



16.01.2020 // Berlin.JS

Let's Test That Real Quick

Jan Stępień
janstepien.com

INNOQ

JAN STĘPIEŃ

Senior Consultant
@janstepien

**"If you liked it you
should have had
tests for it."**



**I want you to become
better at testing**

Why do we test?

Quality.

Quality.

Automation.

We automate builds
linting
deployment
and tests.



time/date

GARMIN

training

Today
6:18

11.83^{km}
836^{cal}

01:03:17
11.2^{km}

GPS

menu

lap / reset

```
function stepsPerHour(steps, seconds) {  
    return 60 * 60 * steps / seconds;  
}
```

```
describe('stepsPerHour', () => {  
  it('hopefully works', () => {  
    expect(stepsPerHour(1, 7200)).toBe(0.5);  
    expect(stepsPerHour(2, 7200)).toBe(1.0);  
    expect(stepsPerHour(0, 1800)).toBe(0.0);  
  });  
});
```

```
describe('stepsPerHour', () => {  
  it('hopefully works', () => {  
    [  
      [1, 7200, 0.5],  
      [2, 7200, 1.0],  
      [0, 1800, 0.0]  
    ].forEach(([steps, secs, speed]) => {  
      expect(stepsPerHour(steps, secs))  
        .toBe(speed)  
    })  
  })  
});
```

Ask Your

Domain Expert

$$\frac{a}{b} = \frac{a}{b}$$

$$\frac{a}{b} \cdot b = \frac{a}{b} \cdot b$$

$$\frac{a}{b} \cdot b = a$$

```
describe('stepsPerHour', () => {
  it('hopefully works', () => {
    [
      [1, 7200, 0.5],
      [2, 7200, 1.0],
      [0, 1800, 0.0]
    ].forEach(([steps, secs, speed]) => {
      expect(stepsPerHour(steps, secs))
        .toBe(speed)
    })
  })
});
```

```
describe('stepsPerHour', () => {
  it('hopefully works', () => {
    [
      [1, 7200],
      [2, 7200],
      [0, 1800]
    ].forEach(([steps, secs]) => {
      expect(stepsPerHour(steps, secs) * secs / 3600)
        .toBe(steps)
    })
  })
});
```



```
describe('stepsPerHour', () => {
  it('hopefully works', () => {
    [
      /*
       * arbitrary pairs of steps and seconds
       */
    ].forEach(([steps, secs]) => {
      expect(stepsPerHour(steps, secs) * secs / 3600)
        .toBe(steps)
    })
  })
});
```

Property-Based Testing

QuickCheck.

fast-check

github.com/dubzzz/fast-check

```
describe('stepsPerHour', () => {
  it('hopefully works', () => {
    [[1, 7200],
     [2, 7200],
     [0, 1800]].forEach(([steps, secs]) => {
      expect(stepsPerHour(steps, secs) * secs / 3600)
        .toBe(steps)
    })
  })
});
```

```
const fc = require('fast-check');

describe('stepsPerHour', () => {
  it('hopefully works', () => {
    fc.assert(fc.property(fc.integer(), fc.integer()),
      (steps, secs) =>
        stepsPerHour(steps, secs) * secs / 3600 === steps
    ));
  });
});
```

Failures:

1) stepsPerHour hopefully works

Message:

Error: Property failed after 25 tests

{ seed: 1986116297, path: "24:1:0:0:0:0", endOnFailure: true }

Counterexample: [1,0]

Shrunk 5 time(s)

Got error: Property failed by returning false

Hint: Enable verbose mode in order to have the list of all failing values encountered during the run

Failures:

1) stepsPerHour hopefully works

Message:

Error: Property failed after 25 tests

{ seed: 1986116297, path: "24:1:0:0:0:0", endOnFailure: true }

Counterexample: [1,0]

Shrunk 5 time(s)

Got error: Property failed by returning false

Hint: Enable verbose mode in order to have the list of all failing values encountered during the run


```
const fc = require('fast-check');
```

```
describe('stepsPerHour', () => {  
  it('hopefully works', () => {  
    fc.assert(fc.property(fc.integer(), fc.integer()),  
      (steps, secs) => stepsPerHour(steps, secs) * secs / 3600 === steps  
    ));  
  });  
});
```

```
const fc = require('fast-check');
```

```
describe('stepsPerHour', () => {  
  it('hopefully works', () => {  
    fc.assert(fc.property(fc.integer(), fc.integer().filter(s => s > 5),  
      (steps, secs) => stepsPerHour(steps, secs) * secs / 3600 === steps  
    ));  
  });  
});
```

```
const fc = require('fast-check');
```

```
describe('stepsPerHour', () => {  
  it('hopefully works', () => {  
    fc.assert(fc.property(fc.integer(), fc.integer(5, 10_000),  
      (steps, secs) => stepsPerHour(steps, secs) * secs / 3600 === steps  
    ));  
  });  
});
```

Failures:

1) stepsPerHour hopefully works

Message:

Error: Property failed after 2 tests

{ seed: -1073083424, path: "1:1:0:0:0:0:4:0:5", endOnFailure: true }

Counterexample: [1,342]

Shrunk 8 time(s)

Got error: Property failed by returning false

Hint: Enable verbose mode in order to have the list of all failing values encountered during the run

Failures:

1) stepsPerHour hopefully works

Message:

Error: Property failed after 2 tests

{ seed: -1073083424, path: "1:1:0:0:0:0:4:0:5", endOnFailure: true }

Counterexample: [1 342]

The screenshot shows a browser's developer console with the following elements:

- Toolbar:** Includes icons for close, copy, paste, refresh, and download. It also shows 11 files, 0 errors, and 0 warnings. A search bar is present on the right.
- Navigation:** A row of tabs for Elements, Network, Resources, Timelines, Debugger, Storage, Console, and Search. The Console tab is active.
- Console Filter:** A search input, navigation arrows, and filter buttons for "All", "Errors" (selected), "Warnings", and "Logs". There are also buttons for "Preserve Log", "Refresh", and "Clear".
- Message:** "Console opened at 11:33:04 AM"
- Log Entries:**
 - Input: `> 0.1 + 0.2`
 - Output: `< 0.30000000000000004 = $1`
 - Current prompt: `> |`

Automation.

Design.

```
function reverse(array) {  
    ...  
}
```

```
// and by array I mean  
// an immutable one
```



```
describe('reverse', () => {  
  it('reverses', () => {  
    fc.assert(fc.property(  
  
      ))  
    });  
  });  
});
```

```
describe('reverse', () => {  
  it('reverses', () => {  
    fc.assert(fc.property(fc.array(fc.integer()), 0, 10),  
  
    ))  
  });  
});
```

```
describe('reverse', () => {
  it('reverses', () => {
    fc.assert(fc.property(fc.array(fc.integer(), 0, 10),
      array => /* ??? */
    ))
  });
});
```

```
describe('reverse', () => {
  it('makes no difference when called twice', () => {
    fc.assert(fc.property(fc.array(fc.integer(), 0, 10),
      array => reverse(reverse(array)).equals(array)
    ))
  });
});
```

```
function reverse(array) {  
    ...  
}
```


```
function reverse(array) {  
    return array;  
}
```

```
describe('reverse', () => {  
  it('flips the first and the last element', () => {  
    fc.assert(fc.property(  
      ))  
    });  
  })  
})
```

```
describe('reverse', () => {  
  it('flips the first and the last element', () => {  
    fc.assert(fc.property(fc.array(fc.integer()), 1, 10),  
      array =>  
        reverse(array)[0] === array[array.length - 1]  
      ))  
  });  
})
```



```
describe('reverse', () => {
  it('flips the first and the last element', () => {
    fc.assert(fc.property(fc.array(fc.integer()), 1, 10),
      array =>
        reverse(array)[0] === array[array.length - 1])
  })
});
})
```



Failures:

1) reverse flips the first and the last element

Message:

Error: Property failed after 1 tests

{ seed: 1227040564, path: "0:1:0:0:0:1:0:0:0", endOnFailure: true }

Counterexample: [[0,1]]

Shrunk 35 time(s)

Got error: Property failed by returning false

Hint: Enable verbose mode in order to have the list of all failing values encountered during the run

Failures:

1) reverse flips the first and the last element

Message:

Error: Property failed after 1 tests

{ seed: 1227040564, path: "0:1:0:0:0:1:0:0:0", endOnFailure: true }

Counterexample: [[0,1]]

Shrunk 35 time(s)



Got error: Property failed by returning false

Hint: Enable verbose mode in order to have the list of all failing values encountered during the run

Failures:

1) reverse flips the first and the last element

Message:

Error: Property failed after 1 tests

{ seed: -1259617890, path: "0:1:0:0:0:0:0:0", endOnFailure: true }

Counterexample: [[0,1]]

Shrunk 32 time(s)

Got error: Property failed by returning false

Encountered failures were:

- [[-1071512695,39964964,674730683]]

- [[39964964,674730683]]

- [[0,674730683]]

- [[0,337365342]]

- [[0,168682671]]

- $[[0, 5148]]$
- $[[0, 2574]]$
- $[[0, 1287]]$
- $[[0, 644]]$
- $[[0, 322]]$
- $[[0, 161]]$
- $[[0, 81]]$
- $[[0, 41]]$
- $[[0, 21]]$
- $[[0, 11]]$
- $[[0, 6]]$
- $[[0, 3]]$
- $[[0, 2]]$
- $[[0, 1]]$

```
{
  firstname: "...", lastname: "...", phone: "...",
  shipmentAddresses: [
    { street: "...", houseNumber: "...", city: "...",
      postalCode: "...", country: "..." },
    { street: "...", houseNumber: "...", city: "...",
      postalCode: "...", country: "..." }
  ],
  invoiceAddress: {
    street: "...", houseNumber: "...", city: "...",
    postalCode: "...", country: "..."
  }
}
```

GET /hosen/lee/farbe-hellblau/40-70-euro/seite-2



```
{  
  category: "trousers",  
  brand:    "lee",  
  color:    "light-blue",  
  page:     2,  
  price:    { from: 40, to: 70 }  
}
```



Property-Driven Development

Design.

State Happens

```
{
  firstname: "...", lastname: "...", phone: "...",
  shipmentAddresses: [
    { street: "...", houseNumber: "...", city: "...",
      postalCode: "...", country: "..." },
    { street: "...", houseNumber: "...", city: "...",
      postalCode: "...", country: "..." }
  ],
  invoiceAddress: {
    street: "...", houseNumber: "...", city: "...",
    postalCode: "...", country: "..."
  }
}
```

```
{
  event: "invoiceAddressUpdated",
  customerId: "b572adc8-e5c9-4db1-b3f1-cb3979b0a171",
  emittedAt: "2019-09-13T13:18:49Z",
  newAddress: {
    street: "...", houseNumber: "...", city: "...",
    postalCode: "...", country: "..."
  }
}
```

RedisCache

set **get** **delete** **clear**

```
class Set { constructor(key, value) { /* ... */ } }
```

```
class Get { constructor(key) { /* ... */ } }
```

```
class Delete { constructor(key) { /* ... */ } }
```

```
class Clear { constructor() { /* ... */ } }
```

```
[  
  Set("a", "b"),  
  Clear(),  
  Set("a", "c"),  
  Delete("a"),  
  Get("a")  
]
```

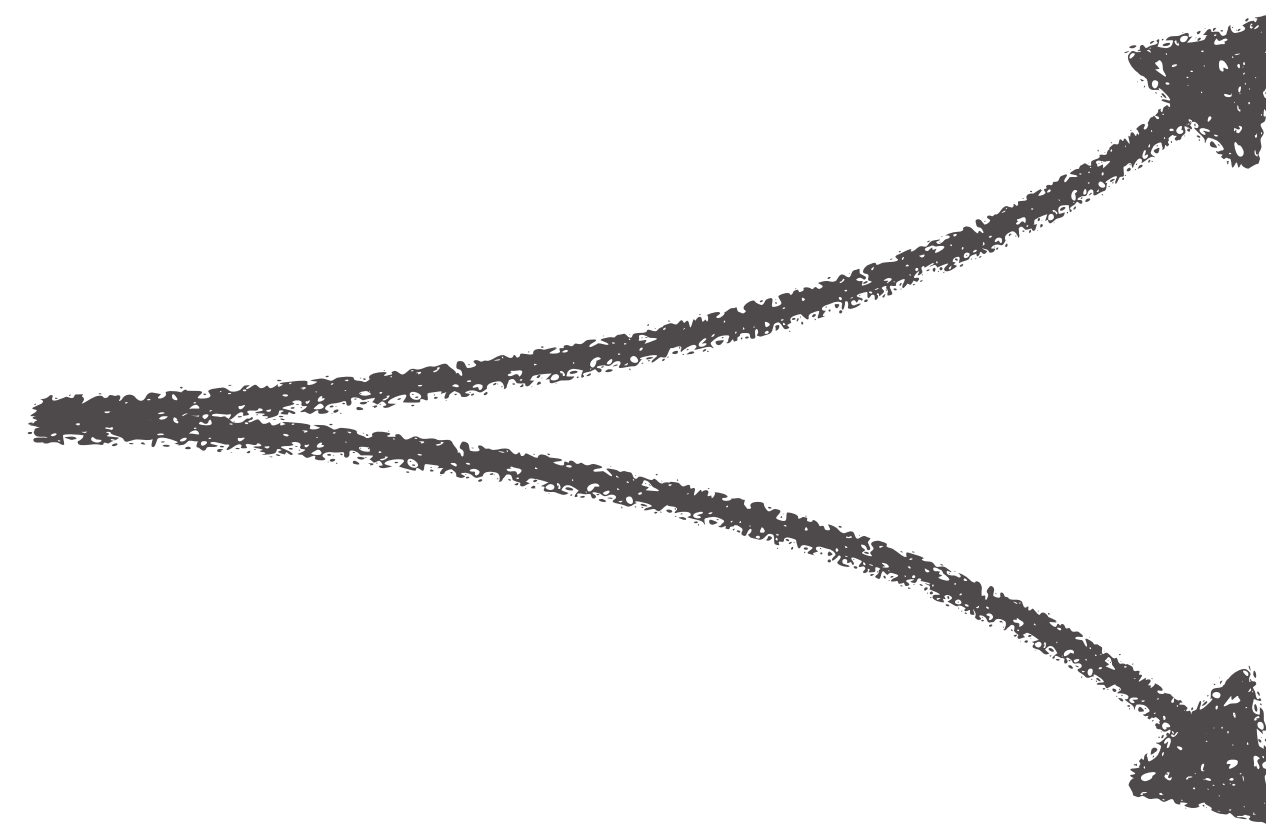
```
[  
  Set("a", "b"),  
  Clear(),  
  Set("a", "c"),  
  Delete("a"),  
  Get("a")  
]
```

RedisCache




```
[  
  Set("a", "b"),  
  Clear(),  
  Set("a", "c"),  
  Delete("a"),  
  Get("a")  
]
```

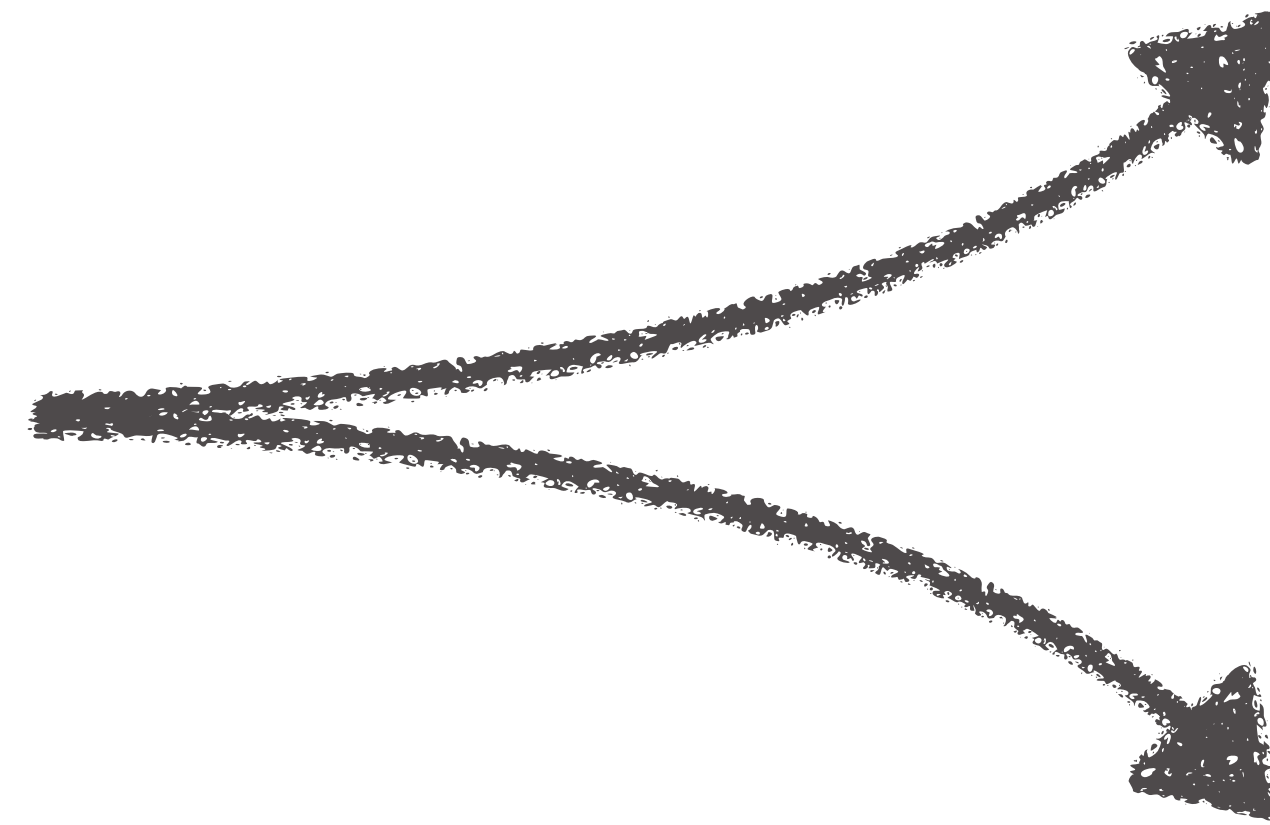
RedisCache



Oracle

```
[  
  Set("a", "b"),  
  Clear(),  
  Set("a", "c"),  
  Delete("a"),  
  Get("a")  
]
```

RedisCache



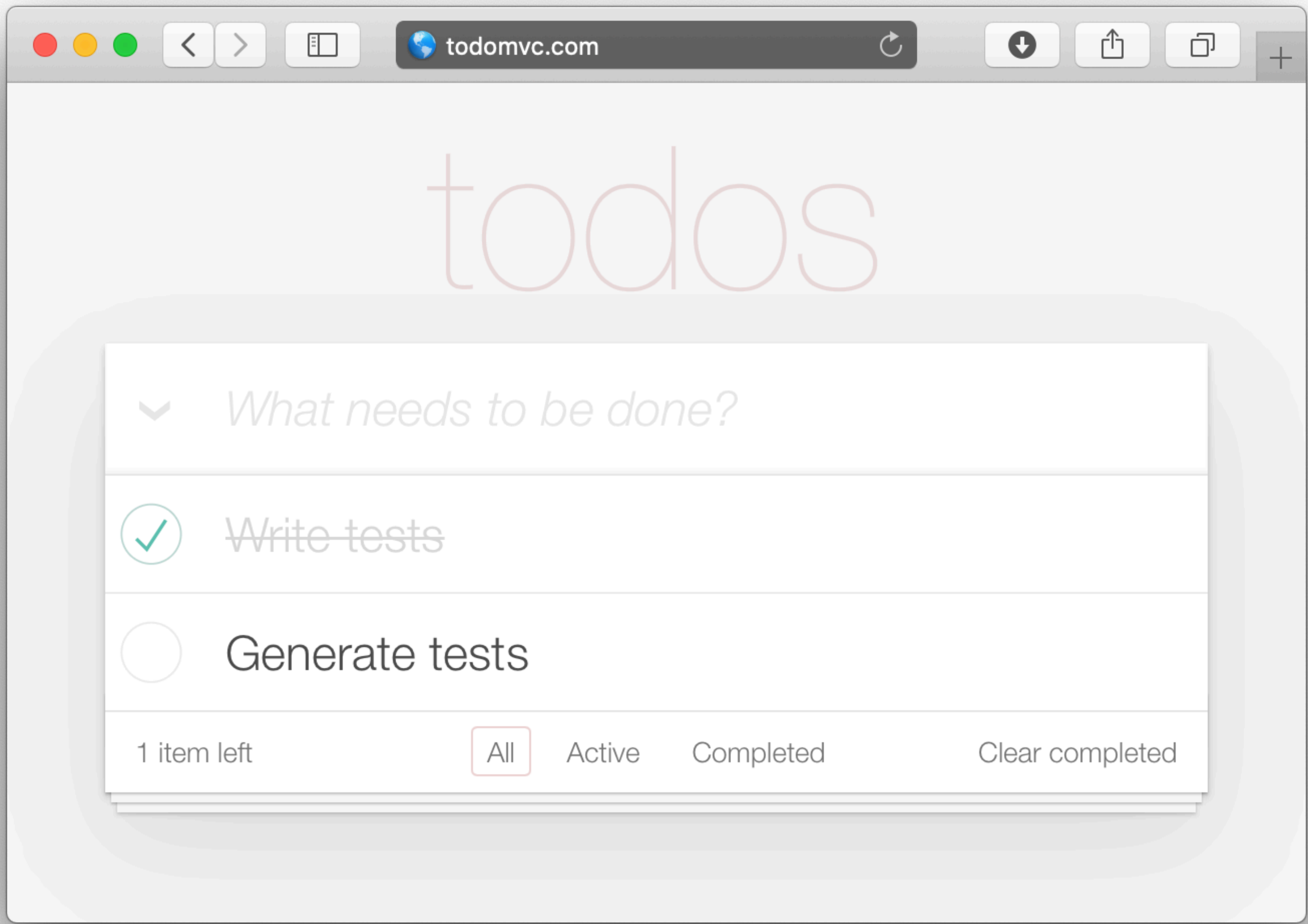
```
new Map()
```

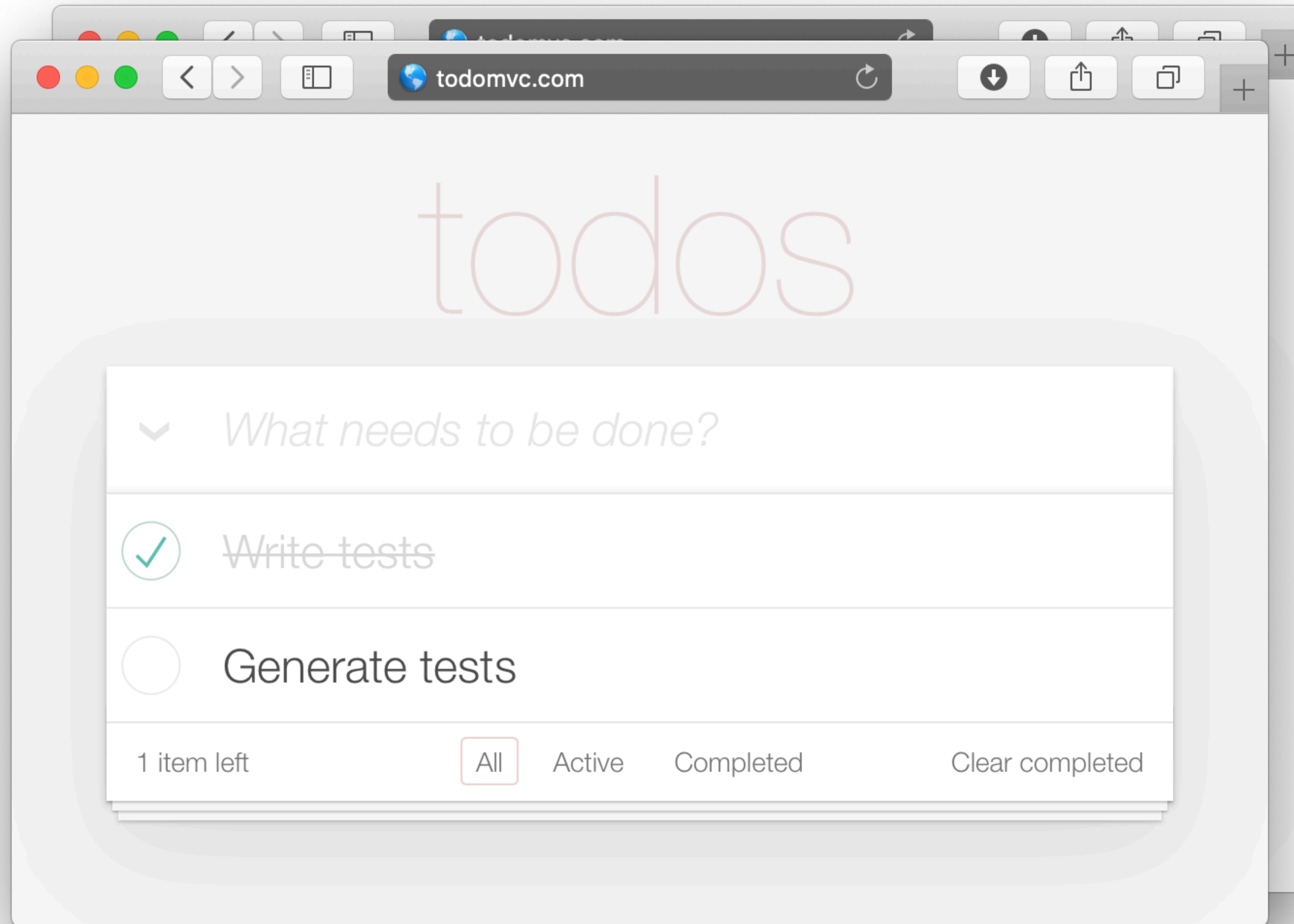
Testing the Hard Stuff and Staying Sane

John Hughes

State Happens

Happens Happens Happens Happens Happens Happens
States States States States States States
Happens Happens Happens Happens Happens Happens
States States States States States States
Happens Happens Happens Happens Happens Happens
States States States States States States
Happens Happens Happens Happens Happens Happens
States States States States States States
Happens Happens Happens Happens Happens Happens
States States States States States States





todos

▼ *What needs to be done?*

~~Write tests~~

Generate tests

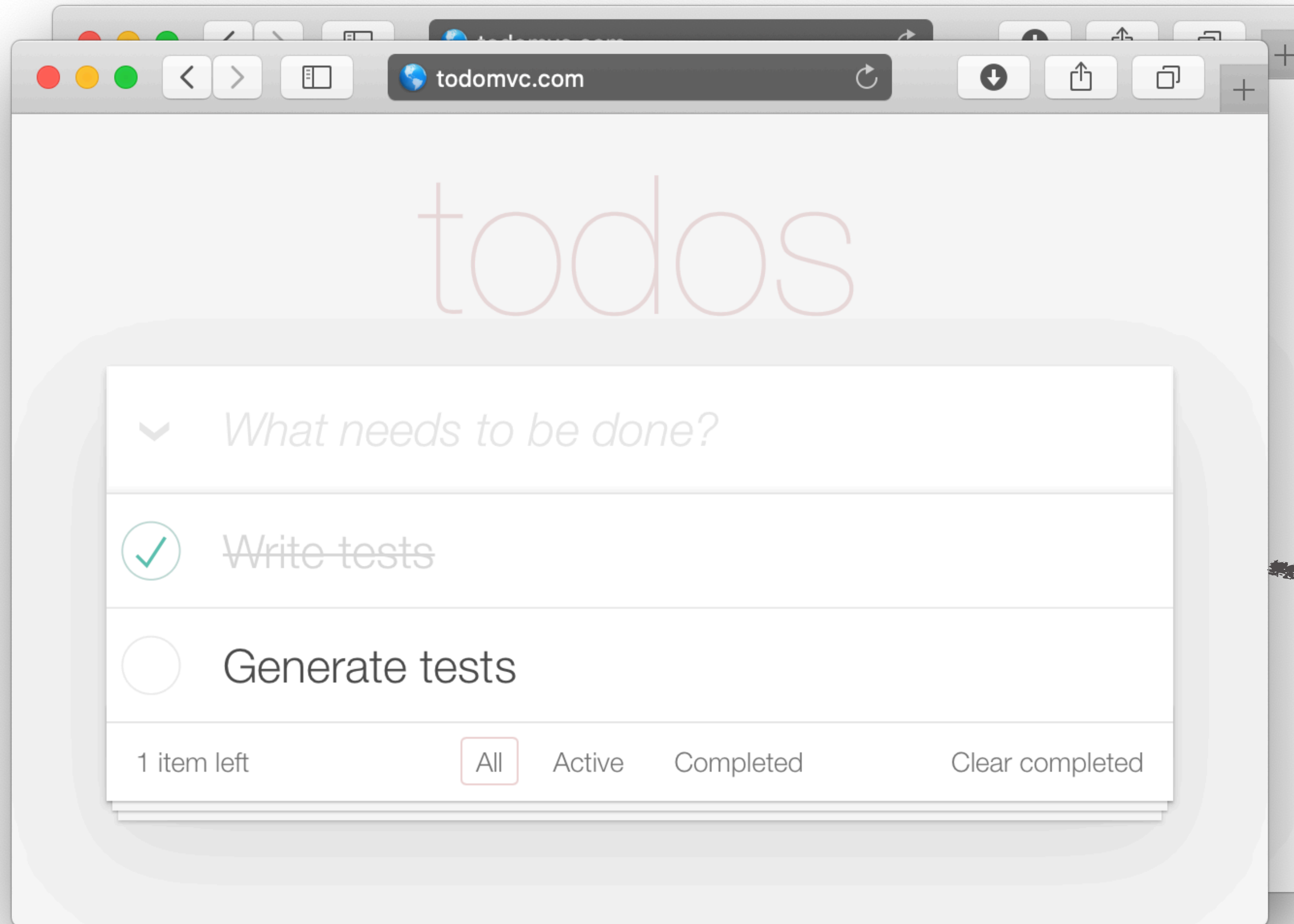
1 item left

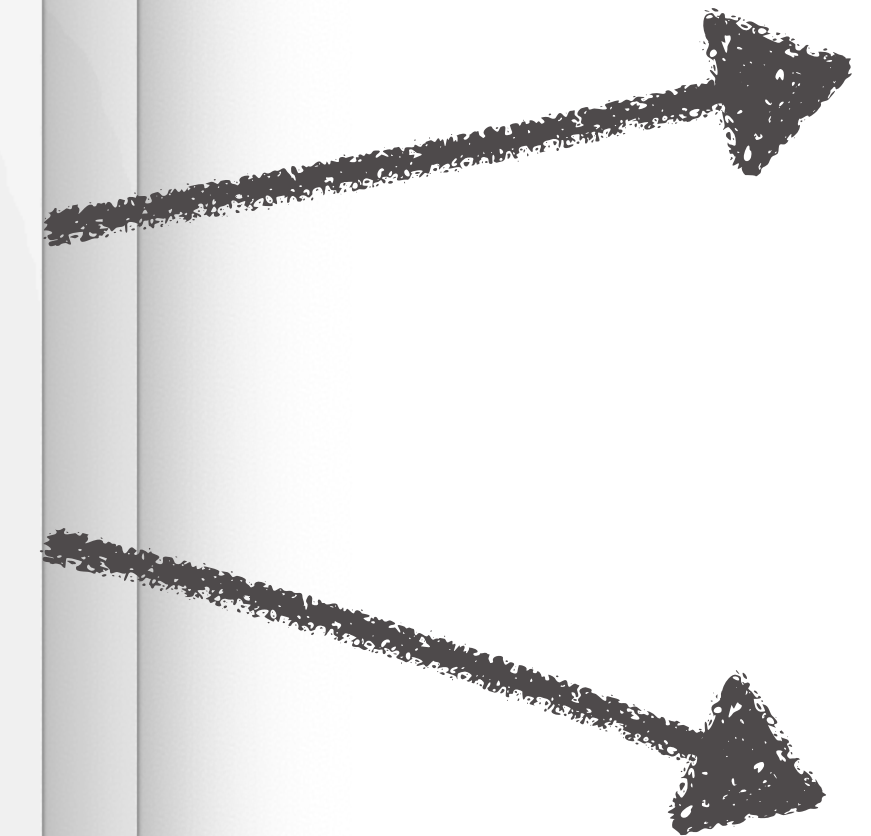
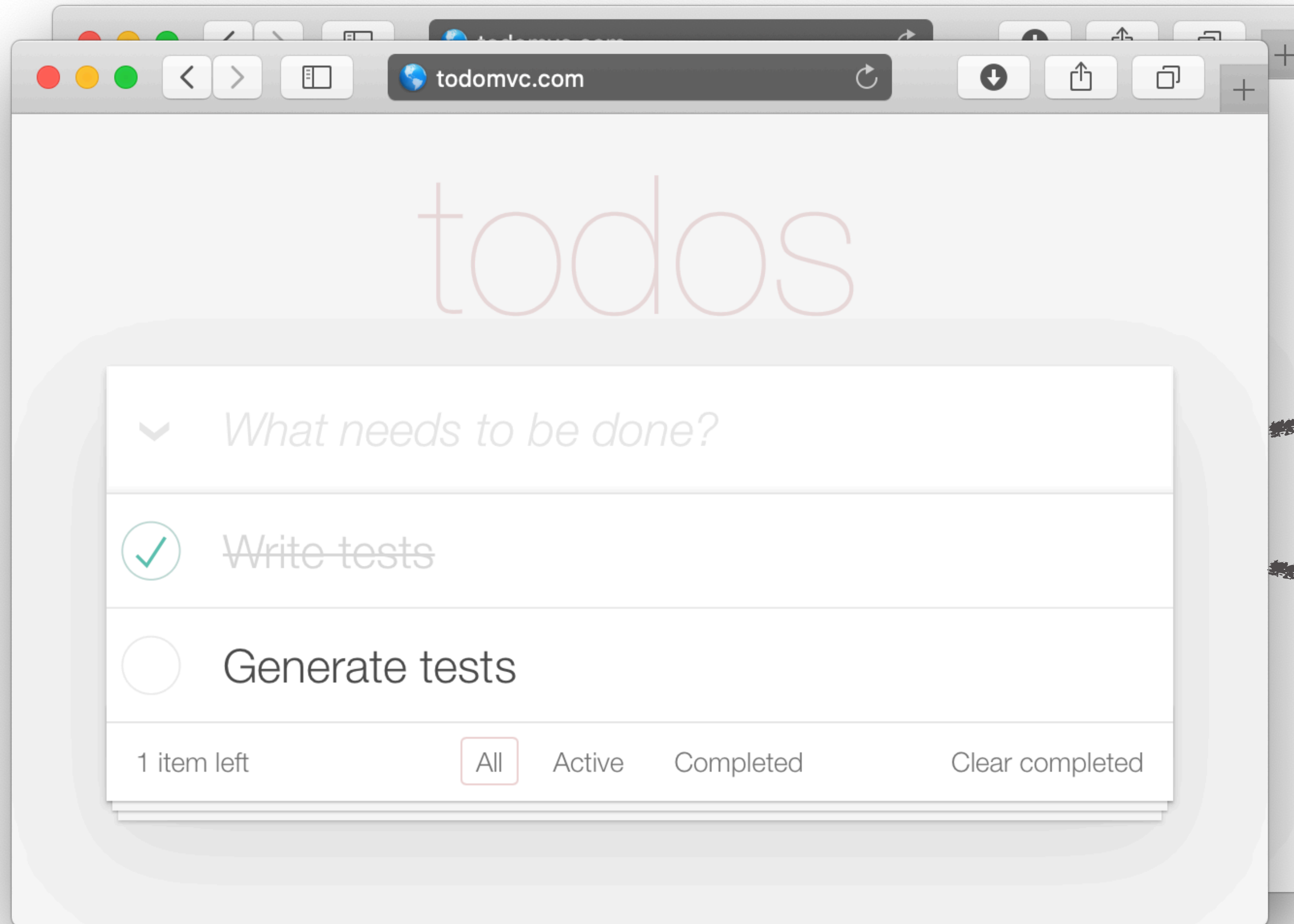
All

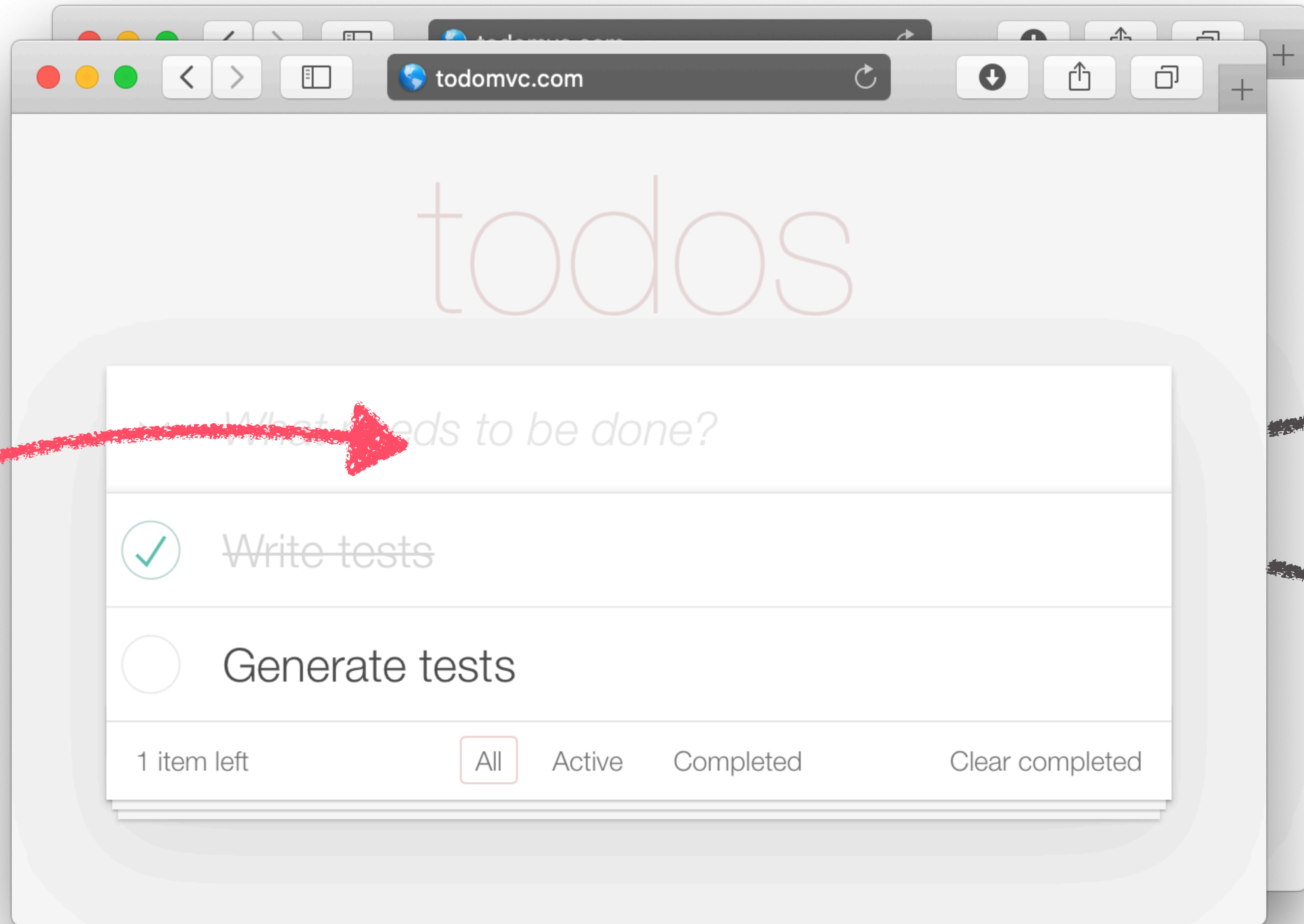
Active

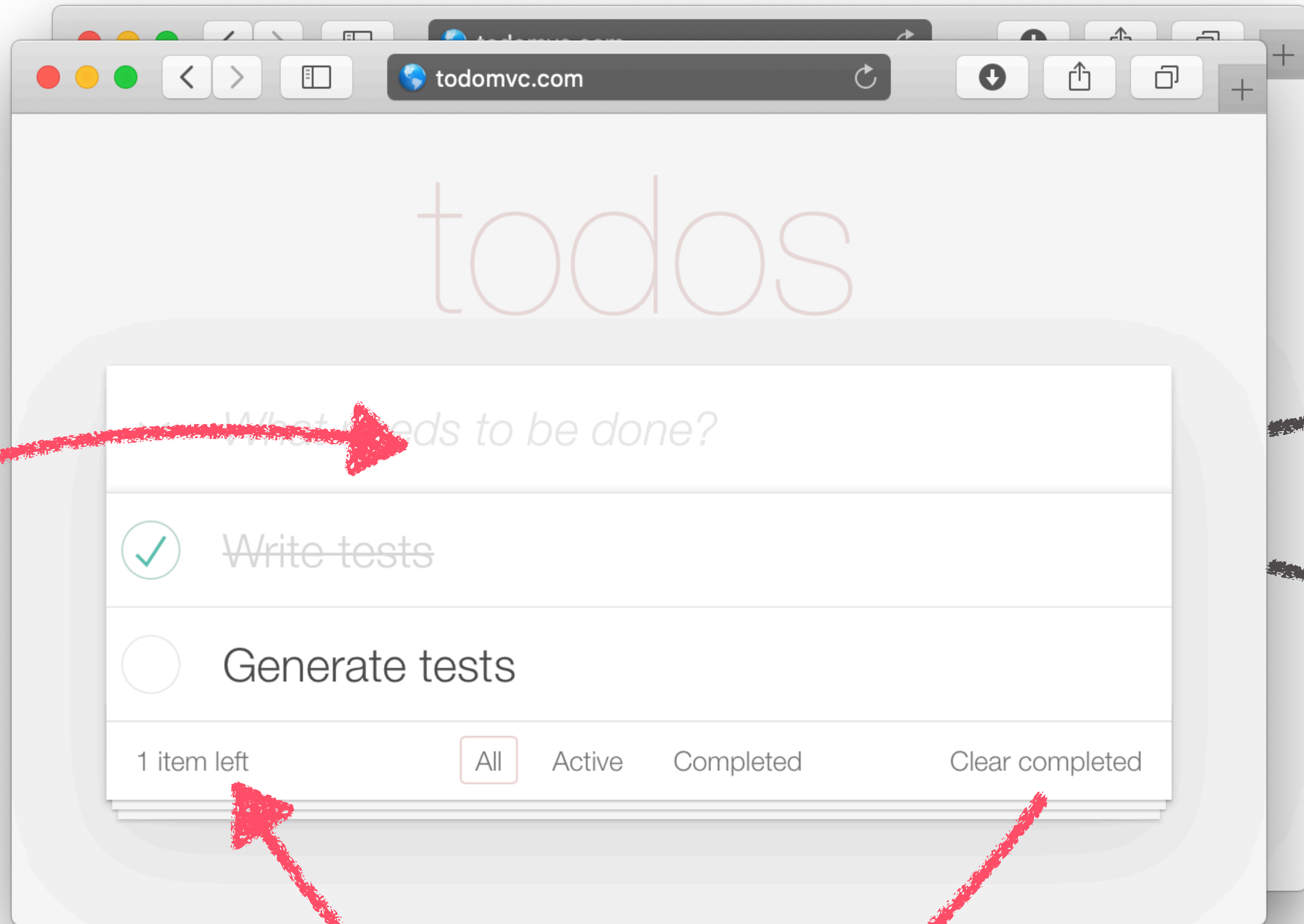
Completed

Clear completed









Hmm. We're having trouble finding that site.

We can't connect to the server at `www.example.org`.



If that address is correct, here are three other things you can try:

- Try again later.
- Check your network connection.
- If you are connected but behind a firewall, check that Firefox has permission to access the Web.

[Try Again](#)

Race Conditions

[α β γ δ]
[1 2 3 4]

[α β γ δ]

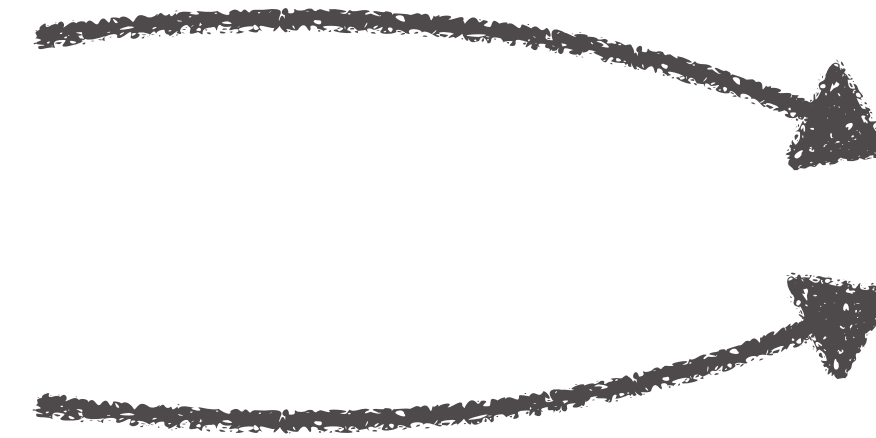
[1 2 3 4]

[α β γ δ 1 2 3 4]

[[[α β γ δ]
[1 2 3 4]
[α β 1 2 3 4 γ δ]

[[[α β γ δ]
[1 2 3 4]
[α 1 β 2 γ 3 δ 4]

[α β γ δ]
[1 2 3 4]



Ⓢ

[α β 1 2 3 γ δ 4]



①

[1 α β 2 3 4 γ δ]



②

⋮

[α 1 β 2 γ 3 δ 4]



⑨

Jepsen: On the perils of network partitions

Kyle Kingsbury

**I want you to become
better at testing**

Automation.

Design.

Quality.



16.01.2020 // Berlin.JS

Let's Test That Real Quick

Jan Stępień
janstepien.com

INNOQ