

React v16.12.0 as framework

- Developed with Node v10.16.0
- Using this React version provides [Hooks](#). Writing code with Hooks it becomes more readable and reusable
- Also we implement **state management with Hooks** (useReducer + useContext) and avoid using third-party libraries such as Redux
- <https://create-react-app.dev/> for creating first scaffolding as you get quickly all dev tooling ready

Folder structure by features

- **'components/products'** for products features and **'components/summary'** for order summary
- **'components/shared'** as a small UI component library susceptible to be reused in other apps (index.js per folder for easy imports).
- **'common'** for common services
- This structure invites to add new functionality in a clean way

DataService

- In data.service we abstract data needs for all the components. We create a 'db.json' to simulate a data source that could be easily replaced for fetching remote data

Discounts driven for data source

- Two available discounts, "N_FOR_1" and "BULK", implemented as single files inside discounts folder.
- Data source brings discount rules for each product
- So we get **dynamic discount logic** driven for data source. **Stakeholders can change business rules without development team dependencies.**

Continuous integration

- Each JS file covered for unit tests for having robust maintenance and scalability
- **'npm run test -- --coverage --watchAll'** command for unit testing & coverage results.
- Jest testing framework (default for create-react-app). Enzyme dependency for some DOM testing
- Coverage > 80% :

PASS	src/App.test.js
PASS	src/components/__tests__/ProductDetailModal.test.js
PASS	src/components/common/__tests__/CtaButton.test.js
PASS	src/components/summary/__tests__/Summary.test.js
PASS	src/common/__tests__/state.service.test.js
PASS	src/components/products/__tests__/Products.test.js
PASS	src/components/__tests__/ShoppingCart.test.js
PASS	src/components/__tests__/MainContainer.test.js
PASS	src/components/summary/__tests__/SummaryDiscounts.test.js
PASS	src/components/products/__tests__/ProductListItem.test.js
PASS	src/common/__tests__/checkout.test.js
PASS	src/common/__tests__/helper.service.test.js
PASS	src/components/common/__tests__/Title.test.js
PASS	src/components/summary/__tests__/SummaryTotal.test.js
PASS	src/components/summary/__tests__/SummaryItems.test.js
PASS	src/common/services/discounts/__tests__/BULK_DiscountService.test.js
PASS	src/components/products/__tests__/ProductListHead.test.js
PASS	src/common/__tests__/models.test.js
PASS	src/components/products/__tests__/ProductList.test.js
PASS	src/common/services/discounts/__tests__/N_FOR_1_DiscountService.test.js
PASS	src/common/__tests__/data.service.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files					
src	100	100	100	100	
App.js	100	100	100	100	
src/common/services	94.37	91.67	100	93.65	
checkout.js	100	92.86	100	100	23
data.service.js	100	100	100	100	
helper.service.js	91.67	100	100	90	15,29
state.service.js	87.5	75	100	86.67	34,43
src/common/services/discounts	100	100	100	100	
BULK_DiscountService.js	100	100	100	100	
N_FOR_1_DiscountService.js	100	100	100	100	
src/components	87.5	87.5	50	87.5	
MainContainer.jsx	100	83.33	50	100	15
ProductDetailModal.jsx	80	100	33.33	80	22
ShoppingCart.jsx	100	100	100	100	
src/components/common	100	50	100	100	
CtaButton.jsx	100	50	100	100	5
src/components/products	72.22	66.67	60	70.59	
ProductList.jsx	100	100	100	100	
ProductListItem.jsx	64.29	50	42.86	61.54	27,43,44,46,58
Products.jsx	100	100	100	100	
src/components/summary	91.67	100	66.67	91.67	
Summary.jsx	100	100	100	100	
SummaryDiscounts.jsx	75	100	50	75	15
SummaryItems.jsx	100	100	100	100	
SummaryTotal.jsx	100	100	50	100	

Other notes

- **Components organization by creating small components** helps reusability and unit testing
- **SplitPane layout** component shared for MainContainer and ProductDetailModal. It is easy to create more screens / functionalities with the same layout

- All **styling** are located in index.css (for resets) and App.css (for main app styles). This decision is based on the assumption that there is an UI engineer that can be in charge of maintaining styles
- **'npm run build'** command for building a production release. We get benefit of using default create-react-app builds to get optimization techniques such as code splitting for long term caching techniques and concatenation for better load performance

```
40.27 KB (+1 B)    build/static/js/2.734139ab.chunk.js
18.75 KB (+762 B) build/static/js/main.de8fa8ca.chunk.js
1.96 KB (+87 B)   build/static/css/main.d0d61589.chunk.css
770 B (-1 B)      build/static/js/runtime-main.8eea71de.js
```