

```

flightMenu.js
class FlightMenu
-> menuTop()
/** appear/disappear Flight Menu
scroll up/down */

```

```

activeClass.js
class ActiveClass
-> classActive()
/** Set class active/not on aMenu */

```

```

removeIndex.js
function removeIndex()
/** removeChild() on article */

```

```

GetPhs.js
class GetPhs
-> listPhs(dataPhotos)
/** display boxProfil in section index page */

```

```

main.js
databaseAvalaible(r)
Header
editNav
flightMenu
tabIndex
keyDown

new Header().getHeader(r)
new Header().getNavFilter(r)
new flightMenu().menuTop()
('.entete').addEventListener("click",editNav)
new TabIndex().setTab();
new KeyDown().rightArrow();

```

```

json2Js.js
databaseAvalaible
/** Catch the json */

```

```

navResp.js
function editNav()
/** add responsive class/not
to menu for css*/

```

```

tabIndex
class TabIndex
-> setTab()
/** set tabindex in
index Html page with
setAttribute */

```

```

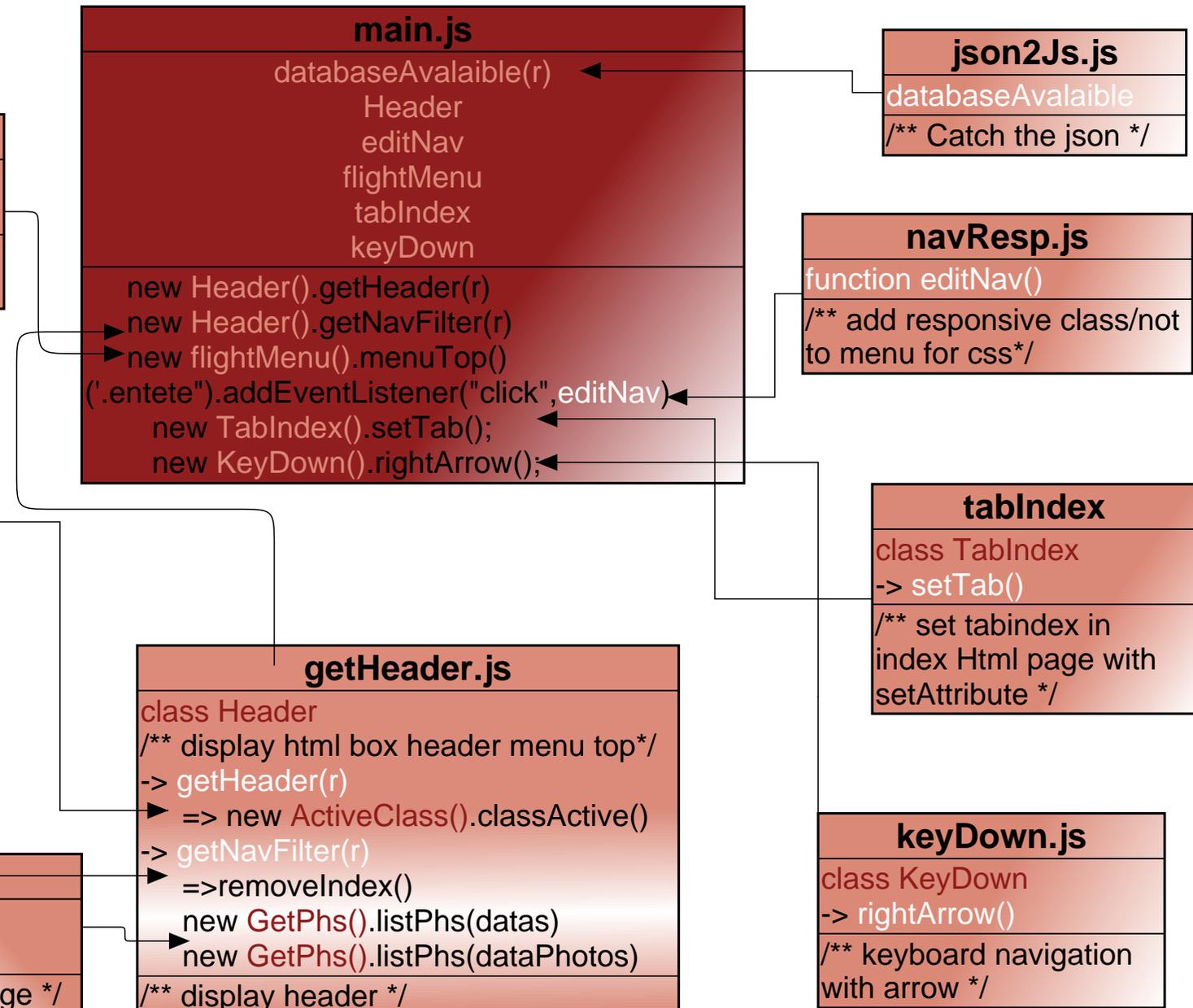
getHeader.js
class Header
/** display html box header menu top*/
-> getHeader(r)
=> new ActiveClass().classActive()
-> getNavFilter(r)
=>removeIndex()
new GetPhs().listPhs(datas)
new GetPhs().listPhs(dataPhotos)
/** display header */

```

```

keyDown.js
class KeyDown
-> rightArrow()
/** keyboard navigation
with arrow */

```



FISHEYE
Photographer
Page

```
DropDown.js  
Modal  
Form  
LightBox  
AddLikes  
DropDown().upDown(r)  
-> displayMediaSort(dataMediaSort)
```

```
AddLikes.js  
class AddLikes  
/** on heart empty click change to  
full retire/add 1 */  
-> adHeart()  
/** pick up all likes and sum up */  
-> var fn totalLikes()
```

```
tabIndexPh.js  
TabIndexPh().settriSelect()  
/** setAttribute tabindex in  
DropDown Menu */
```

```
keyDownPh.js  
KeyDown().rightArrow(r)  
/**addEventListener keydown  
left and right Arrow  
up/down dropdown */  
-> var fn  
displayMediaSort(dataMediaSort)
```

```
photographer.js  
databaseAvailable(r)  
Modal  
Form  
DropDown  
KeyDown  
TabIndex  
new Photograph().getProfilById(r)  
new Photograph().getProfilByTags(r)  
new Dropdown().upDown(r)  
new AddLikes().addHeart()  
new KeyDownPh().rightArrow(r)  
new TabIndexPh().settriSelect()  
Class Photograph  
/** id from url = Ph id **/  
->getProfilById(r) => display Ph header Html  
/** id from url = Md photographerId **/  
->getProfilByTags(r) => display Md Html  
new Modal().showHtmlModal(),  
new Modal().addModal();  
new Form().getFields();  
new LightBox().addModal();
```

```
json2Js.js  
databaseAvailable  
/** Catch the json */
```

```
Slider.js  
SliderModal()  
Slider system + -
```

```
lightBox.js  
SliderModal  
/**Display Modal slider*/  
LightBox().addModal()  
new SliderModal(pv, index)
```

```
modal.js  
/** display Modal */  
Modal().showHtmlModal()  
/** open / close */  
Modal().addModal()
```

```
form.js  
Form().getFields()  
checkFirstName  
checkLastName  
checkMessage  
checkEmail  
displayNoneErrorSelector  
displayBlock  
displayNone
```

FISHEYE Page Index

main.js - json2js.js

Datas().getJSON() récupère de manière asynchrone la base de données json.
la fonction databaseAvalaible permet de disposer des données

- getHeader.js

- Header().getHeader(r) affiche le header Html : logo, menu flottant (passer au contenu)

activeClass.js

-> ActiveClass().classActive() pour ajouter/retirer class : active dans le menu afin de gérer en Css le style background "bordeaux" et color white sur le lien sélectionné.

- Header().getNavFilter(r) affiche la liste des photographes en fonction du tag cliqué dans le menu

removeIndex.js

-> fonction removeIndex() supprime la liste

GetPhs.js

-> GetPhs().listPhs(datas) affiche la liste des photographes filtrés via le forEach

-> GetPhs().listPhs(dataPhotos) affiche tous les photographes, par défaut sans filtre

- flightMenu.js

- FlightMenu().menuTop() fait apparaître/disparaître le menu flottant (passer au contenu) en scrollant up/down avec la souris.

- navResp.js

- la fonction editNav ajoute/retire la class responsive à la class entete Html pour gérer le menu responsive
Css trois petites lignes en haut à droite.

- tabIndex.js

- TabIndex().setTab() assigne des tabIndex aux différents éléments Html : logo, menu, titre... pour la navigation au clavier.

- keyDown.js

- keyDown().rightArrow() permet la navigation au clavier dans le menu en utilisant la flèche droite, l'utilisateur rentre dans le bloc menu et peut voyager d'un lien à l'autre avec tabulation/maj tab

FISHEYE Page Photographe

photograph.js

- json2js.js

Datas().getJSON() récupère de manière asynchrone la base de données json.
la fonction databaseAvalaible permet de disposer des données

-> Photograph().getProfilByld(r)

affiche le Html header correspondant à l'id dans l'url cliqué dans la page index du menu
nom, lieu, tagline, tags, portrait du Photographe sélectionné

-> Photograph().getProfilByTags(r)

affiche les blocs media du Photographe sélectionné en fonction du photographId, image, titre, prix, likes

Modal.js

->Modal()

-> showHtmlModal() crée le formulaire Html du Modal

-> addModal() affiche le Modal avec ses fonctionnalités accessibles

form.js

-> Form().getFields() vérifie la validité des champs du formulaire avant envoi

lightbox.js

-> LightBox().addModal() crée et affiche la lightBox image + titre et ses fonctionnalités accessibles

Slider.js

->SliderModal(pv, index) permet de faire défiler les images du light box en cliquant sur les flèches

tabIndexModal.js

-> TabIndexModal().tabIndexPhModal() ajoute les tabIndex dans le Modal Html
permet la navigation au clavier Tab/Maj Tab

- DropDown.js

-> DropDown().upDown(r) gère le filtre popularité, date et titre au clic

- AddLikes.js

-> AddLikes().adHeart() ajoute/retire 1 au likes et au total likes

- keyDownPh.js

-> KeyDownPh().rightArrow(r) gère le filtre popularité, date et titre avec la navigation au clavier

- tabIndexPh.js

-> TabIndexPh().setTabIndexPh(r) ajoute tabIndex dans le dropdown select pour la navigation au clavier et exécute les filtres