



We can make the following check in our test:

```
it('sets data into local storage', () => {
  const jsonId = '222';
  const newJson = {data: 'json data'};
  setItem(jsonId, newJson);
  expect(localStorage.getItem(jsonId)).toBe(
    JSON.stringify(newJson)
  );
});
```

When we address localStorage in our test we in fact address our mockLocalStorage.

If we need our localStorage to contain some item before running test we can set item before calling function:

```
it('has data in local storage', () => {
  const jsonId = '123';
  const newJson = {data: 'json data'};
  window.localStorage.setItem(jsonId, JSON.stringify(newJson));
  // run function
});
```

Also after each test local storage must be cleared to avoid tests contaminate each other:

```
afterEach(() => {
  window.localStorage.clear();
});
```

Other VUE

[async-](#)[dynam](#)[event-](#)[v-deep](#)[vue-\\$c](#)[vue-cli](#)[vue-co](#)[vue-co](#)[vue-dc](#)[vue-fil](#)[vue-gl](#)

Other topics:

18 JULY 2020

Tracking
variables
inside timers

10 SEPTEMBER 2020

Jest
mockClear

