































































































































































































































































































































































































































































































































































Name	△ Date Modified Size		Kind
∨ 🛅 coverage	Today at 20:12		Folder
in clover.xml	Today at 20:12	2 KB	XML Document
coverage-final.json	Today at 20:12	3 KB	JSON Document
∨ is loov-report	Today at 20:12		Folder
base.css	Today at 20:12	5 KB	CSS style sheet
block-navigation.js	Today at 20:12	3 KB	JavaScript
🖂 favicon.png	Today at 20:12 44	5 bytes	PNG image
index.html	Today at 20:12	4 KB	HTML text
e merge.js.html	Today at 20:12	7 KB	HTML text
prettify.css	Today at 20:12 67	6 bytes	CSS style sheet
prettify.js	Today at 20:12	18 KB	JavaScript
sort-arrow-sprite.png	Today at 20:12 13	8 bytes	PNG image
sorter.js	Today at 20:12	6 KB	JavaScript
lcov.info	Today at 20:12 38	6 bytes	Document
merge.js	Today at 18:28	1 KB	JavaScript
merge.test.js	Today at 18:28	1 KB	JavaScript
> iii node_modules	Today at 18:40		Folder
package-lock.json	Today at 18:40	153 KB	JSON Document
package.json	Today at 20:12 37	9 bytes	JSON Document

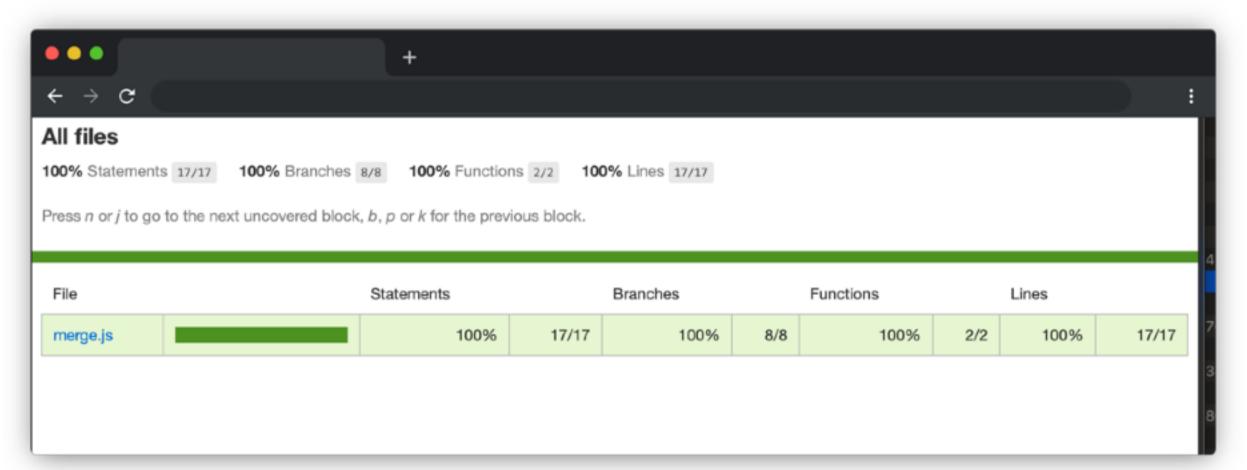


All files merge.js

100% Statements 17/17 100% Branches I/N 100% Functions 2/2 100% Lines 17/17

Press n or j to go to the next uncovered block, b, p or k for the previous block.

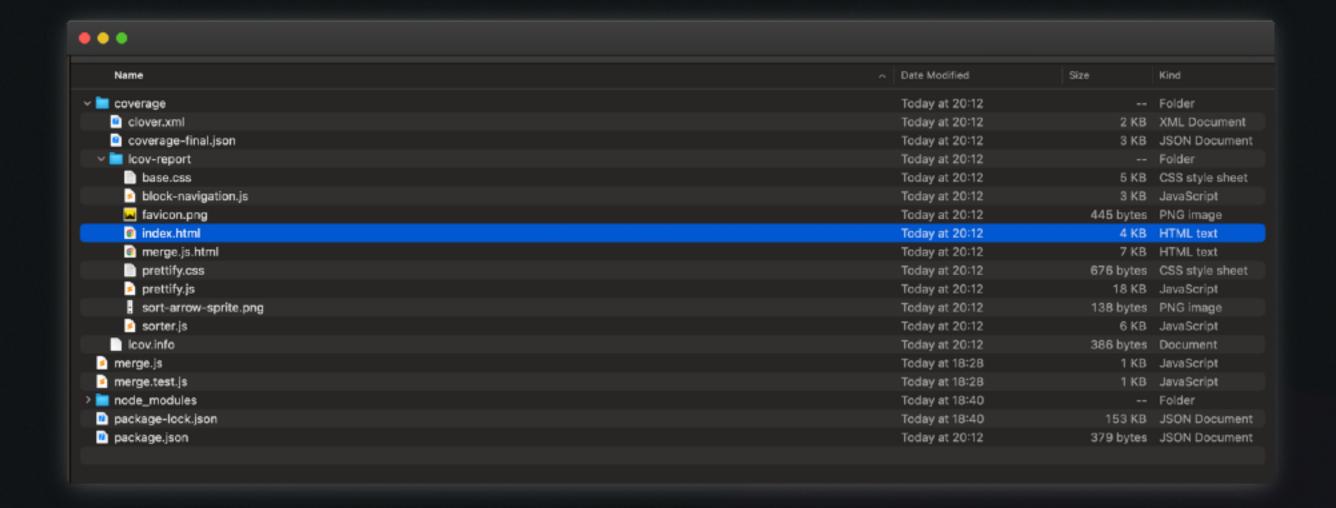
```
function merge(list1, list2) {
 2 51x
         if (list1.length < 1) (
           // List 1 is empty
 4 28×
           return list2;
 5 31x
         } else if (list2.length < 1) {
           // List 2 is empty
 7 1x
           return list1:
          } else {
            war first, rest, other;
30 30x
            if (list1[0] < list2[0]) {
11
             // First element of list1 is smaller
             first = list1.slice(0, 1);
12 18x
13 18x
             rest = list1.slice(1, list1.length);
14 18x
             other = list2;
15
           ) wise (
16
             // First element of list2 is smaller
37 12x
             first = list2.slice(0, 1):
             rest = list2.slice(1, list2.length);
18 12×
39 12x
             other = list1;
28
           3
71 36x
            return first.comcat(merge(rest, other)); // Recursively merge the rest of the lists
22
23
24
25
        // Splits the list for merge sort
25
        function mergeSort(array_list) (
27 41K
         if [array_list.length <= 1) {
28 24×
            return erray_list;
29
          } else {
30 17x
           return merge(
31
             mergeSort(array_list.slice(0, array_list.length / 2)),
12
              nergeSort(array_list.slice(array_list.length / 2, array_list.length))
33
           3:
34
35
36
37 1x module.experts.merge = merge;
38 1x module.exports.mergeSort = mergeSort;
39
```

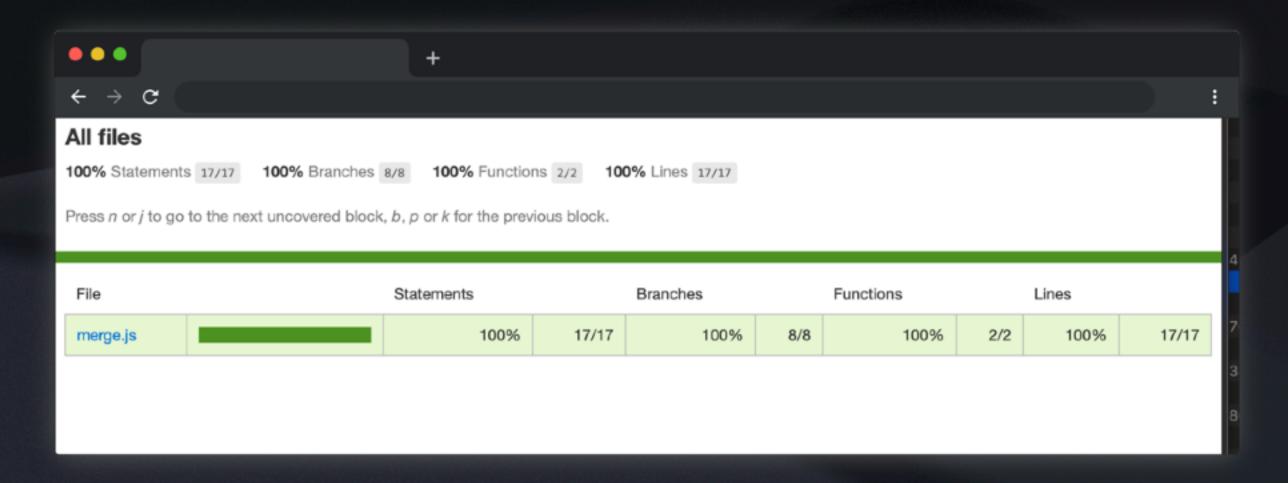






Accessing the Coverage Report





```
• • •
← → C
All files merge.js
100% Statements 17/17 100% Branches I/II 100% Functions 2/2 100% Lines 17/12
  Press n or j to go to the next uncovered block, b, p or k for the previous block.
            function merge(list1, list2) {
            if (list1.length < 1) {
   // List 1 is empty
   return list2;</pre>
              } else if (list2.length < 1) {
                return list1;
               if (list10) < list200) {
  // First element of list1 is smaller
  first = list1.slice(0, 1);</pre>
                  other = list2;
                  rest = list2.slice(1, list2.length);
                 return first.comcat(merge(rest, other)); // Recursively merge the rest of the lists
           function mergeSort(array_list) {
  if (array_list.length <= 1) {</pre>
                   nergeSort(array list.slice(8, array list.length / 21).
   37 1x module.exports.merge = merge;
```

```
include monopole (racry, list) {
   if incry, list.length on it {
      return array, list;
} else {
      mergeSort(array, list.slice(8, array, list.length / 2);
      mergeSort(array, list.slice(array, list.length / 2, array, list.length))
      };
}

ix module.exports.merge = merge;

ix module.exports.mergeSort = mergeSort;
```