Menú lateral

El menú lateral es genera de forma dinàmica a partir d'informació proporcionada per l'aplicació.

El menú s'estructura com una llista de seccions, i cada secció està formada per un conjunt d'items

Aquesta estructura de dades el proporciona l'aplicació mitjançant un `@Component` que implementa l'interface `cat.dipta.starters.layout.menu.MenuService`.

```
@Component
@Slf4j
public class MyAppMenuService implements MenuService {

    @Override
    public List<MenuSection> buildMenu() {
      [...]
    }

}
```

Com exemple es pot consultar la implementació del mòdul de guia: GuideMenuService.java

Cada secció de menú (MenuSection) conté les dades següents:

- String id: Identificador de la secció de cara a poder-la definir com a activa
- String iconClass: Classe de l'icona de la secció, basat en FontAwesome 4.7.0
- String label: Etiqueta que apareixerà al menú
- String description: Títol (tooltip) de l'etiqueta
- List items: Llistat d'items de la secció
- Map<String,Object> properties: Llistat de parells clau-valor per extendre la funcionalitat (exemple: seguretat)

Exemple:

```
@Override
public List<MenuSection> buildMenu() {
    List<MenuSection> menu = new ArrayList<>();

    // Afegir secció 1
    MenuSection ms = new MenuSection();
    ms.setId("conceptes");
    ms.setIconClass("fa fa-th-large");
    ms.setLabel("Conceptes");
    ms.setDescripció("Gestió de conceptes d'ingrés");
    [ ... afegir els ítems a la secció ...]
    menu.add(ms);

    // Afegir la resta de seccions
    [ ... ]

    return menu;
}
```

Igual que en cas de les seccions, cada item (MenuItem) està definit per les següents dades:

- String id: Identificador de l'item de cara a poder definir-lo com a actiu
- String href: URL de destí de l'acció, interna a l'aplicació, i relativa al *context path*
- String absHref: URL absoluta de destí de l'acció, en cas que sigui externa a l'aplicació
- String label: Etiqueta de l'opció de menú
- String description: Descripció que apareixerà com a títol de l'etiqueta
- Boolean newWindow: Indica si l'enllaç s'ha d'obrir en una finestra nova (`target="_blank"`)
- Map<String,Object> properties: Llistat de parells clau-valor per extendre la funcionalitat (exemple: seguretat)

La classe MenuItem inclou un `Builder` per facilitat la creació d'instàncies.

Exemple:

```
MenuItem item1 = MenuItem.builder()
      .id("llistaConceptes")
      .href("/admin/conceptes")
      .label("Llistat")
      .description("Accedir al llistat de conceptes")
      .property(RoleMenuPermissionEvaluatorImpl.ROLES_PROPERTY, "admin") // Restringir l'accés només al rol 'admin'
      .build();
menuSection.getItems().add(item1);
```

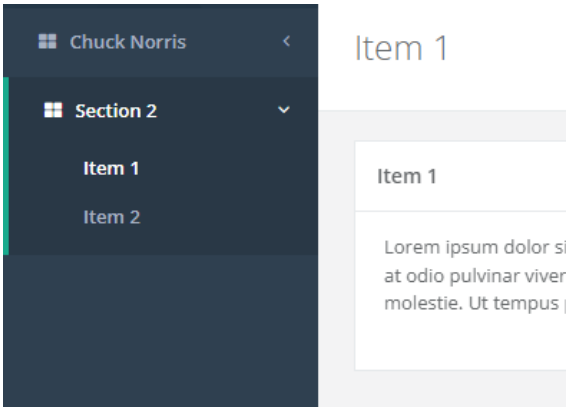
Opció activa

Quan la plantilla genera el menú per pantalla ressalta l'ítem actiu i desplega la secció de la que forma part, col·lapsant la resta de seccions.

Per indicar l'item actiu de la pàgina es fa servir el paràmetre `activeMenu` de la crida al layout

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
xmlns:th="http://www.thymeleaf.org"
      layout:decorate="~{layouts/guide(pageTitle='Item 1', activeMenu='item1')}">
```

En aquest cas, s'ha determinat com a element actiu l'item amb id = 'item1'



Com es pot veure en la imatge, l'opció de menú apareix ressaltada, juntament amb la secció "Section 2", de la que forma part. La primera secció del menú apareix replegada, tot i que l'usuari la pot desplegar per navegar pel menú

En el cas que el paràmetre `activeMenu` tingui per valor l'ID d'una secció, aquesta apareixerà desplegada però cap de les seves opcions apareixerà marcada

Seguretat

El procés de renderització del menú aplica un filtre de seguretat configurable que permet decidir quines seccions/opcions apareixen en funció del perfil de l'usuari.

Per defecte l'starter aporta dues implementacions: una basada en els rols de l'usuari i una altra *neutra* que no restringeix cap acció i mostra totes les opcions.

Per tal de configurar el comportament del filtre de seguretat cal informar la propietat `dipta.layout.menuPermissionType`. Aquesta propietat pot tenir dos valors: ROLE o DEFAULT (sent aquest el valor per defecte)

Seguretat per rol

La seguretat per rol s'activa configurant la propietat de l'aplicació `dipta.layout.menuPermissionType` amb el valor '*ROLE*'

`dipta.layout.menuPermissionType = ROLE`

Configurant la seguretat per rol d'usuari ens permet indicar per cada objecte de tipus MenuSection o MenuItem el rol (o llistat de rols) per als quals es mostrarà la secció (o opció) mitjançant la propietat `RoleMenuPermissionEvaluatorImpl.ROLES_PROPERTY`. En el cas que s'informi un conjunt de rols, es pot definir tant com una llista (`List<String>`) o un array (`String[]`).

Si per una secció/item no s'informa la propietat de rols, aquesta es mostrarà sempre independentment dels rols de l'usuari

Seguretat custom

Si es vol personalitzar el comportament del filtre de seguretat del menú, cal configurar la propietat `dipta.layout.menuPermissionType` amb el valor 'DEFAULT' i crear un `@Component` que implementa l'interface `cat.dipta.starters.layout.menu.MenuPermissionEvaluator`.

Aquest interface defineix dos mètodes que retornen un booleà indicant si s'ha de mostrar la secció (o item) passat per paràmetre:

```
public interface MenuPermissionEvaluator {

    public boolean hasPermissionOnSection(MenuSection section);

    public boolean hasPermissionOnItem(MenuItem item);
}
```

Si es necessita es pot fer servir el següent codi com a punt de partir per tal d'implementar un filtre personalitzat per l'aplicació:

```
@Component
@ConditionalOnProperty(value="dipta.layout.menu-permission-type", havingValue = "DEFAULT", matchIfMissing = false)
public class CustomMenuPermissionEvaluator implements MenuPermissionEvaluator {

  // .. Injectar els seveis/components necessaris per a avaluar els permisos ...

  @Override
  public boolean hasPermissionOnSection(MenuSection section) {
    // [ .. Avaluar l'accés a l'objecte section ...]
```

```
    return ...;
  }

  @Override
  public boolean hasPermissionOnItem(MenuItem item) {
    // [ .. Avaluar l'accés a l'objecte item ...]
    return ...;
  }

}
```

Quan el menú no s'adapta a la necessitat de l'aplicació

Si tot això no s'adapta a les necessitats de l'aplicació el que es pot fer és substituïr el fragment "sidemenu" per un de diferent, ja sigui a la plantilla de l'aplicació o a la de la pàgina concreta.

Per exemple:

```
<th:block layout:fragment="sidemenu">
    <li class="active">
        <a href="#"><i class="fa fa-th-large"></i> <span class="nav-label">Secció CUSTOM</span> <span class="fa arrow">
</span></a>
        <ul class="nav nav-second-level">
            <li><a th:href="@{/customItem1}">Item custom 1</a></li>
        </ul>
    </li>
</th:block>
```

Amb això substituirem l'HTML del menú que posa el layout pel propi de l'aplicació.

Tornar a l'índex