



Spartacus Training

Day 6

Introduction

- ❑ Navigation in a web application is mostly done via URLs
- ❑ Spartacus allows you to customize URLs:
- ❑ Using application url as state
- ❑ Good for SEO
- ❑ Its developed over angular routing

Introduction

☐ SEO

- Enhance product searchability

☐ Stateful URLs

- Maintain the application state

☐ Pretty URL

- Human readable

☐ Configurable URLs

- You can configure almost anything



SEO

- **SEO**
 - Products are intuitively searchable
- Use **keywords**
 - E.g., [/digital-camera/compact/](#)
- Use the **hyphen** as a word separator
 - E.g., [/digital-camera/](#) instead of [/digital%20cameras/](#)
- Keep **URLs short**
 - E.g., [/p/1981415/pl60-silver](#) instead of [/products/awesome/cool/digital-cameras/samsung/1981415/pl60-silver](#)
- Use only **lowercase** characters
 - E.g. [/digital-camera/](#) instead of [/Digital-Camera/](#)



Customizations

- You can configure your application by defining **context properties** such as *baseSite*, *language*, and *currency*.
 - These properties are arrays that take their first element as the default value.
 - Context properties also set the default values for the language and currency drop-down lists, which you can use to change the context of the storefront dynamically.
 - By default, context does not appear in the Spartacus storefront URL.
 - When you append the values of these properties to the storefront URL, the storefront is configured based on these values.
- The `urlParameters` property takes the values of the other `context` properties to create the structure of the context that is appended to the storefront URL.

```
B2cStorefrontModule.withConfig({
  context: {
    baseSite: ['electronics-spa'],
    language: ['en'],
    currency: ['USD'],
    urlParameters: ['baseSite', 'language', 'currency']
  },
  ...
```

`app.module.ts`



Routing in PDP

- *Product* or *Category Page* routes can be configured in Spartacus. They have to contain the `:productCode` or `:categoryCode` param to identify the product or category.
- To leverage SEO, you may want to **include more parameters** that can be accessed in the product entity, e.g., *product name*.

```
ConfigModule.withConfig({
  routing: {
    routes: {
      product: {
        paths: [
          'product/:manufacturer/:category/:productCode/:name',
          'product/:manufacturer/:productCode/:name',
          'product/:productCode'
        ],
        paramsMapping: [ productCode: 'code'],
      }
    }
  }
} as RouteConfig)
```

Sample Config

- The optional `paramsMapping` config can be used for properties that have a different name than the route parameter (e.g., to map from *product.code* to *:productCode*).

11




Routing in PDP

- Multiple route aliases can be configured in the **paths** array.
- Spartacus uses **the first configured path alias** from the **paths** array **that can satisfy its params** with given params object.
 - It's good to order aliases from those that require the most specific parameters to those having fewer parameters (order of alias matters).

```
ConfigModule.withConfig({
  routing: {
    routes: {
      product: {
        paths: [
          ':campaignName/p/:productCode', /* this will be used when `campaignName` param is given */
          'p/:productCode' /* this will be used otherwise */
        ]
      }
    }
  }
})
```

Is campaignName param given?



Routing in PDP-PLP

- When campaignName param **is** given

```
<a [routerLink]="{ cxRoute: 'product', params: { productCode: 1234, campaignName: 'sale' } } | cxUrl"></a>
```



```
<a [routerLink]="['/', 'sale', 'p', '1234']"></a>
```

- When campaignName param **is NOT** given

```
<a [routerLink]="{ cxRoute: 'product', params: { productCode: 1234 } } | cxUrl"></a>
```



```
<a [routerLink]="['/', 'p', '1234']"></a>
```



Creating a custom Url

```
const routes: Routes = [  
  {  
    component: ComponentName,  
    path: 'pathname',  
    data: {pageLabel: 'somePagelabelname'},  
    canActivate: guardName  
  }  
];
```



Resources

- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/adding-and-customizing-routes>
- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/creating-pages-and-components/#static-pages>
- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/route-configuration/#angulars-routes>
- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/configurable-routing/>
- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/static-context-configuration/>
- <https://sap.github.io/cloud-commerce-spartacus-storefront-docs/configurable-router-links/>



Exercise

1. Create a new component called 'contact'
2. Add some contact details in the component's template
3. Configure your project so that the /contact route directs to this component

The image features a dark blue background with four decorative geometric patterns in the corners. The top-right corner has a teal diamond grid pattern. The bottom-left corner has a teal stepped line pattern. The bottom-right corner has a teal zigzag line pattern. The top-left corner is empty. In the center, the text "Thank you !" is written in a bold, orange-brown sans-serif font. Below the text is a short, horizontal orange-brown line.

Thank you !
