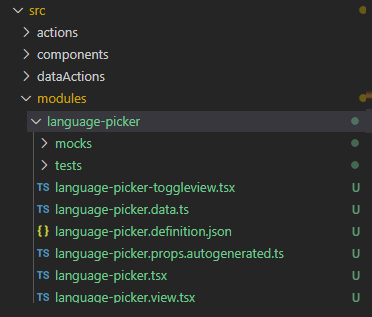
**Custom-Language Picker Module**

**Language Picker Module**

The language-picker module can be added anywhere in the header or footer and user can pick a language from the dropdown and get redirected to the page with that locale.

1. **Add the custom module from sample source code.**
   1. Unzip the language-picker source code.
   2. Copy the module code of language-picker module under “/src/module” as below:



* 1. **Definition File Explained**: Please refer below table for the configurable settings for this module based on the need.

**Configurations (language-picker.definition.json)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property Name** | **Type** | **Description** | **Default** |
| languages | Object array | defined as an Array of Objects in Config. Below are the four properties that are there in the object in the array | - |
| * Name | string | Name of the language which is the display text on the UI (view) | - |
| * Code | String (Required) | Locale specific code which is unique for each language. (en, en-us, fr) | - |
| * imageUrl | String | Image for the locale can be shown next to the language name. This image source url can be set up from cms | - |
| * linkUrl | String | This field can be mandatory if we need to redirect to a specific url for a locale all the time. To do that, we need to add required attribute to this property in the definition file as "required": true | - |
| linkRedirectFlag | Boolean | flag to check if we need to redirect to a specific link.  If true, we will redirect to link url for that language. | false |
| linkBaseUrl | String | User can make this field as mandatory if customer wants to append locale to a specific base url. To do that, we need to add required attribute to this property in the definition file as "required": true |  |

**Resource Strings**

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Description** |
| selectLocaleText | Language | Text/label for the language picker |

* 1. **Business-logic: (language-picker.tsx)**

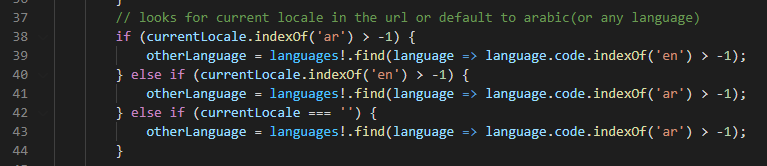
The module will be displayed on the header for user to select a language. Once user selects a language, he will be redirected to a specific landing page for that locale if the link redirectFlag is true. If the flag is false, locale for the chosen language will be appended to the base url set in config and the query parameters in the current url (from request context) will also be appended and then redirected to that page. This way user will be on the same page with only locale code getting changed.

* 1. **View: (language-picker.view.tsx)**

There are two possible ways to configure a view for language-picker.

1. **Dropdown view**: The view is a dropdown view and once user selects a language from dropdown, it will redirect to a page defined by the business logic above. If the current page url has a locale code appended, then that language will be shown as the active selection when user comes directly to that page next time /stays on that page.
2. **Toggle view:** If there are only two locales, we have a toggle implementation for the view which is present in language-picker-toogleview.tsx. The code from this file can be copied to the view file(language-picker.view.tsx) or this file can be renamed to language-picker.view.tsx to make it work.

**Note**: As toggle view works for only two locale support to toggle between, below code in “*language-picker-toggleview.tsx*” need to update for specific locales supported for a particular site.

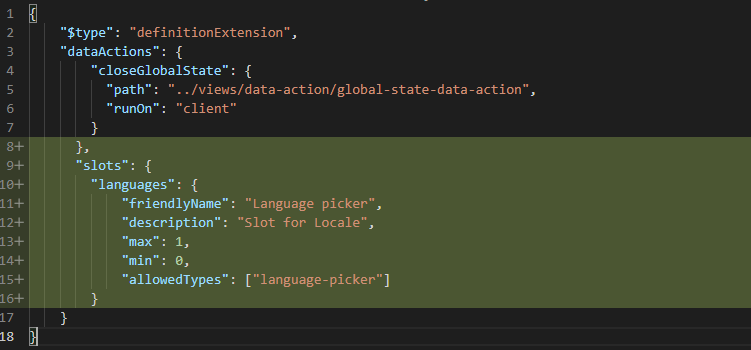


1. **If further customization is required, module definition and view file can be updated.**
2. **Build the repository**:

Execute yarn build from command line interface on your repository.

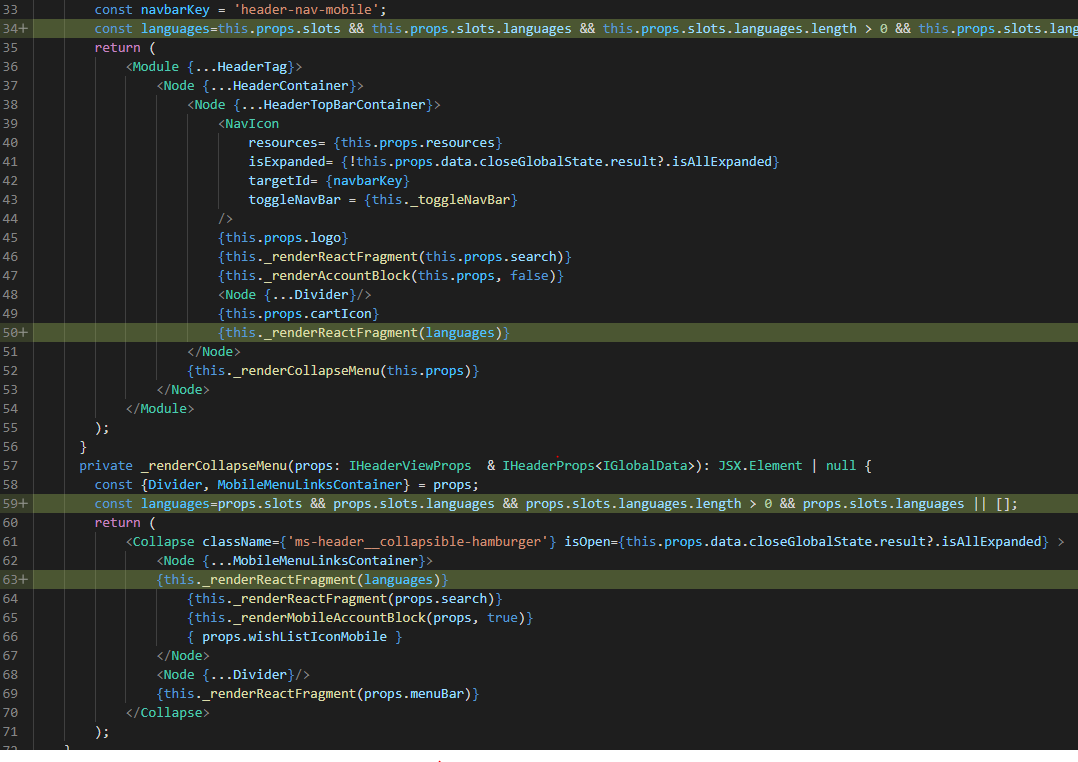
1. **Adding Site-picker module to header**: Extend header module definition and header module view to include language-picker.
   1. **Header.definition.ext.json:** Check for this file under src/themes/”customer-theme”/definition-extensions and add a new slot for language-picker module. (refer highlighted code snippet below).

**Note:** here the yellow-highlighted is the name of the theme folder associated with the site.



* 1. **Header View extension: (Header.view.tsx):** Check for this file under src/themes/”customer-theme”/views and make below changes in this file to render the language picker.

**Note:** here the yellow-highlighted is the name of the theme folder associated with the site.



1. **Module Styling (scss):** The css for this module(language-picker.scss) file should be placed in styles folder for modules css and the reference should be made in index.scss. Sample file attached.



1. **Testing:** The testing can be done by using the homepage mock in the pageMocks folder (**homepage.json** ) and the code can be rendered locally by hitting the below url

<https://localhost:4000/Pages?mock=homepage>

1. **CMS Configuration post deployment**: Add the language-picker module to header fragment and configure the same as per above details.

Follow below steps in CMS:

1. Open site authoring and click on fragments tab under the site to find the header fragment.
2. Click on any container, click three dots to Add language-picker module in that.
3. Add configuration to language picker module.
4. Save and publish the changes.
5. Browse your site to test language picker in the header.