

js\FlightCollection.js

```
1 import { Flight } from "./Flight.js";
2 import { Db } from "./db.js";
3
4 /**
5  * Gère la collection de vols
6  * @authors mdevoldere, grognone
7  */
8 class FlightCollection
9 {
10     /**
11      * Constructeur
12      * initialise le tableau de données
13      */
14     constructor()
15     {
16         this.data = [];
17
18         // valeur par défaut du sens de tri, si initialisé en true asc sinon desc
19         this.sortedAsc = true;
20     }
21
22     /**
23      * Chargement des données de la collection
24      */
25     async loadData()
26     {
27         this.data = await Db.getData("https://arfp.github.io/tp/web/frontend/flights/flights.json");
28
29         // console.log(this.data);
30
31         // pour chaque élément de la collection, on crée une instance de Flight
32         this.data = this.data.map(f => new Flight(f));
33     }
34
35     /**
36      * tri la collection par l'attribut flight_duration
37      */
38     sortByDuration(){
39         // fonction trie la collection en ordre croissant
40         this.data = this.data.sort(
41             (a,b) =>a.flight_duration - b.flight_duration
```

```

42     )
43
44     // si le boleen sortedAsc est évalué à faux, alors on inverse le tri
45     if (!this.sortedAsc){
46         this.data.reverse()
47     }
48
49     //on inverse le boleen comme cela le tri s'inversera au prochain clic
50     this.sortedAsc = !this.sortedAsc;
51 }
52
53 /**
54  * Filtre les données du tableau en fonction d'une recherche sur le nom de la compagnie
55  * @param {String} val valeur à rechercher
56  */
57 async searchByCompany(val)
58 {
59     await this.loadData();
60     val = val.trim();
61     if(val.length > 0)
62     {
63         val = val.toLowerCase()
64         this.data = this.data.filter(flight => flight.airline_name.toLowerCase().includes(val));
65         console.log(val);
66     }
67 }
68 }
69
70 export { FlightCollection }

```