

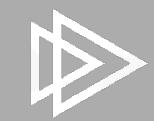
Customizing Locale Handling



Alisa Duncan

Developer | Google Developer Expert

linkedin.com/in/jalisaduncan/ | alisaduncan.dev



LOCALE_ID Injection Token



Provides locale information at runtime
Used by Angular pipes



Access the LOCALE_ID

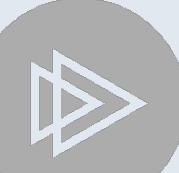
app.ts

```
import { LOCALE_ID } from '@angular/core';

@Component{
  selector: 'app-root',
  templateUrl: './app-root'
}
export class AppComponent {
  locale = inject(LOCALE_ID);
}
```

app.html

```
<p>My locale is {{locale}}</p>
```



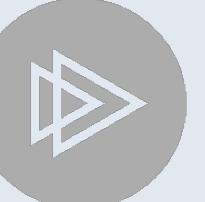
Form URLs Based on Locale

If your API endpoints take the form of `/api/<locale_id>/products`

`products.ts`

```
export class ProductsService {
  private locale = inject(LOCALE_ID);
  private URL = `/api/${this.locale}/products`;

  // GET, POST, PUT, PATCH, DELETE products
}
```



Set Locale

Provide `LOCALE_ID` in application initialization

Set locale to use for a pipe



Override Locale on Pipes

Set locale individually on pipes after importing global variants for the locale

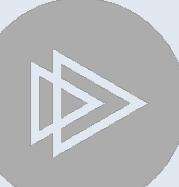
app.config.ts

```
import '@angular/common/locales/global/ja';

export const app: ApplicationConfig = {
  providers: [
    /* providers here */
  ]
};
```

app.ts

```
@Component{
  selector: 'app-root',
  template: `<p>Today is
    {{today | date:'full':undefined:'ja'}}
  </p>`
}
export class AppComponent {
  today = new Date();
}
```

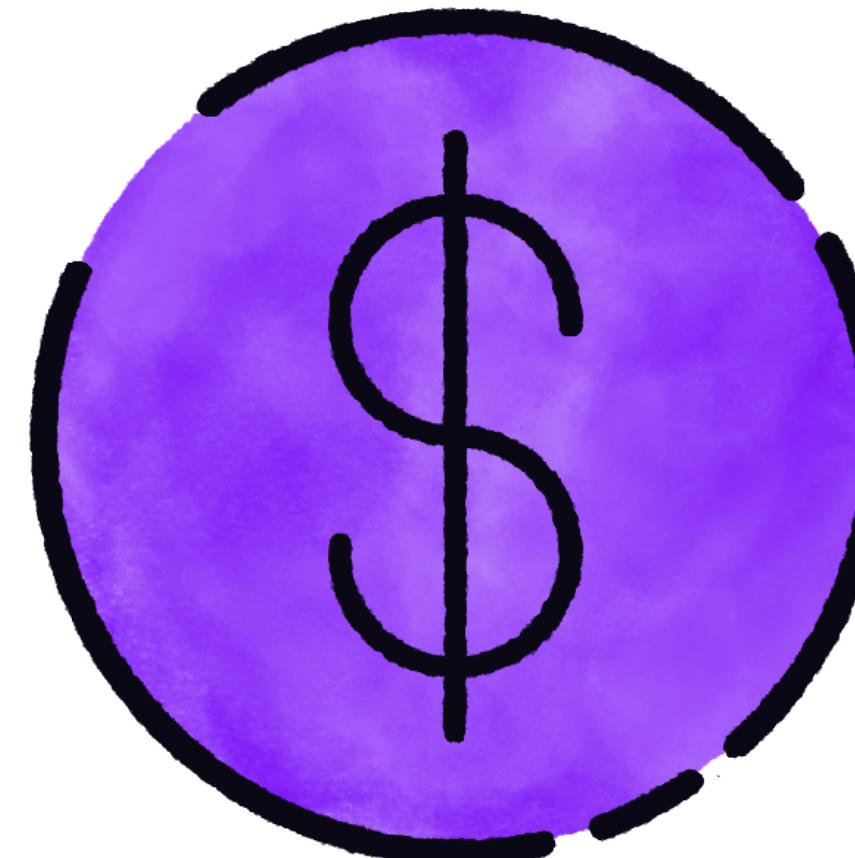


Locale and Currency



Locale

Language with
regional data format



Currency

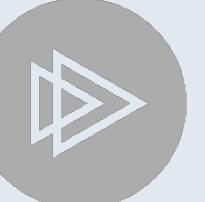
Monetary format



Define Currency

cart.ts

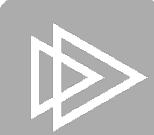
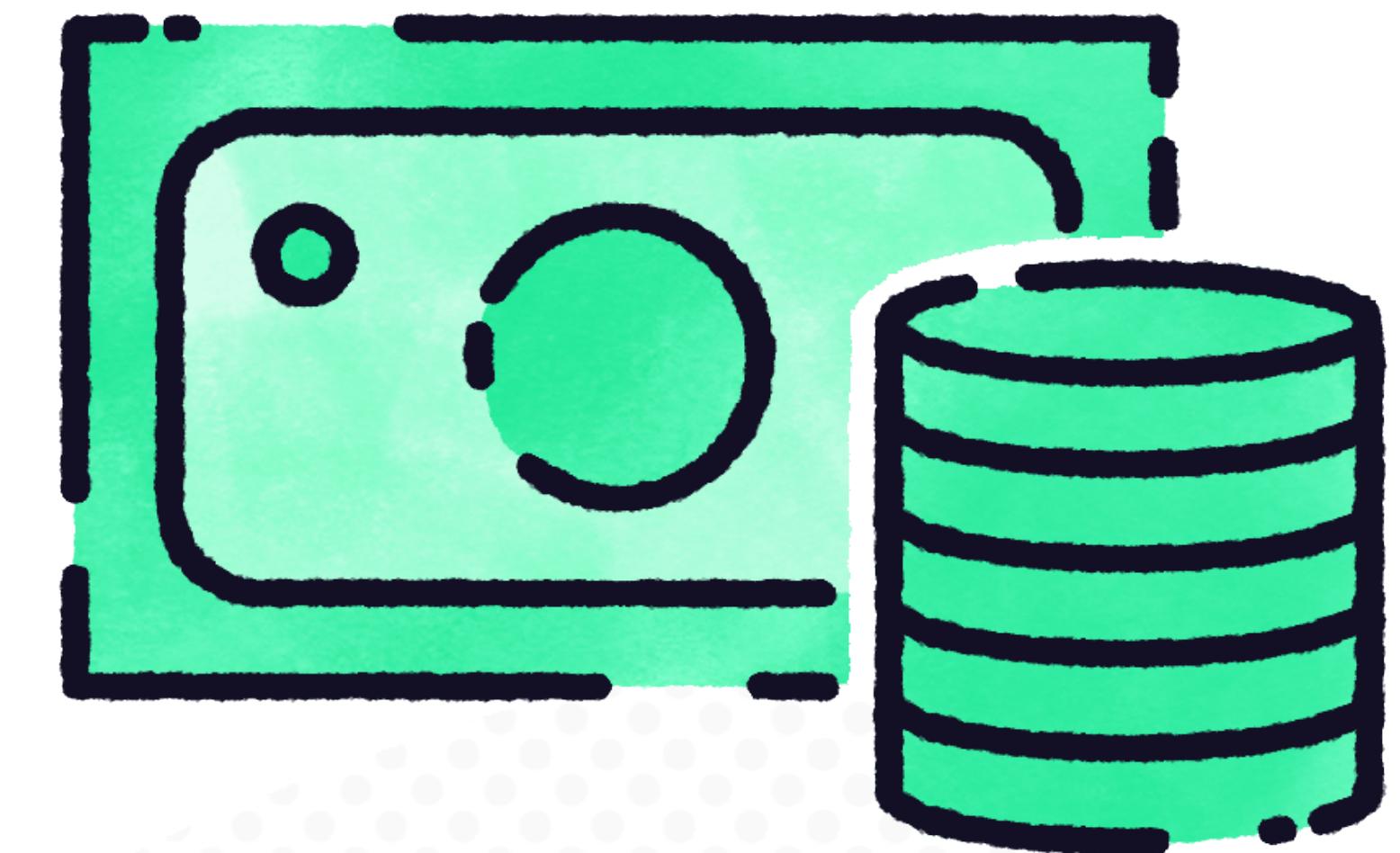
```
@Component( {  
    selector: 'app-cart',  
    template: `<p>{{ total | currency:'CAD' }}</p>`,  
}  
export class CartComponent {  
    total = 100;  
}
```



DEFAULT_CURRENCY_CODE Injection Token

Set currency globally

Provide the `DEFAULT_CURRENCY_CODE` injection token in application initialization



Add Support for Multiple Currencies

Use currency pipe and provide the DEFAULT_CURRENCY_CODE

cart.ts

```
@Component({
  selector: 'app-cart',
  template: `
    Your total is {{total | currency}}
  `
})
export class CartComponent {
  total = 2550;
}
```

App Initialization

```
{
  provide: DEFAULT_CURRENCY_CODE,
  useFactory: (locale: string) =>
    locale === 'ja' ? 'JPY': 'USD',
  deps: [LOCALE_ID]
}
```



Course Summary

Ensure your app supports global needs:

- UI layout and spacing
- Concise, clear language
- Format data using Angular pipes
- Logical styles

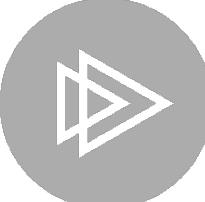
Use ICU expressions for complex translations

Ensure app maintainability and growth

Add currency support as required

Test with native speakers

Rely on a well-maintained localization library



Thank you for your time!

linkedin.com/in/jalisaduncan/ | alisaduncan.dev

