



SAP Customer Experience

Composable Storefront Briefing

INTERNAL – SAP and Partners Only

We will learn about:

- Composable Storefront Overview
- Component Storefront - Essentials & Features

The Context



SAP Commerce Cloud, Composable Storefront is a lean, Angular-based **Single-Page Application (SPA) storefront** for SAP Commerce Cloud.

Composable Storefront Overview

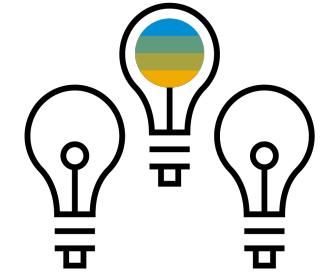


What is Composable Storefront?

- Starting with 5.0, “SAP Commerce Cloud, composable storefront” is the name of the **strategic** storefront solution for SAP Commerce Cloud (the Accelerator storefronts are now deprecated).
 - It has its own help portal:
https://help.sap.com/docs/SAP_COMMERCE_COMPOSABLE_STOREFRONT
- It is a JavaScript storefront for SAP Commerce Cloud
 - **Angular**-based
 - Single-page application (**SPA**)
 - Progressive web application (**PWA**)
- It is based on the Spartacus open source code
 - Managed and developed by SAP.
 - Provided with official lines of support on Stack Overflow or SNOW
 - Feedback, ideas, requests, and code contributions are welcome
- It consists of a set of libraries: *core*, *storefront*, *cart*, *product*, *styles*, *assets* ...
 - Downloaded from the **Repository Based Shipment Channel(RBSC)**

Why Composable Storefront?

- **Extendable** - Composable Storefront is designed to be easily extendable
 - You can build your own custom storefront without any limitations
- **Upgradable** - Upgradability of prior solutions was difficult
 - With Composable Storefront, you can easily adopt new minor and patch releases at any time
- **Progressive** - Support for all major features of a PWA-enabled storefront
 - Aims to be fully PWA-compliant in the future
- **Open Source** - It will be continually developed by the SAP Commerce Cloud team; contributors are welcome and help foster an inclusive, active development community
- **Modern** - Keep up with the most modern technologies towards a customizable-yet-upgradable technology for SAP Commerce Cloud installations
 - For instance, the Composable Storefront roadmap is aligned with the Angular roadmap, and is upgraded to the newest version as soon as possible



Composable Storefront Features

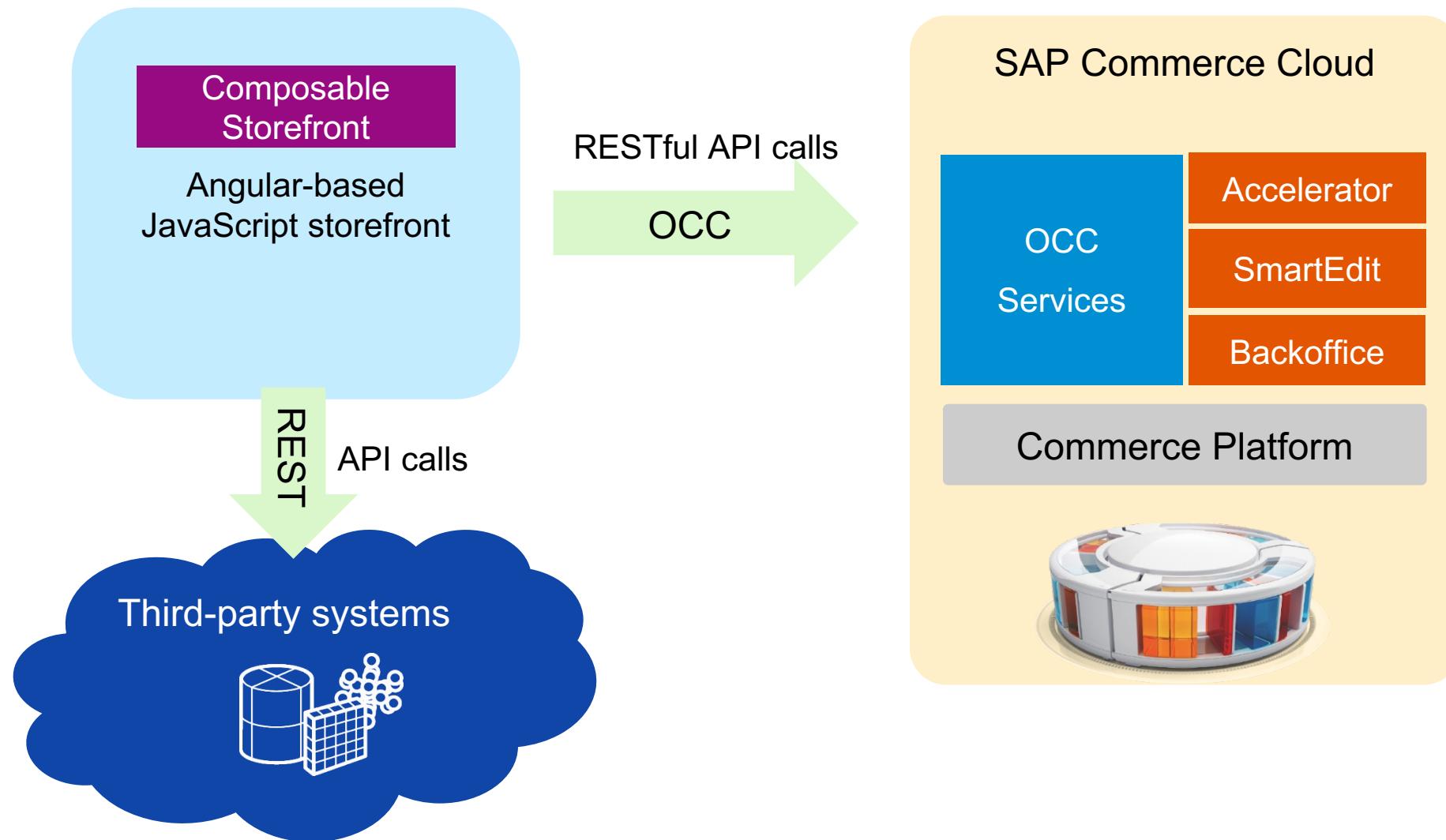
Composable Storefront provides core storefront features such as:

- Anonymous consent
- Assisted Service Module
- Cancellations and returns
- Consignment tracking
- Coupons/Customer Coupons
- Customer interests
- Notification preferences
- Stock notification
- Store locator
- Guest/Express checkout
- Wish list
- B2B commerce organization
- Pre-built components
- SEO
- i18n
- OCC integration
- SmartEdit integration
- Customizable styling
- SSR
- PWA
- CPQ Configurable Product integration
- SAP Customer Data Cloud integration
- Kyma integration
- Qualtrics integration

(check latest release notes for the most up-to-date features):

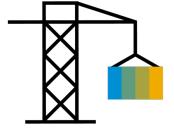
[What's New in SAP Commerce Cloud, composable storefront](#)

SAP Commerce Cloud Architecture



Composable Storefront – Essentials & Features





Development Environment Tools

Standardized in the project

- **CLI tools**

- **Node**
- **Javascript Package Manager**
 - **yarn** or **npm**
 - Only 1 lock file is used (shared)
- **Angular CLI**



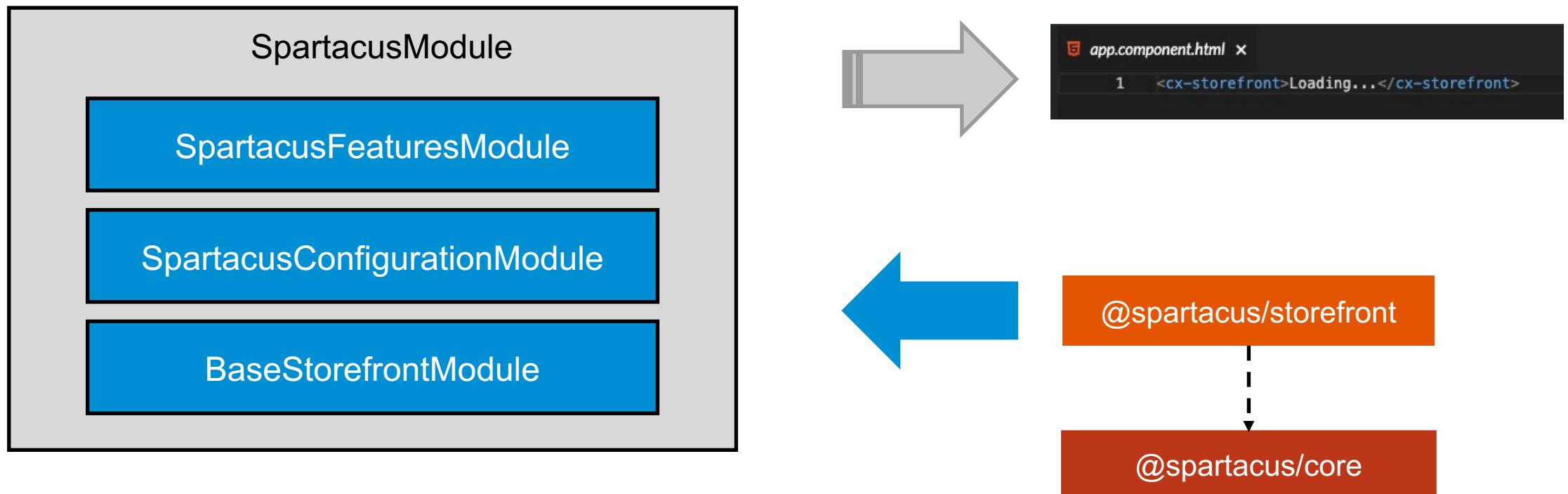
- **IDE**

- **Visual Studio Code** (Recommended)



Composable Storefront Module

Composable Storefront modules offer **pre-built storefront features and configurations**. It takes only a few minutes, without coding, to create a standard Composable Storefront from scratch.





Outlets

- Outlets allow customization of the standard UI by plugging a custom UI into the standard Composable Storefront DOM.
- Outlets are places in your application where your custom code can be injected
 - You can add code to these places or entirely replace them

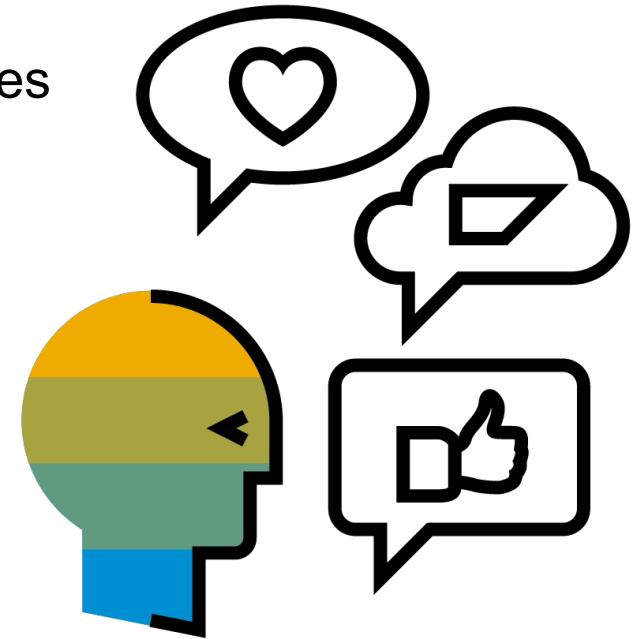
The screenshot shows a browser window displaying the SAP homepage at <https://localhost:4200/USD/en/electronics-spa/>. The page features the SAP logo, a search bar, and a navigation menu with links like BRANDS, DIGITAL CAMERAS, FILM CAMERAS, CAMCORDERS, WEBCAMS, and ACCESSORIES. Below the header, there's a main content area with sections for Section1, Section2A, Section2B, Section2C, Section3, and Section4. A blue box highlights the `<header id="cx-header" ...>` element in the browser's Inspector tab. A large blue bracket on the right side of the screen, labeled "Outlet Examples", spans from the highlighted header element down to the bottom of the page, covering the entire content area.

Outlet Examples



Component Styling

- Customize Composable Storefront **global theming**
- Customize **component styles**
- Use **Flat DOM** to avoid opinionated hierarchy
- Completely opt-out from **built-in styles**
- Selectively opt-out from **component styles**
- Avoid breaking **style changes** between Composable Storefront releases



Integrated with SmartEdit OOTB



- With simple configuration (via Backoffice and Composable Storefront config)

The screenshot shows the SAP SmartEdit Backoffice interface. The top navigation bar includes 'Navigation', 'Pages', 'Storefront' (which is highlighted in blue), and 'Sites'. The top right shows the current page as 'SPA Electronics Content Catalog - Staged | English | Electronics Product Catalog (Online)'. Below the navigation is a toolbar with icons for 'Advanced Edit', 'Page Structure', 'Component', 'Sync', 'Versions', 'Clone', 'Info', 'Trash', and a sync status indicator ('Synced'). The main content area displays a storefront layout with a header containing a SAP logo, a search bar ('Enter product name or SKU'), a sign-in link ('Sign In / Register'), and a shopping cart icon ('0'). A navigation menu below the header includes 'BRANDS', 'DIGITAL CAMERAS', 'FILM CAMERAS', 'CAMCORDERS', 'WEBCAMS', and 'ACCESSORIES'. The main content area features a large image of cameras and lenses with a 'SAVE BIG' promotional banner. The bottom of the page has a footer with a 'SAVE BIG ON SELECT CAMERA ACCESSORIES & SUPPLIES' message and a 'SHOP NOW' button.



Routing URLs

▪ SEO

- Enhances Product searchability

▪ Stateful URLs

- Maintain the application State

▪ Pretty URLs

- Human readable format

▪ Configurable URLs

- Decide what info is included in the URL

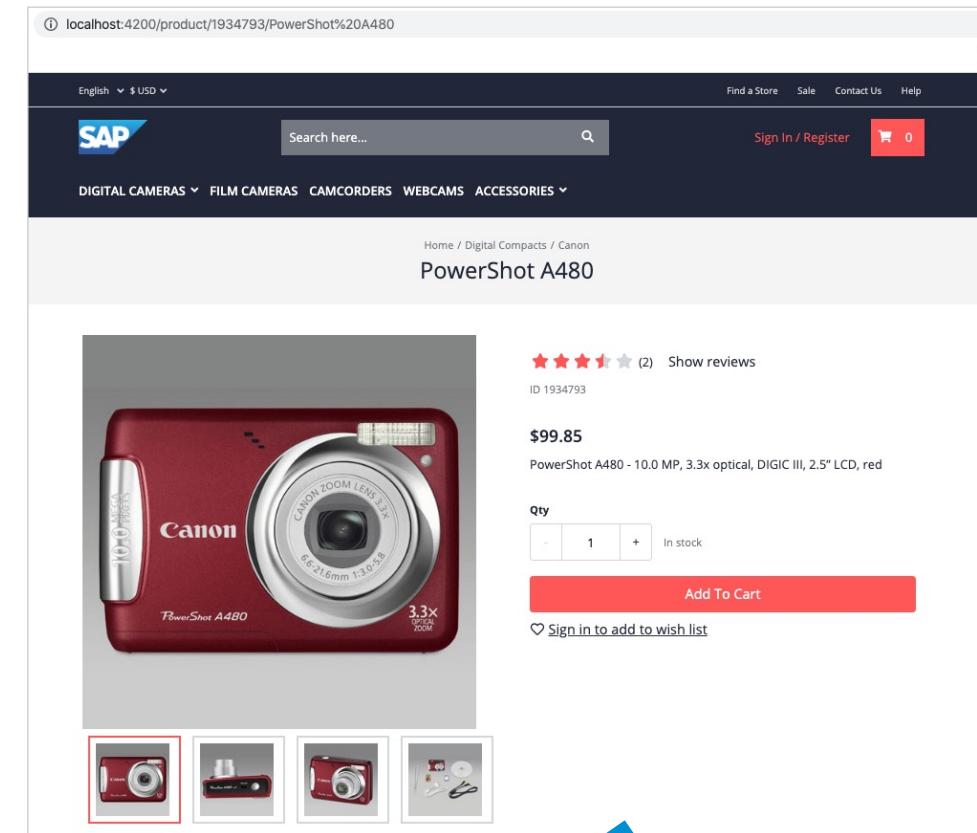
▪ Migration

- Keep legacy URLs working

▪ Integration

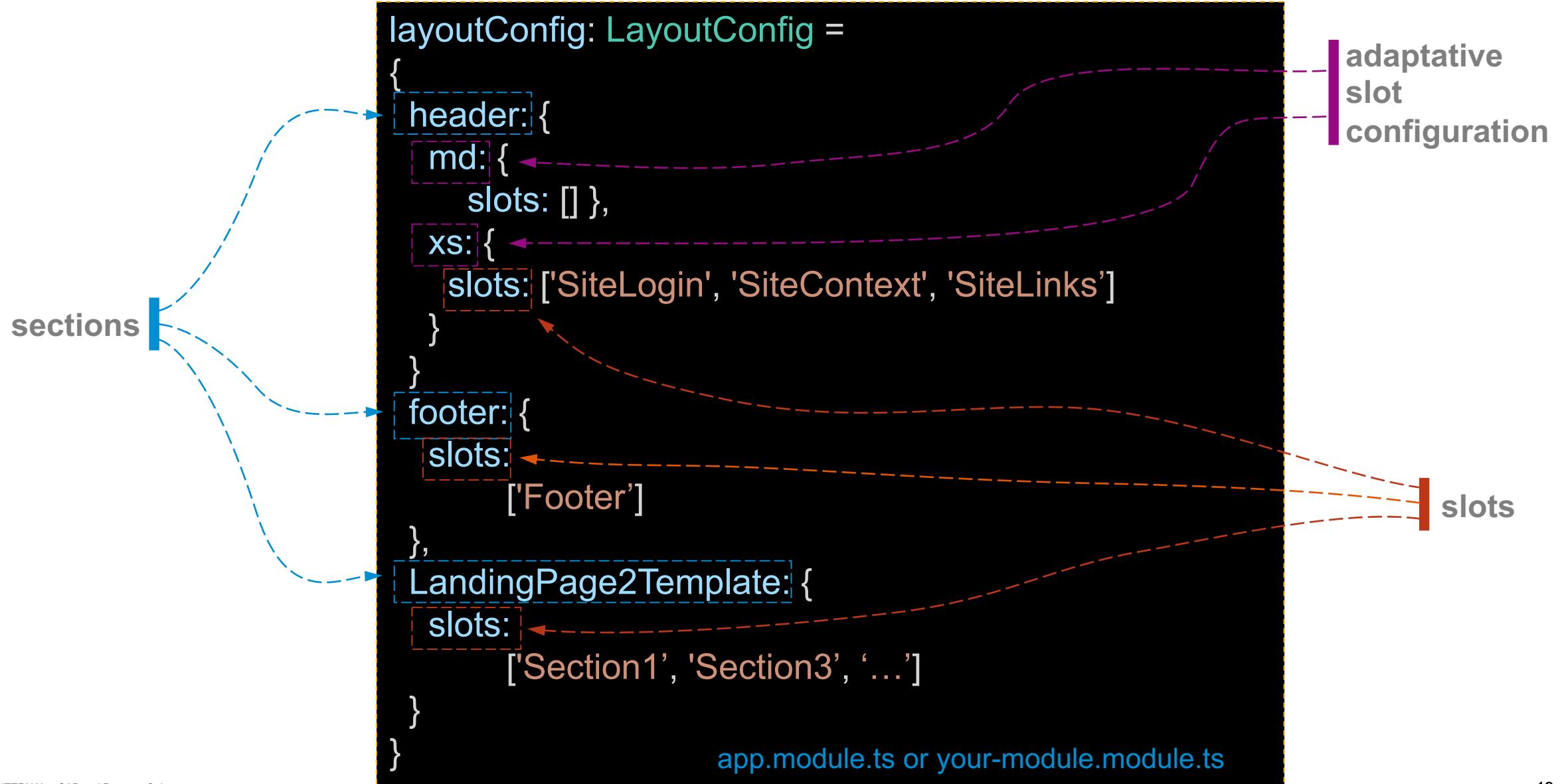
- Provide working URLs usable by third parties

<http://localhost:4200/product/1934793/PowerShot%20A480>
(default Composable Storefront PDP URL format)





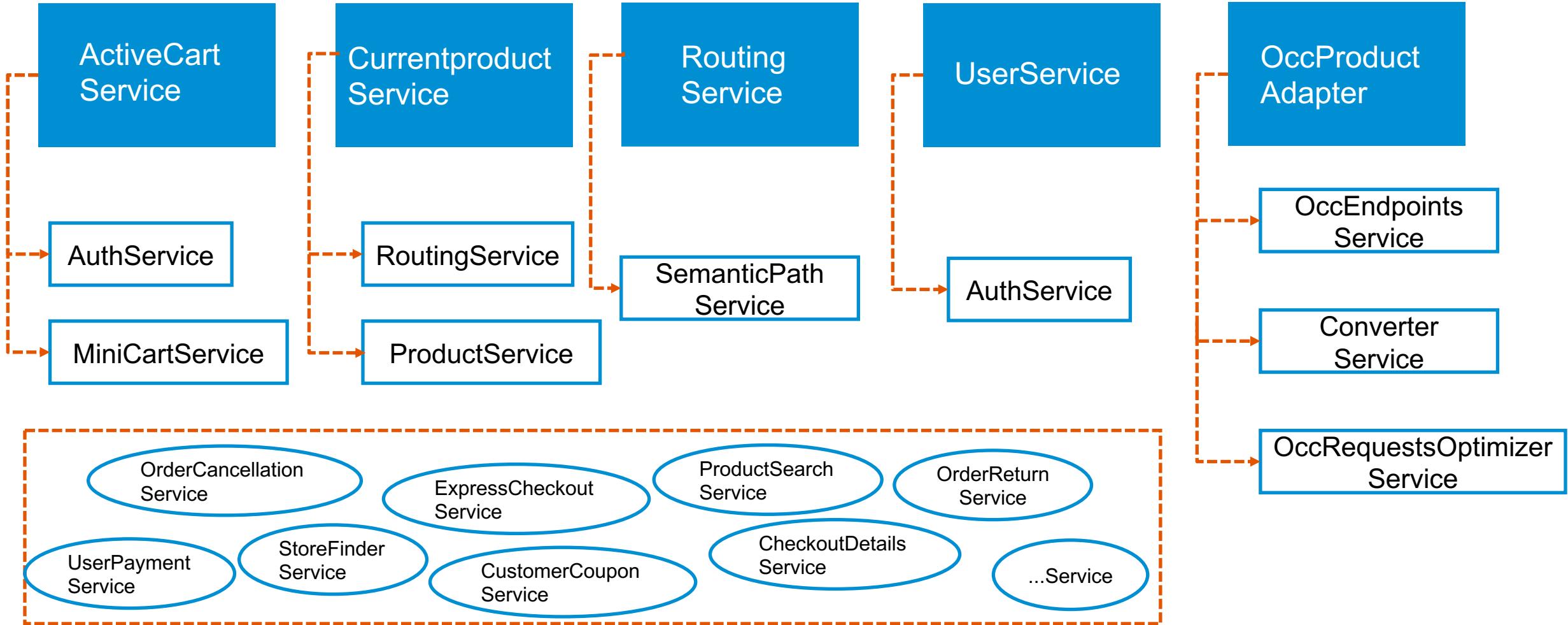
Layout Config





Standard Services

Get the complete list of injectables: <https://sap.github.io/spartacus/>





Contributing to the Project

- We welcome contributions in all forms.
- **Helping others** who may have questions or need support. Look for such requests here:
 - [Stack Overflow posts tagged with 'spartacus-storefront'](#) to get answers from our developers and the composable storefront community
 - Create a [customer support ticket](#) if you have an SAP Commerce Cloud license
- **Reporting issues.** Bug reports are welcome!
 - We are using [GitHub issue tracking](#) to track user stories and bugs.
- **Analyzing issues.** Don't have to be a programmer to help us determine the specifics of a bug.
 - Check [list of open issues](#) and [list of issues where analysis is requested](#).
- **Contributing code** to the Spartacus codebase.
 - You must agree to the [Contributors License Agreement](#).
 - Contributions must be compliant with the project code style, quality, and standards
 - Follow the [Contribution process](#)

We look forward to hearing from you!

SAP Spartacus Developer Training C4H370

- In this class we teach how to install and develop all aspects of a customized Spartacus storefront
- The target audience is technical, mainly focused on backend developers, frontend programmers and software architects
- It's a 3-day Instructor-led class
- The topics are: Overview, Setup & Libraries, Outlets, Custom CMS Components, Global Configuration, Routing, Styling, Layout Config, Internationalization, Dependency Injection & Built-in Services, SEO, Data Binding, Spartacus Internals
- We recommend participants be familiar with modern SPA Javascript-based frameworks such as Angular/React/Vue, frontend web technologies, and software development in general

For booking, please go to [Spartacus ILT Developer Training](#) (English US SAP Training Shop)

Key Points

1. SAP Commerce Cloud, Composable Storefront is an Angular-based SPA storefront for SAP Commerce Cloud (runs on top of the OCC API)
2. It is Open Source and the libraries are available for download from the RBSC repositories
3. Supports Angular schematics and recipe modules
4. Quick UI customizations through Outlets and CSS/SASS Styles
5. Interfaces with standard CMS Components to support page configuration using SmartEdit
6. Supports routing and SEO
7. Many Angular Injectables (Services), Components, Pipes, and Directives are provided OOTB
8. SAP Spartacus Training C4H370 has all you need to implement a custom Composable Storefront

References

- Help Portal of SAP Commerce Cloud, Composable Storefront:
https://help.sap.com/docs/SAP_COMMERCESCOMPOSABLE_STOREFRONT
- Stack Overflow posts tagged with ‘spartacus-storefront’:
<https://stackoverflow.com/questions/tagged/spartacus-storefront>
- GitHub issue tracking:
<https://github.com/SAP/cloud-commerce-spartacus-storefront/issues>
- List of open issues:
<https://github.com/SAP/cloud-commerce-spartacus-storefront/issues?q=is%3Aopen>
- List of issues where analysis is requested :
<https://github.com/SAP/cloud-commerce-spartacus-storefront/labels/triage/analysis%20wanted>
- Contributors License Agreement:
<https://github.com/SAP/spartacus/blob/develop/LICENSE.txt>
- Contribution Process:
<https://sap.github.io/spartacus-docs/contributing-to-the-spartacus-storefront/#contribution-process>
- Create a support ticket:
<https://launchpad.support.sap.com/#incident/create>

Composable Storefront Exercise



Thank you.

