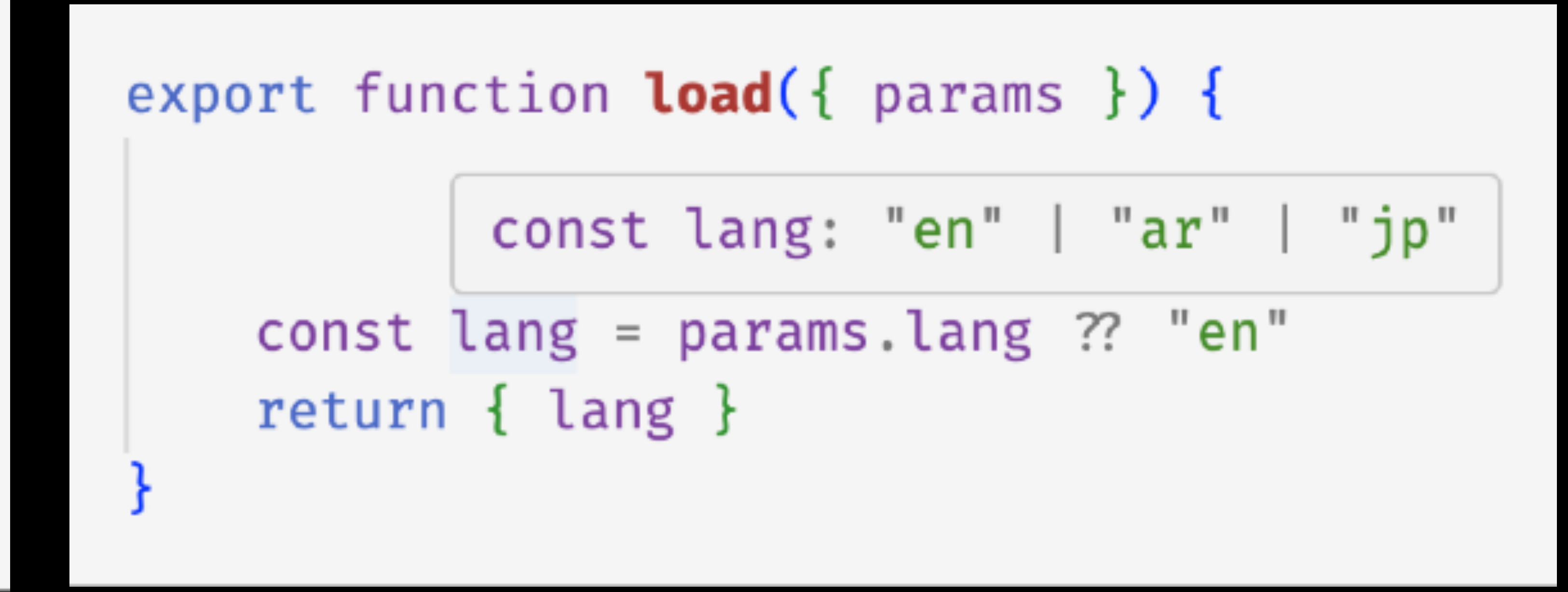
src/params/lang.js



```
const availableLanguages = ["en", "ar", "jp"]

/**
 * @param {any} lang
 * @returns {lang is "en" | "ar" | "jp"}
 */
export const match = lang =>
    availableLanguages.includes(lang)
```

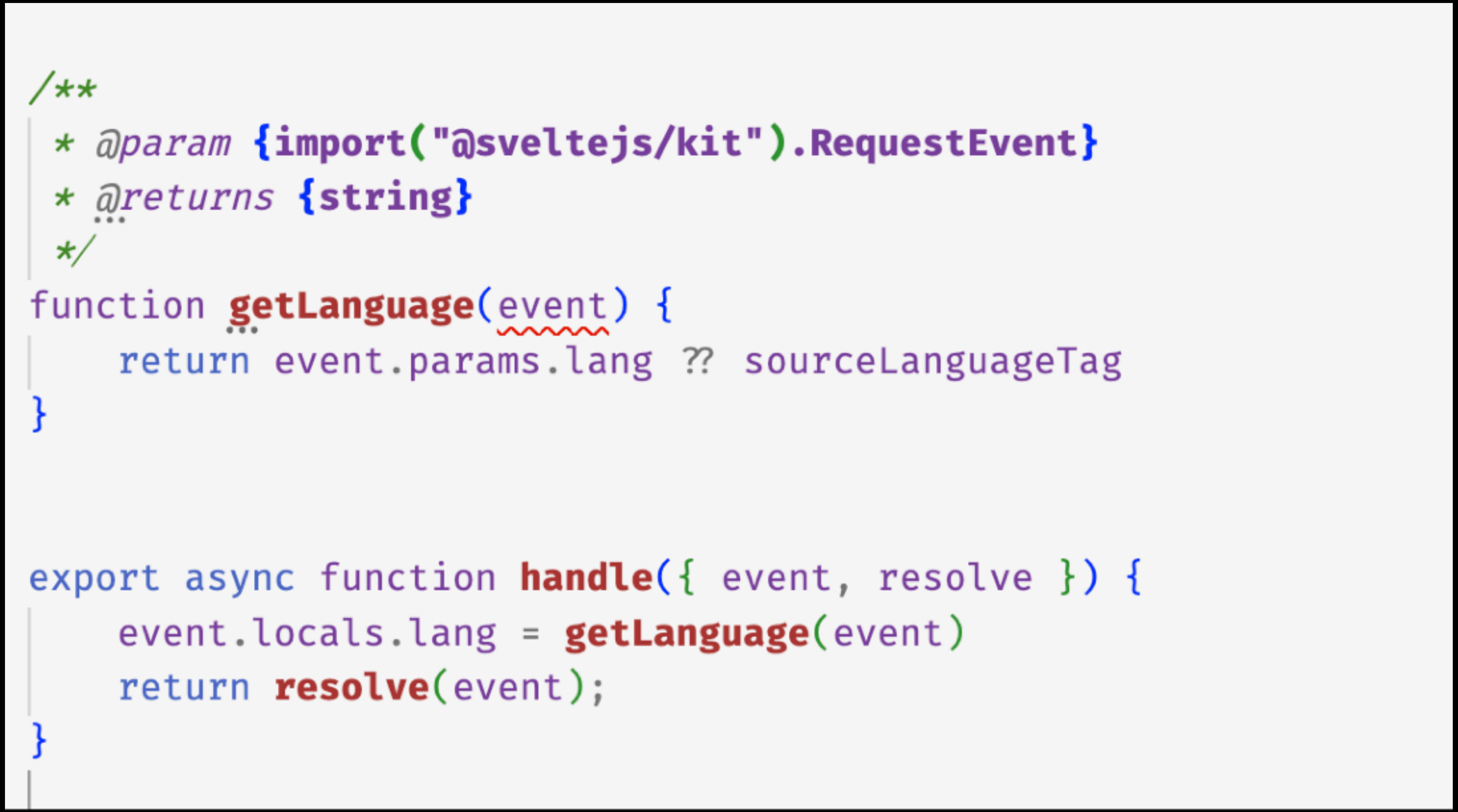
src/routes/+layout.js



```
export function load({ params }) {
                const lang: "en" | "ar" | "jp"
    const lang = params.lang ?? "en"
    return { lang }
}
```

Internationalised Routing in SvelteKit

Detecting the Language on the Server



```
/**
 * @param {import("@sveltejs/kit").RequestEvent}
 * @returns {string}
 */
function getLanguage(event) {
    return event.params.lang ?? sourceLanguageTag
}


export async function handle({ event, resolve }) {
    event.locals.lang = getLanguage(event)
    return resolve(event);
}
```