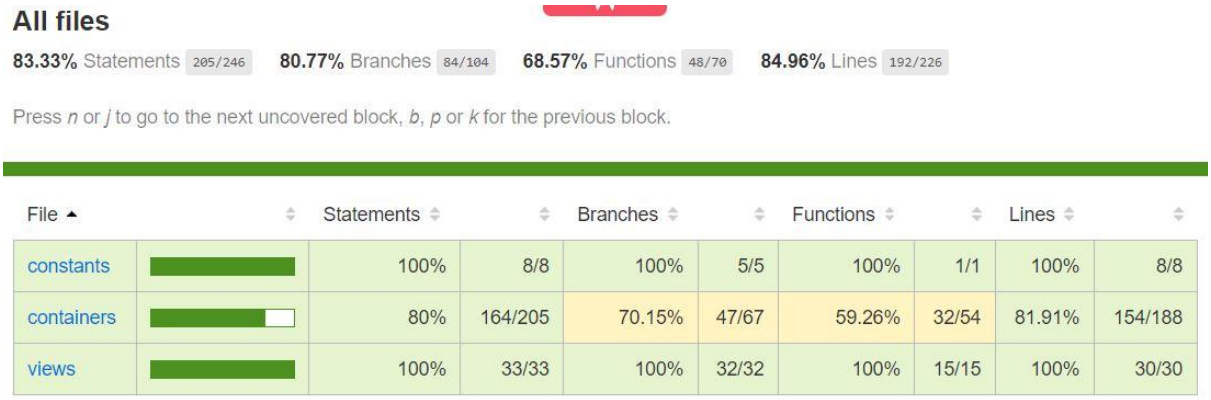
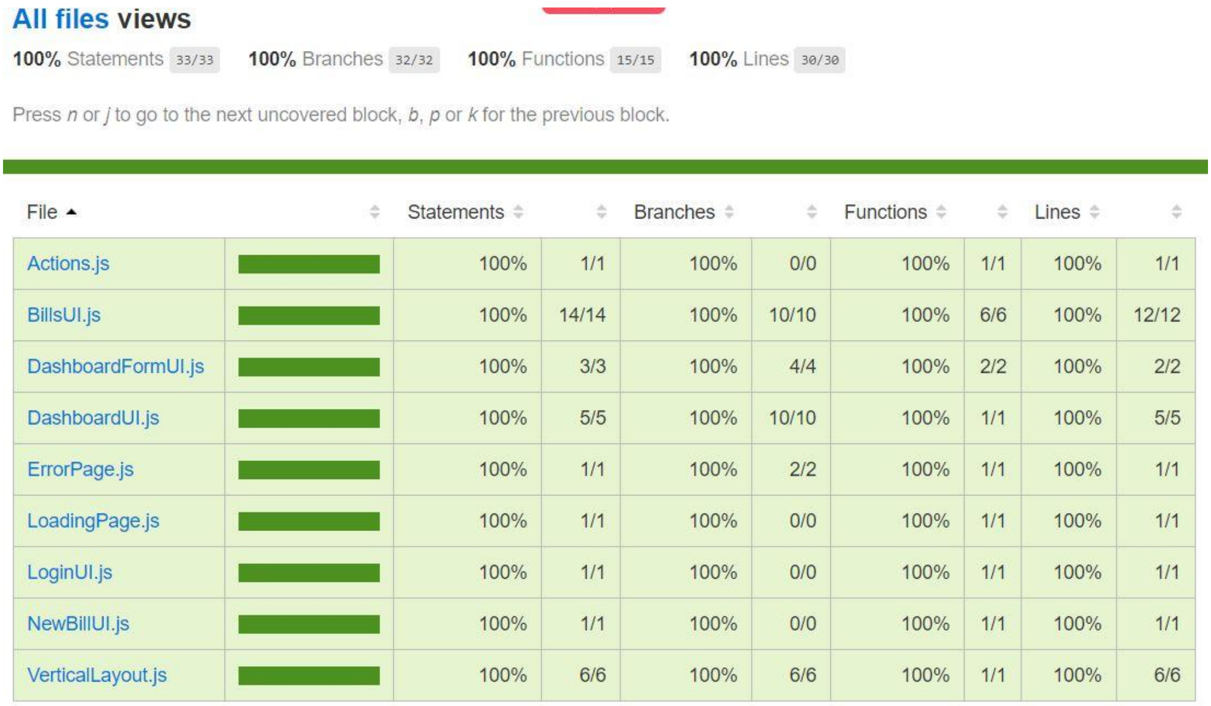


All files

83.33% Statements 205/246 **80.77%** Branches 84/104 **68.57%** Functions 48/70 **84.96%** Lines 192/226

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

File ▲		Statements		Branches		Functions		Lines	
constants		100%	8/8	100%	5/5	100%	1/1	100%	8/8
containers		80%	164/205	70.15%	47/67	59.26%	32/54	81.91%	154/188
views		100%	33/33	100%	32/32	100%	15/15	100%	30/30

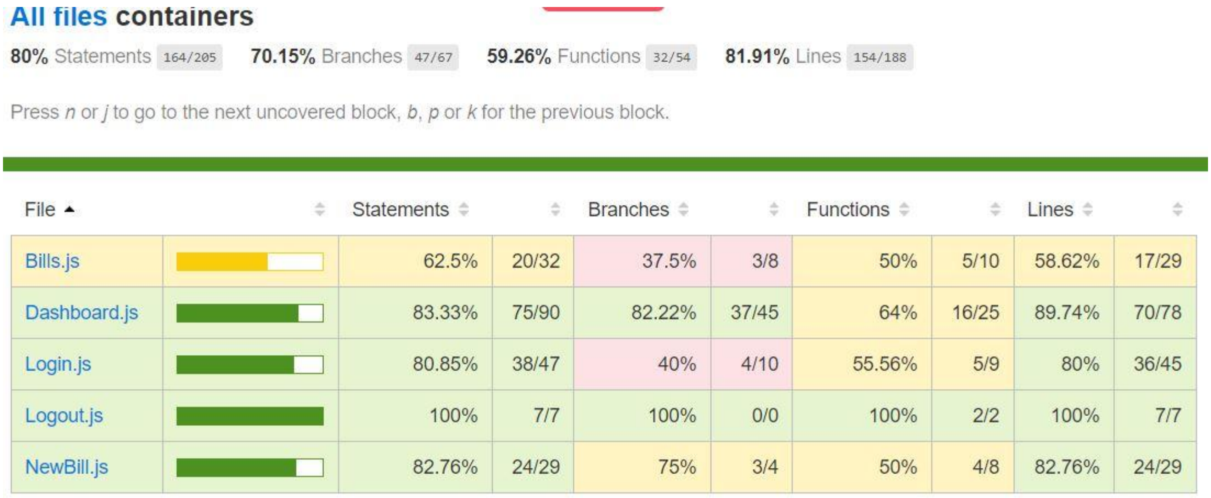


All files views

100% Statements 33/33 **100%** Branches 32/32 **100%** Functions 15/15 **100%** Lines 30/30

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

File ▲		Statements		Branches		Functions		Lines	
Actions.js		100%	1/1	100%	0/0	100%	1/1	100%	1/1
BillsUI.js		100%	14/14	100%	10/10	100%	6/6	100%	12/12
DashboardFormUI.js		100%	3/3	100%	4/4	100%	2/2	100%	2/2
DashboardUI.js		100%	5/5	100%	10/10	100%	1/1	100%	5/5
ErrorPage.js		100%	1/1	100%	2/2	100%	1/1	100%	1/1
LoadingPage.js		100%	1/1	100%	0/0	100%	1/1	100%	1/1
LoginUI.js		100%	1/1	100%	0/0	100%	1/1	100%	1/1
NewBillUI.js		100%	1/1	100%	0/0	100%	1/1	100%	1/1
VerticalLayout.js		100%	6/6	100%	6/6	100%	1/1	100%	6/6

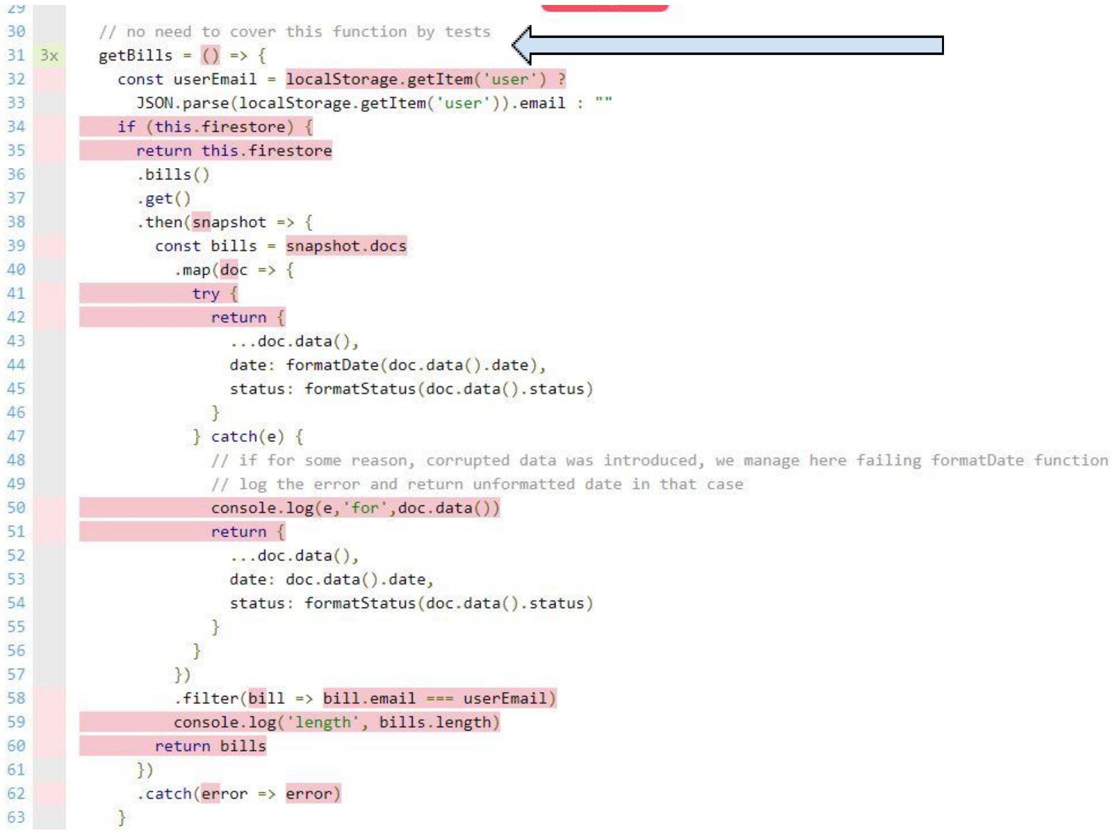


All files containers

80% Statements 164/205 **70.15%** Branches 47/67 **59.26%** Functions 32/54 **81.91%** Lines 154/188

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

File ▲		Statements		Branches		Functions		Lines	
Bills.js		62.5%	20/32	37.5%	3/8	50%	5/10	58.62%	17/29
Dashboard.js		83.33%	75/90	82.22%	37/45	64%	16/25	89.74%	70/78
Login.js		80.85%	38/47	40%	4/10	55.56%	5/9	80%	36/45
Logout.js		100%	7/7	100%	0/0	100%	2/2	100%	7/7
NewBill.js		82.76%	24/29	75%	3/4	50%	4/8	82.76%	24/29



```
  // no need to cover this function by tests
  getBills = () => {
    const userEmail = localStorage.getItem('user') ?
      JSON.parse(localStorage.getItem('user')).email : ""
    if (this.firestore) {
      return this.firestore
      .bills()
      .get()
      .then(snapshot => {
        const bills = snapshot.docs
          .map(doc => {
            try {
              return {
                ...doc.data(),
                date: formatDate(doc.data().date),
                status: formatStatus(doc.data().status)
              }
            } catch(e) {
              // if for some reason, corrupted data was introduced, we manage here failing formatDate function
              // log the error and return unformatted date in that case
              console.log(e,'for',doc.data())
              return {
                ...doc.data(),
                date: doc.data().date,
                status: formatStatus(doc.data().status)
              }
            }
          })
          .filter(bill => bill.email === userEmail)
          console.log('length', bills.length)
        return bills
      })
      .catch(error => error)
    }
```