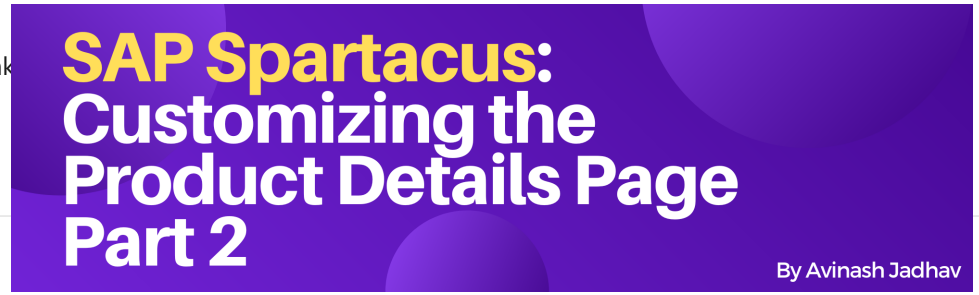


SAP Spartacus: Customizing the Product Details Page Part 2

Avinash Jadhav
SAP Composable Storefront(formerly Spartacus) Trainer / Coach / Expert |...

5 articles ✓ Following

November 5, 2022

Open Immersive Reader

Hey, welcome back! Today I will show you how to customize the Product Details Page by SAP Spartacus library.

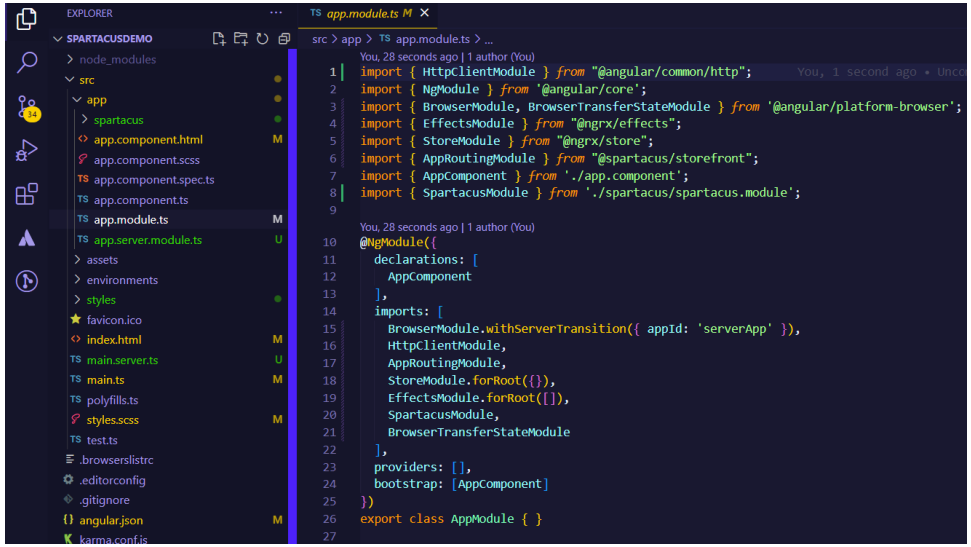
Note: Before moving into this article, please read **first part** of this article series on product details page customization .

In the previous article we were learning how to identify CMS component which we need to customize and it was Product Details Page(PDP). In this article, I'm going to show you actual implementation with some small configurations.

Previous article was finished with identifying CMS component which we are going to customize inside PDP page. Here is the last snippet of the previous article:

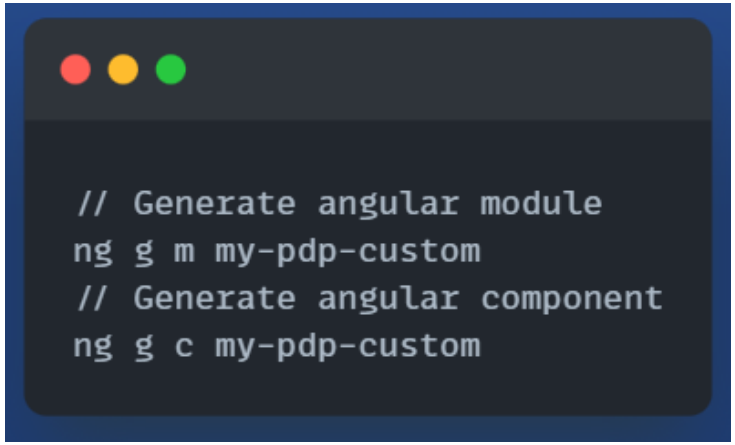
The CMS component which we are going to map is **ProductIntroComponent** from **ProductSummarySlot.** From above screenshot we have to use property called **flexType** from ProductIntroComponent which is to be mapped with our custom angular component. We are going to use this **flexType** value in our Spartacus configuration.

Let me show the the current structure of my Spartacus Demo project.

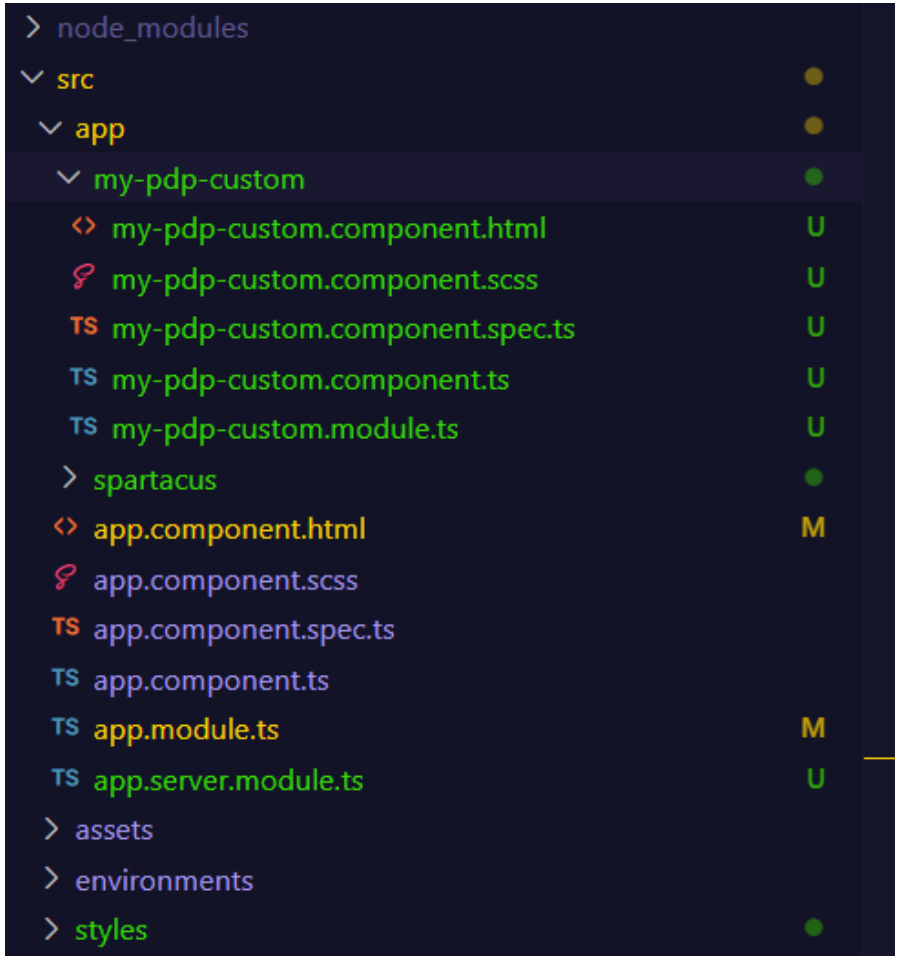


Now let's create our angular component and module which will be mapped with **ProductIntroComponent**.

Please use below commands to create angular component and module.



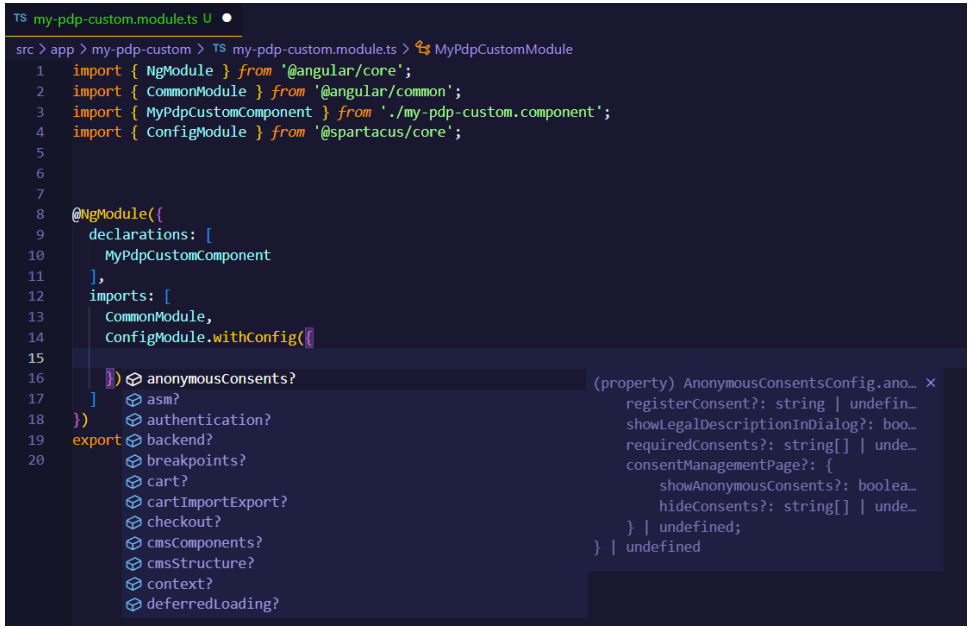
With above commands my project structure will look something like this.



Now lets dive into actual configuration part of this article. The file we are going to add our spartacus configuration are my-pdp-custom.module.ts

For adding spartacus configuration we have to use module called **ConfigModule** from **spartacus core** libraries. This

ConfigModule ships with method called **withConfig()**. In this method we have to pass our configuartion.

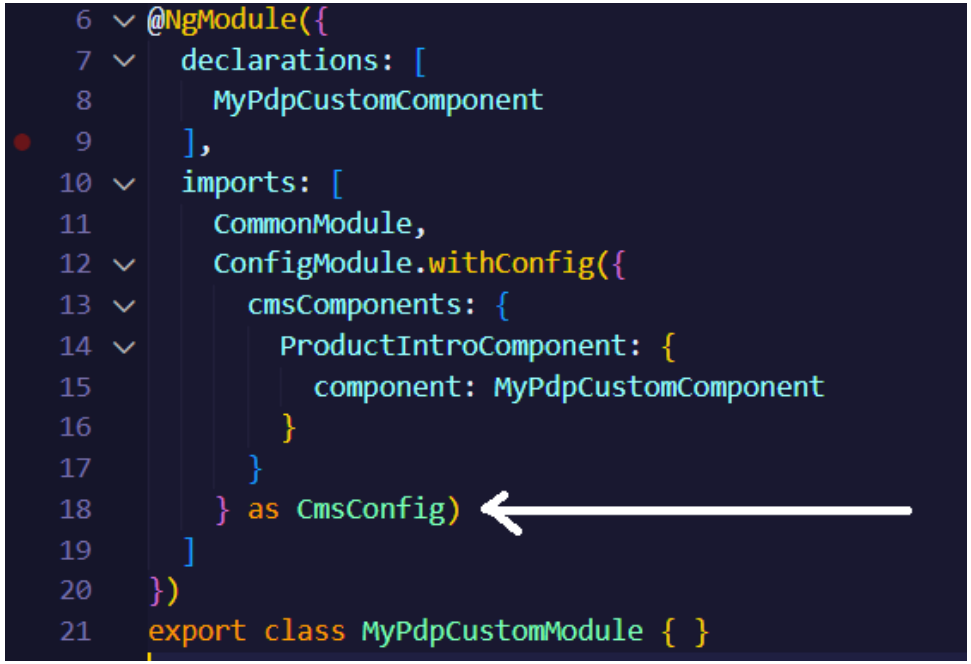


If you look at above screenshot, you will get an idea of using **ConfigModule.** Inside withConfig method if you pass the object and if you press CTRL + Space, VScode will give you available configuration settings from spartacus. The one which we are going to use to map our component is **cmsComponent**.

The next screenshot will show you complete settings for mapping CMS component with angular component.

The ProductIntroComponent is the name coming from CMS as **flexType** which we have used here in our configuration and as a value to that flexType component we have to provide object with name of your custom angular component.

One very small step I missed is to add cast to configuration object:



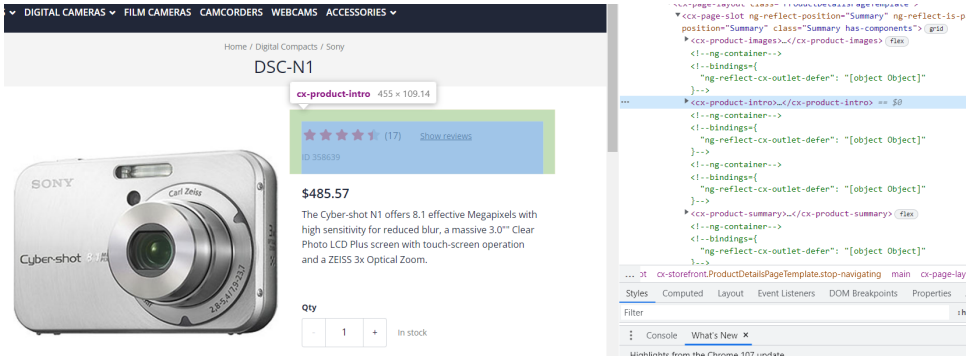
```
@NgModule({
  declarations: [
    MyPdpCustomComponent
  ],
  imports: [
    CommonModule,
    ConfigModule.withConfig({
      cmsComponents: {
        ProductIntroComponent: {
          component: MyPdpCustomComponent
        }
      }
    } as CmsConfig)
  ]
})
export class MyPdpCustomModule { }
```

Next step is to import your custom module inside app module of your application.

```
You, 4 minutes ago | 1 author (You)
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule.withServerTransition({ appId: 'serverApp' }),
    HttpClientModule,
    AppRoutingModule,
    StoreModule.forRoot({}),
    EffectsModule.forRoot([]),
    SpartacusModule,
    BrowserTransferStateModule,
    MyPdpCustomModule      You, 4 minutes ago • Uncommitted changes
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

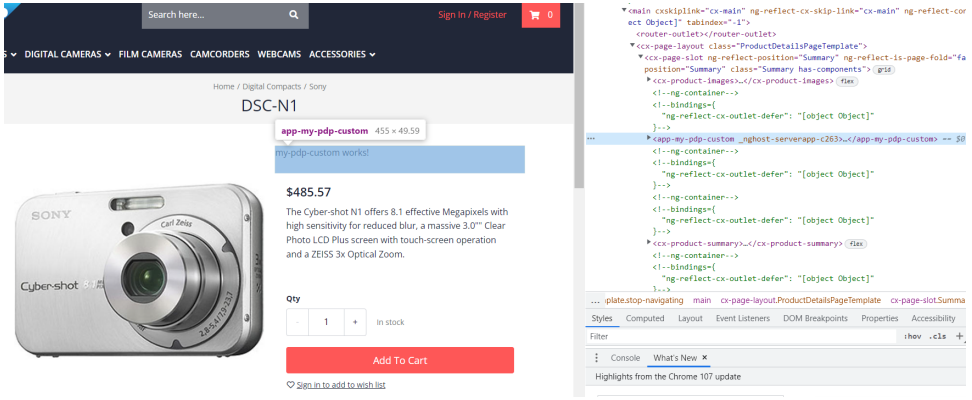
Below I have attached before and after adding the custom component.

Before:



You can see in the inspect section of developer console we have **cx-product-intro** which is coming from CMS.

After:



After successfully adding our custom component you can see that we have replaced spartacus out of the box ProductIntroComponent with our custom angular component. You can also see in the inspect section of developer tools where you can see **app-my-pdp-custom** which is our custom angular component we created in earlier steps.

And that's it guys for today! I hope you have learned a lot from this second part of my series on Customizing PDP component.

Please share this article with your friends, colleagues, team mates and help them too.

Sharing will motivate me to write more and more on this topic! Cheers!

Published by

Avinash Jadhav
SAP Composable Storefront(formerly Spartacus) Trainer / Coach / Expert | Ang...
Published • 1y

5 articles

✓ Following

Hi #connections , 2nd part of my article series SAP Spartacus : Product details page customization is available now. #sap #spartacus #sapcommercecloud

#sapcommerce #storefront #sapcommunity #headless #headlesscommerce #vuestorefront

Like Comment Share

aditya yadav and 3 others 1 comment

Reactions

1 Comment

Add a comment...

aditya yadav (He/Him) • 1st
Technical Lead

1y •••

Great 👍

Like | Reply

Avinash Jadhav

SAP Composable Storefront(formerly Spartacus) Trainer / Coach / Expert | Angular | React.js

✓ Following

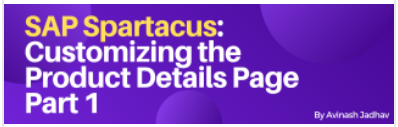
More from Avinash Jadhav

Customizing Your Store with LayoutConfig in SAP Spartacus

Avinash Jadhav on LinkedIn

SAP Spartacus: The Key to Greater Success as an Angular Developer

Avinash Jadhav on LinkedIn



SAP Spartacus: Customizing the Product Details Page Part 1

Avinash Jadhav on LinkedIn

See all 5 articles