

c:\grin-eat-tpi2022\grineat-client\index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>GrinEat - Home</title>
8   <script src='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.js'></script>
9   <link href='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.css' rel='stylesheet' />
10  <script
11    src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v5.0.0/mapbox-gl-
geocoder.min.js"></script>
12    <link rel="stylesheet"
13      href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v5.0.0/mapbox-
gl-geocoder.css"
14      type="text/css">
15    <link rel="stylesheet" href="./assets/css/style.css">
16 </head>
17 <body>
18   <header>
19     <h3 class="text-white">Grin<strong>Eat</strong></h3>
20   </header>
21   <div class="main">
22     <div class="search-container text-white text-center">
23       <h3>Faites vous livrer vos plats de votre restaurant favoris directement chez
vous grâce à <i>GrinEat</i>.</h3>
24       <div class="search-inputs-container">
25         <div id="idSearchBarContainer" class=""></div>
26         <button id="idBtnSearch" class="btn-search text-white">Chercher</button>
27       </div>
28     </div>
29   </div>
30   <footer>
31     <p>&copy; 2022 GrinEat, developped by <a href="https://github.com/miyokidev"
class="text-white">@miyokidev</a>.</p>
32   </footer>
33   <script src="./assets/js/main.js"></script>
34 </body>
35 </html>
```

c:\grin-eat-tpi2022\grineat-client\menu.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>GrinEat - Menu</title>
9     <script src='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.js'></script>
10    <link href='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.css' rel='stylesheet'
11    />
12    <script
13        src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v5.0.0/mapbox-
14        gl-geocoder.min.js"></script>
15    <link rel="stylesheet"
16        href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v5.0.0/mapbox-
17        gl-geocoder.css"
18        type="text/css">
19    <link rel="stylesheet" href="./assets/css/style.css">
20 </head>
21
22 <body>
23     <header>
24         <h3 id="idLogo" class="text-white">Grin<strong>Eat</strong></h3>
25     </header>
26     <div class="main-menus">
27         <div class="restaurant-info-container text-white">
28             <img id="idImage">
29             <div class="restaurant-name-city">
30                 <h2 id="idName"></h2>
31                 <p><span id="idCp"></span> - <span id="idCity"></span></p>
32             </div>
33             <div class="restaurant-street-number">
34                 <div>
35                     <h1>Adresse</h1>
36                     <p id="idStreet"></p>
37                 </div>
38                 <div>
39                     <h1>Numéro</h1>
40                     <p id="idPhone"></p>
41                 </div>
42                 <div id="idCategories">
43                     <h1>Catégories</h1>
44                 </div>
45             </div>
46         </div>
47         <div class="menu-items-container">
48             <div id="idListMenuItems">
49                 <div class="card-menu">
50                     
52                     <div class="card-menu-body">
53                         <p>Big Mac</p>
54                     </div>
55                 </div>
56             </div>
57         </div>
58     </body>
59 </html>
```

```
51         </div>
52
53         <div class="card-menu">
54             
55             <div class="card-menu-body">
56                 <p>6 Chicken McNuggets</p>
57             </div>
58         </div>
59
60         <div class="card-menu">
61             
62             <div class="card-menu-body">
63                 <p>6 Chicken McNuggets</p>
64                 <p class="card-menu-description">Le burger le plus légendaire du
monde est désormais encore meilleur. De la viande de bœuf bien juteuse de la boucherie Bell,
de la salade croquante et du fromage fondu, dans un petit pain fabriqué avec de la farine de
qualité IP-Suisse. Le tout rehaussé de la mystérieuse sauce Big Mac. Es-tu à la hauteur de
son goût ?</p>
65             </div>
66         </div>
67
68         <div class="card-menu">
69             
70             <div class="card-menu-body">
71                 <p>6 Chicken McNuggets</p>
72             </div>
73         </div>
74
75         <div class="card-menu">
76             
77             <div class="card-menu-body">
78                 <p>6 Chicken McNuggets</p>
79             </div>
80         </div>
81
82         <div class="card-menu">
83             
84             <div class="card-menu-body">
85                 <p>6 Chicken McNuggets</p>
86             </div>
87         </div>
88
89         <div class="card-menu">
90             
91             <div class="card-menu-body">
92                 <p>6 Chicken McNuggets</p>
93             </div>
94         </div>
95
96         <div class="card-menu">
```

```
97         
98         <div class="card-menu-body">
99             <p>6 Chicken McNuggets</p>
100         </div>
101     </div>
102
103     <div class="card-menu">
104         
105         <div class="card-menu-body">
106             <p>6 Chicken McNuggets</p>
107         </div>
108     </div>
109
110     <div class="card-menu">
111         
112         <div class="card-menu-body">
113             <p>6 Chicken McNuggets</p>
114         </div>
115     </div>
116
117     <div class="card-menu">
118         
119         <div class="card-menu-body">
120             <p>6 Chicken McNuggets</p>
121         </div>
122     </div>
123
124     <div class="card-menu">
125         
126         <div class="card-menu-body">
127             <p>6 Chicken McNuggets</p>
128         </div>
129     </div>
130
131     <div class="card-menu">
132         
133         <div class="card-menu-body">
134             <p>6 Chicken McNuggets</p>
135         </div>
136     </div>
137 </div>
138 </div>
139 </div>
140 <footer>
141     <p>&copy; 2022 GrinEat, developped by <a href="https://github.com/miyokidev"
class="text-white">@miyokidev</a>.
142     </p>
143 </footer>
144 <script src="./assets/js/menu.js"></script>
145 </body>
```

```
146 |  
147 </html>
```

c:\grin-eat-tpi2022\grineat-client\restaurants.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>GrinEat - Restaurants</title>
9     <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
10         integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq
11         /sMZM19scR4PsZChSR7A=="
12         crossorigin="" />
13     <!-- Make sure you put this AFTER Leaflet's CSS -->
14     <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
15         integrity="sha512-
16         XQoYMqMTK8LvdxXYG3nZ448h0EQiglfqkJs1NOQV44cWnUrBc8PkA0cXy20w0vlaXaVUearIOBhiXZ5V3ynxwA=="
17         crossorigin=""></script>
18     <link rel="stylesheet" href="./assets/css/style.css">
19 </head>
20
21 <body>
22     <header class="text-white shadow-header">
23         <h3 id="idLogo">Grin<strong>Eat</strong></h3>
24         <p id="idAddress" class="address-display">address</p>
25     </header>
26     <div class="tools-container">
27         <input id="idSearchName" type="text" placeholder="Restaurants">
28         <div>
29             <label for="idSlectCategory" class="text-white">Catégories: </label>
30             <select id="idSelectCategory" onchange="categoryChanged(); changed()">
31                 <option value=""></option>
32             </select>
33         </div>
34         <div class="select-editable">
35             <select id="idSelectRadius"
36                 onchange="this.nextElementSibling.value=this.value; radiusChanged();
37                 changed();">
38                 <option value="1 km">1 km</option>
39                 <option value="3 km">3 km</option>
40                 <option value="5 km" selected>5 km</option>
41                 <option value="8 km">8 km</option>
42                 <option value="10 km">10 km</option>
43                 <option value="20 km">20 km</option>
44             </select>
45             <input id="idInputRadius" type="text" name="format" value="5 km"
46                 onchange="radiusChanged(); changed();" />
47         </div>
48     </div>
49     <div class="main-restaurants">
50         <div class="list-container">
51             <div id="idListDisplay"></div>
52         </div>
53         <div class="map-container">
```

```
51         <div id="map"></div>
52     </div>
53 </div>
54 <footer>
55     <p>&copy; 2022 GrinEat, developped by <a href="https://github.com/miyokidev"
class="text-white">@miyokidev</a>.
56     </p>
57 </footer>
58 <script src="./assets/js/restaurants.js"></script>
59 </body>
60
61 </html>
```

c:\grin-eat-tpi2022\grineat-client\assets\css\style.css

```
1 * {
2     /*padding: 0;*/
3     margin: 0;
4 }
5
6 :root {
7     --bs-font-sans-serif: system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue",
    Arial, "Noto Sans", "Liberation Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji",
    "Segoe UI Symbol", "Noto Color Emoji";
8     --bs-body-font-family: var(--bs-font-sans-serif);
9     --bs-body-font-size: 1rem;
10    --bs-body-font-weight: 400;
11    --bs-body-line-height: 1.5;
12    --bs-white-rgb: 255, 255, 255;
13    --bs-greenBg-rgb: 25, 135, 84;
14 }
15
16 body {
17     height: 100%;
18     font-family: var(--bs-body-font-family);
19     font-size: var(--bs-body-font-size);
20     font-weight: var(--bs-body-font-weight);
21     line-height: var(--bs-body-line-height);
22     text-shadow: 0 .05rem .1rem rgba(0, 0, 0, .5);
23     --bs-bg-opacity: 1;
24     background-color: rgba(var(--bs-greenBg-rgb), var(--bs-bg-opacity)) !important;
25     box-shadow: inset 0 0 5rem rgba(0, 0, 0, .5);
26     overflow: hidden;
27 }
28
29 header {
30     width: 100%;
31     height: 60px;
32     display: flex;
33     /* align horizontal */
34     justify-content: center;
35     /* align vertical */
36     align-items: center;
37     user-select: none;
38     flex-wrap: wrap;
39 }
40
41 header
42
43 .header-logo-start {
44     /* align horizontal */
45     justify-content: start;
46 }
47
48 .header-logo-start h3 {
49     margin-left: 1rem;
50 }
51
52 #idLogo {
```



```
53     position: absolute;
54     left: 10px;
55     cursor: pointer;
56 }
57
58 .address-display {
59     cursor: pointer;
60 }
61
62 .shadow-header {
63     box-shadow: 0 4px 2px -2px rgba(0, 0, 0, .2);
64 }
65
66 h1,
67 h2,
68 h3,
69 h4,
70 h5,
71 h6 {
72     font-size: 1.75rem;
73     font-weight: 500;
74     line-height: 1.2;
75 }
76
77 .text-white {
78     --bs-text-opacity: 1;
79     color: rgba(var(--bs-white-rgb), var(--bs-text-opacity)) !important;
80 }
81
82 /* If the screen size is 601px wide or more, set the font-size of <div> to 80px */
83 @media screen and (min-width: 661px) {
84     .main {
85         height: calc(100vh - (60px + 50px));
86         display: flex;
87         justify-content: center;
88         /* align horizontal */
89         align-items: center;
90         /* align vertical */
91     }
92
93     .search-container {
94         width: 60%;
95         text-align: center !important;
96         padding-left: 1rem;
97         padding-right: 1rem;
98     }
99
100     .search-container h3 {
101         font-size: 200%;
102         margin-bottom: 1rem;
103     }
104
105     .btn-search {
106         background-color: rgb(1, 114, 52);
107         border: none;
```

```
108     font-weight: 700 !important;
109     padding: .375rem .75rem;
110     border-radius: .25rem;
111     transform: scale(1.25);
112     flex: 0;
113 }
114
115 .mapboxgl-ctrl-geocoder {
116     flex: 1;
117     transform: scale(1.25);
118     min-width: 100%;
119 }
120
121 .main-restaurants {
122     min-width: 100%;
123     height: calc(100vh - (60px + 30px + 50px));
124     display: flex;
125 }
126 }
127
128 /* If the screen size is 600px wide or less, set the font-size of <div> to 30px */
129 @media screen and (max-width: 600px) {
130     .main {
131         height: calc(100vh - (60px + 50px));
132         display: flex;
133         justify-content: center;
134         /* align horizontal */
135         align-items: center;
136         /* align vertical */
137     }
138
139     .search-container {
140         width: 60%;
141         text-align: center !important;
142         padding-left: 1rem;
143         padding-right: 1rem;
144     }
145
146     .search-container h3 {
147         font-size: 150%;
148         margin-bottom: 0.5rem;
149     }
150
151     .btn-search {
152         background-color: rgb(1, 114, 52);
153         border: none;
154         font-weight: 700 !important;
155         padding: .375rem .75rem;
156         border-radius: .25rem;
157         transform: scale(1.25);
158         flex: 0;
159     }
160
161     .mapboxgl-ctrl-geocoder {
162         flex: 1;
```

```
163     transform: scale(0.75);
164     width: 300px !important;
165 }
166
167 .main-restaurants {
168     height: calc(100vh - (60px + 30px + 50px));
169     display: flex;
170     flex-wrap: wrap-reverse;
171 }
172
173 .map-container {
174     height: 400px;
175 }
176 }
177
178 .btn-search:hover {
179     background-color: rgb(0, 189, 85);
180 }
181
182 .btn-search:enabled {
183     cursor: pointer;
184 }
185
186 .btn-search:disabled {
187     background-color: #6c757d;
188     opacity: .65;
189     pointer-events: none;
190 }
191
192 .search-inputs-container {
193     display: flex;
194     justify-content: center;
195     flex-wrap: wrap;
196     gap: 10px 40px;
197 }
198
199 footer {
200     height: 50px;
201     display: flex;
202     justify-content: center;
203     /* align horizontal */
204     align-items: center;
205     /* align vertical */
206     color: rgba(255, 255, 255, .5) !important;
207 }
208
209 .tools-container {
210     height: 40px;
211     display: flex;
212     flex-wrap: wrap;
213     gap: 10px 30px;
214     display: flex;
215     align-items: center;
216     /* align vertical */
217 }
```

```
218
219 .list-container {
220     flex: 1 1 15%;
221     background-color: white;
222     min-height: 100%;
223     border: solid 2px black;
224 }
225
226 #idListDisplay {
227     height: 100%;
228     overflow-y: scroll;
229 }
230
231 .card {
232     width: 100%;
233     height: 280px;
234     display: flex;
235     flex-direction: column;
236     align-items: center;
237     border-bottom: solid 2px black;
238 }
239
240 .card img {
241     object-fit: cover;
242     height: 70%;
243     min-width: 100%;
244 }
245
246 .card-body {
247     display: flex;
248     flex-direction: column;
249     width: 100%;
250     height: 30%;
251     justify-content: center;
252     align-items: center;
253 }
254
255 .card p {
256     margin: 5px 0px 5px 0px;
257 }
258
259 .card button {
260     width: 30%;
261     height: 40%;
262     background-color: #0d6efd;
263     border: none;
264     font-weight: 700 !important;
265     padding: .375rem .75rem;
266     border-radius: .25rem;
267     --bs-text-opacity: 1;
268     color: rgba(var(--bs-white-rgb), var(--bs-text-opacity)) !important;
269     cursor: pointer;
270 }
271
272 .card button:hover {
```

```
273     background-color: #0b5ed7;
274     border-color: #0a58ca;
275 }
276
277 .map-container {
278     flex: 1 1 85%;
279     border: solid 2px black;
280     border-left: none;
281 }
282
283 #map {
284     min-height: 100%;
285 }
286
287 #idSearchName {
288     font-size: 16px;
289     height: 20px;
290     margin-left: 1rem;
291     padding: 1px 4px 1px 35px;
292     background: transparent url("../images/search.png") no-repeat 8px center;
293     background-size: 20px;
294     background-color: white;
295     border-radius: .50rem;
296 }
297
298 #idSearchName:focus {
299     outline: none;
300     border: solid 2px;
301     border-color: rgb(1, 114, 52);
302 }
303
304 #idSelectCategory {
305     font-size: 16px;
306     height: 25px;
307 }
308
309 /* EDITABLE SELECT */
310
311 .select-editable {
312     position: relative;
313     background-color: white;
314     border: solid grey 1px;
315     width: 120px;
316     height: 18px;
317 }
318
319 .select-editable select {
320     position: absolute;
321     top: 0px;
322     left: 0px;
323     border: none;
324     font-size: 16px;
325     width: 120px;
326     margin: 0;
327 }
```

```
328
329 .select-editable input {
330     position: absolute;
331     top: 0px;
332     left: 0px;
333     width: 100px;
334     padding: 1px;
335     font-size: 14px;
336     border: none;
337 }
338
339 .select-editable select:focus,
340 .select-editable input:focus {
341     outline: none;
342 }
343
344 /* END EDITABLE SELECT */
345
346 .main-menus {
347     height: calc(100vh - (60px + 50px));
348     margin-top: 5px;
349 }
350
351 .restaurant-info-container {
352     width: 100%;
353     height: 40%;
354     display: flex;
355     gap: 100px;
356     flex-wrap: wrap;
357     border-bottom: solid 2px rgb(1, 114, 52);
358     padding: 5px;
359 }
360
361 .restaurant-info-container img {
362     object-fit: cover;
363     width: 20%;
364     height: 300px;
365 }
366
367 .restaurant-name-city {
368     padding: 1rem;
369     width: auto;
370     display: flex;
371     flex-direction: column;
372     gap: 50px;
373 }
374
375 .restaurant-name-city p {
376     font-size: 1.25rem;
377 }
378
379 .restaurant-street-number {
380     padding: 1rem;
381     width: auto;
382     display: flex;
```

```
383     gap: 20px;
384     flex-direction: column;
385 }
386
387 .menu-items-container {
388     height: 60%;
389     overflow-y: scroll;
390     border-bottom: solid 2px rgb(1, 114, 52);
391 }
392
393 #idListMenuItems {
394     display: flex;
395     justify-content: space-between !important;
396     flex-wrap: wrap;
397 }
398
399 .card-menu {
400     flex: 1 0 calc(33% - 2rem);
401     max-width: calc(33% - 2rem);
402     background-color: white;
403     z-index: 0;
404     padding: .5rem;
405     margin: 1rem .5rem;
406     display: flex;
407     flex-direction: row;
408     flex-wrap: wrap;
409 }
410
411 .card-menu-body {
412     width: 65%;
413 }
414
415 .card-menu img {
416     width: 35%;
417 }
418
419 .card-menu-description {
420     font-size: .75rem;
421 }
```

c:\grin-eat-tpi2022\grineat-client\assets\js\main.js

```

1  /*
2      /\  /\
3      /\ \_/\ /\
4      \_  _/  /  /  Nom et Prénom: GRIN Brian
5      / * * \  /^^^] Enseignant : Monsieur Antoine Schmid
6      \_ \0/_/  [  ] Classe : I.DA-P4B
7      /  _/_/  [  / Date : 18.05.2022
8      \  _/_/  /  / Nom du projet : GrinEat-Client
9      [ [ / \_/_/ Version du projet : 1.0
10     _[ [ \ /_/_/ Cours : TPI
11
12 */
13
14 // Création de l'autocomplétion qu'on récupère grâce au token publique de MapBox
15 mapboxgl.accessToken =
16 'pk.eyJ1IjoibWl5b2tpIiwiaSI6ImNsMW96ejJ5NTAzMjQza3B0NHB3bHMxYncifQ.03AlBHWNd8MiauuZz_sSNQ';
17 const geocoder = new MapboxGeocoder({
18     accessToken: mapboxgl.accessToken,
19     types: 'address'
20 });
21
22 // Récupération des éléments de la page
23 const searchBarContainer = document.getElementById('idSearchBarContainer'); // Conteneur du
24 // champ de recherche
25 const btnSearch = document.getElementById('idBtnSearch'); // Bouton pour effectuer la
26 // recherche
27 let address = {}; // objet address qui prendra en propriété le resultat de l'autocomplete ou
28 // la saisie de l'utilisateur dans le champ
29
30 // Evenement lors du chargement de la page
31 addEventListener("DOMContentLoaded", () => {
32     btnSearch.disabled = true; // On désactive par défaut le bouton rechercher lors du
33     // chargement car le champ de saisi est vide
34     document.getElementsByClassName("mapboxgl-ctrl-geocoder--input")
35     [0].setAttribute("placeholder", "Ex: Quai Capo D'istria 9, 1205 Genève"); // On ajoute un
36     // placeholder au champ crée par MapBox pour l'autocomplétion
37 });
38
39 geocoder.addTo('#idSearchBarContainer'); // Ajout du champ d'autocomplétion dans le conteneur
40
41 // Evenement se lançant lorsque l'utilisateur clique sur une proposition de l'autocomplétion
42 geocoder.on('result', (e) => {
43     address.coordinates = {latitude : e.result.center[1], longitude : e.result.center[0]};
44     address.street = e.result.place_name;
45 });
46
47 // Evenement lancé lorsque l'utilisateur clique sur le bouton recherché
48 btnSearch.addEventListener("click", () => {
49     // On vérifie si l'objet address contient des informations
50     if (Object.keys(address).length !== 0) {
51         sessionStorage.setItem('address', JSON.stringify(address)); // Sauvegarde en session
52         // de l'adresse de la recherche pour s'en servir sur la page suivante
53         location.href = "restaurants.html";
54     } else {
55         alert("Adresse invalide");
56     }
57 });

```



```
48     }
49 });
50
51 // Evenement lancé à chaque fois que l'utilisateur relâche une touche du clavier sur le champ
52 searchBarContainer.addEventListener('keyup', () => {
53     var searchBar = document.getElementsByClassName("mapboxgl-ctrl-geocoder--input")[0];
54
55     address.coordinates = null;
56     address.street = searchBar.value;
57
58     if (searchBar.value === "") {
59         btnSearch.disabled = true;
60     } else {
61         btnSearch.disabled = false;
62     }
63 });
```

c:\grin-eat-tpi2022\grineat-client\assets\js\menu.js

```

1  /*
2      /\  /\
3      /\ \_/\ /\
4      \_  _/  /_  /  Nom et Prénom: GRIN Brian
5      / * * \  /^^^]  Enseignant : Monsieur Antoine Schmid
6      \_ \0/ _/ [ ]   Classe : I.DA-P4B
7      /   \_ [ ]     Date : 18.05.2022
8      \   \_ / /     Nom du projet : GrinEat-Client
9      [ [ / \_/_/    Version du projet : 1.0
10     _[ [ \ /_/_/    Cours : TPI
11
12  */
13
14  const logo = document.getElementById("idLogo"); // Logo du nav
15  // Variables pour la récupération de l'id du restaurant en paramètre dans l'URL
16  const queryString = window.location.search;
17  const urlParams = new URLSearchParams(queryString);
18  // Variable des différents éléments pour l'affichage des informations
19  const image = document.getElementById("idImage");
20  const nameRestaurant = document.getElementById("idName");
21  const cp = document.getElementById("idCp");
22  const city = document.getElementById("idCity");
23  const street = document.getElementById("idStreet");
24  const phone = document.getElementById("idPhone");
25  const categories = document.getElementById("idCategories");
26
27  // Rediriger vers l'index si on clique sur le logo.
28  logo.addEventListener("click", () => {
29      sessionStorage.removeItem('address');
30      location.href = "index.html";
31  });
32
33  addEventListener("DOMContentLoaded", () => {
34      if (sessionStorage.getItem('address') == null) {
35          location.href = "index.html";
36      } else {
37          let idRestaurant = urlParams.get('idRestaurant'); // Récupération de la valeur donné
dans l'URL
38          fetch(`http://localhost/grin-eat-tpi2022/grineat-api/restaurants/${idRestaurant}
/menus`)
39              .then(function (response) {
40                  return response.json()
41              }).then(function (json) {
42                  displayInfoMenuItems(json.result);
43              }).catch(function (ex) {
44                  console.log(ex);
45              });
46      }
47  });
48
49  // Fonction appelée en callback après la récupération des données avec fetch
50  function displayInfoMenuItems(restaurant) {
51      // On assigne pour tout les éléments de la description les données du restaurant
52      image.setAttribute("src", restaurant.info.image);

```

```
53     nameRestaurant.innerText = restaurant.info.name;
54     cp.innerText = restaurant.info.cp;
55     city.innerText = restaurant.info.city;
56     street.innerText = restaurant.info.street;
57     phone.innerText = restaurant.info.phone;
58     // On affiche que les 2 premières catégories du restaurant
59     for (let j = 0; j < restaurant.info.categories.length; j++) {
60         if (j < 2) {
61             let category = document.createElement('p');
62             category.innerText = restaurant.info.categories[j];
63             categories.appendChild(category);
64         } else {
65             break;
66         }
67     }
68
69     // On remplit la liste des menus
70     document.getElementById("idListMenuItems").innerHTML = "";
71     for (let i = 0; i < restaurant.menu_items.length; i++) {
72         let card = document.createElement('div');
73         card.setAttribute('id', `item${restaurant.menu_items[i].id}`);
74         card.classList.add("card-menu");
75         card.innerHTML = `
76         <div class="card-menu-body">
77             <p>${restaurant.menu_items[i].name}</p>
78             <p class="card-menu-description">${restaurant.menu_items[i].description}</p>
79             <p>${restaurant.menu_items[i].price} CHF</p>
80         </div>`;
81         document.getElementById("idListMenuItems").appendChild(card);
82     }
83 }
```

c:\grin-eat-tpi2022\grineat-client\assets\js\restaurants.js

```

1  /*
2      /\      /\
3      /\ \_/\ /\
4      \_ \_/\ /  /  Nom et Prénom: GRIN Brian
5      / * * \ /^^^] Enseignant : Monsieur Antoine Schmid
6      \_ \_/\ [ ] Classe : I.DA-P4B
7      / \_ \_ [ / Date : 18.05.2022
8      \ \_ \_ / / Nom du projet : GrinEat-Client
9      [ [ / \_ \_ Version du projet : 1.0
10     _[ [ \ \_ \_ Cours : TPI
11
12 */
13
14 const search = {}; // objet qu'on envoie dans le body lors des appels au serveur
15 search.categories = [];
16 const addressDisplay = document.getElementById("idAddress"); // Adresse affichée dans le nav
17 const searchName = document.getElementById("idSearchName");
18 const logo = document.getElementById("idLogo"); // Logo du nav
19 let address = {}; // objet address contenant la rue et les coordonnées de l'adresse sélectionn
l'utilisateur
20 let restaurants = []; // Tableau des restaurants récupérés du serveur
21 let categories = []; // Tableau des catégories récupérés du serveur
22 let map;
23 let markers;
24
25 var iconHome = L.icon({
26     iconUrl: './assets/images/home.png',
27     iconSize: [41, 41], // size of the icon
28     iconAnchor: [20, 41]
29 });
30
31 var iconRestaurant = L.icon({
32     iconUrl: './assets/images/restaurant.png',
33     iconSize: [41, 41], // size of the icon
34     iconAnchor: [20, 41]
35 });
36
37
38 addEventListener("DOMContentLoaded", () => {
39     if (sessionStorage.getItem('address') == null) {
40         location.href = "index.html";
41     } else {
42         address = JSON.parse(sessionStorage.getItem('address'));
43
44         // On fait une promesse pour d'abord récupérer les catégories données par l'API pour
select quand c'est terminé on resolve pour effectuer la suite
45         new Promise((resolve, reject) => {
46             fetch('http://localhost/grin-eat-tpi2022/grineat-api/categories')
47                 .then(function (response) {
48                     return response.json()
49                 }).then(function (json) {
50                     categories = json.result;
51                     var select = document.getElementById("idSelectCategory");
52

```

```
53         // Remplir la liste déroulante de catégories.
54         for (let i = 0; i < categories.length; i++) {
55             var option = document.createElement('option');
56             option.setAttribute('value', categories[i].id);
57             option.appendChild(document.createTextNode(categories[i].nameFrench));
58             select.appendChild(option);
59         }
60         resolve();
61     }).catch(function (ex) {
62         reject(ex);
63     })
64 }).then(() => {
65     search.address = address.street;
66     if (address.coordinates != null) {
67         if (address.coordinates.latitude != null && address.coordinates.longitude != null) {
68             search.coordinates = address.coordinates;
69         }
70     }
71     radiusChanged();
72     sendData(displayMapList, "http://localhost/grin-eat-tpi2022/grineat-api/restaurant");
73 });
74 }
75 });
76
77 // Afficher un modal pour modifier son adresse quand on clique sur l'adresse.
78 addressDisplay.addEventListener("click", () => {
79     flyToMarker(address.coordinates);
80 });
81
82 // Rediriger vers l'index si on clique sur le logo.
83 logo.addEventListener("click", () => {
84     sessionStorage.removeItem('address');
85     location.href = "index.html";
86 });
87
88 // A chaque fois que l'utilisateur relâche une touche la variable search prend en valeur la valeur actuelle et
89 // effectue un appel au serveur
90 searchName.addEventListener("keyup", () => {
91     search.name = searchName.value;
92     changed();
93 });
94
95 // Fonction pour appeler le serveur API cette fonction est appelée à chaque fois qu'on filtre ou qu'on recharge la
96 // page quand on arrive sur la page
97 async function sendData(successCallBack, link, obj) {
98     fetch(link, { method: 'POST', body: JSON.stringify(obj) }).then(function (response) {
99         response.json().then(function (myJson) {
100             // On vérifie si les données qu'on a déjà sont similaires à ceux qu'on reçoit pour
101             // recharger toute la page si le contenu est le même
102             if (JSON.stringify(restaurants) != JSON.stringify(myJson.result.restaurants) ||
103                 restaurants.length == 0) {
104                 restaurants = myJson.result.restaurants;
105                 if (address != myJson.result.address) {
106                     address = myJson.result.address;
107                     sessionStorage.setItem('address', JSON.stringify(address));
108                 }
109             }
110         });
111     });
112 }
```

```
105 // Méthode similaire à un pattern Singleton si c'est la première fois qu'on a
méthode on crée la carte interactive
106 if (map == null) {
107     map = L.map('map').setView([address.coordinates.latitude,
address.coordinates.longitude], 15); // Map centrée sur l'adresse de l'utilisateur
108
109     L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}
/{y}?access_token=pk.eyJ1IjoibWl5b2tpIiwiaSI6ImNsMW96ejJ5NTAzMjQza3B0NHB3bHMxYncifQ.03AlBHWNd8
{
110         maxZoom: 18,
111         id: 'mapbox/streets-v11',
112         tileSize: 512,
113         zoomOffset: -1,
114     }).addTo(map);
115 } else {
116     // Pour supprimer les marqueurs
117     map.removeLayer(markers);
118 }
119 // Pour supprimer les restaurants de la liste
120 document.getElementById("idListDisplay").innerHTML = "";
121 addressDisplay.innerText = address.street != null ? address.street : "";
122 successCallBack();
123 }
124 });
125 });
126 }
127
128 // Méthode pour ajouter les marqueurs à la carte interactive et ajouter les restaurants dans l
dynamique
129 function displayMapList() {
130     // Placer les marqueurs et remplir la liste de restaurants affichée.
131     markers = L.featureGroup().addTo(map); // Groupe de marqueurs
132     // On ajoute un marqueur à l'adresse de l'utilisateur
133     L.marker([address.coordinates.latitude, address.coordinates.longitude], { icon: iconHome
}).addTo(markers);
134     for (let j = 0; j < restaurants.length; j++) {
135         // Remplir la liste
136         addToListDisplay(restaurants[j]);
137
138         // Créer le marqueur et l'ajouter au groupe de marqueurs
139         L.marker([restaurants[j].latitude, restaurants[j].longitude], { icon: iconRestaurant
}).addTo(markers).on('click', function(e) {
140             let list = document.getElementById("idListDisplay");
141             let card = document.getElementById(`restaurant${restaurants[j].id}`);
142
143             list.scrollTop = card.offsetTop;
144         });
145     }
146
147     map.flyToBounds(markers.getBounds().pad(0.1), { duration: 1 }); // On recentre la carte p
marqueur le plus éloigné visible
148 }
149
150 // Fonction pour ajouter dans la liste dynamique les restaurants
151 function addToListDisplay(restaurant) {
152     let card = document.createElement('div');
```

```
153 card.setAttribute('id', `restaurant${restaurant.id}`);
154 card.classList.add("card");
155 card.onclick = (event) => {
156     const isButton = event.target.nodeName === 'BUTTON';
157     if (isButton) {
158         location.href = `menu.html?idRestaurant=${restaurant.id}`;
159     } else {
160         flyToMarker(restaurant);
161     }
162 };
163 card.innerHTML = `![${restaurant.name}'s image](${restaurant.image})
```