

Action

```
var statusUpdate = Reflux.createAction();
var status Update = Reflux.creat eAc tion();
```

Store

```
// Creating a DataStore - Listening to statusUpdate
action
var status Store = Reflux.creat eSt ore({
  init: function() {
    this.s.l ist enT o(s tat usU pdate,
this.o utput);
  },
  output: functi on( flag) {
    var status = flag ? 'ONLINE' :
'OFFLINE';
    thi s.t rig ger (st atus);
  }
});
```

Store

```
// Creating a Data Store - Listening to textUpdate
action
var textStore = Reflux.creat eSt ore({
  init: function() {
    this.s.l ist enT o(t ext Update,
this.o utput);
  },
  output: function() {
    var i, args = Array.p ro tot -
ype.sl ice.ca ll( arg uments, 0);
    for (i = 0; i < args.l ength; i++)
    {
      thi s.w rit eOu t(a rgs -
[i]);
    }
  },
  writ eOut: functi on( text) {
    thi s.t rig ger (text);
  }
});
```

Store - aggregate

```
// Creating an aggregate DataStore that is
listening to textStore and statusStore
var storyStore = Reflux.creat eSt ore({
  init: function() {
    thi s.l ist enT o(s tat usS tore,
this.s tat usC han ged);
    thi s.l ist enT o(t ext Store,
this.t ext Upd ated);
    thi s.s toryArr = [];
  },
  sta tus Cha nged: functi on( flag) {
    if (flag === 'OFFLINE') {
      thi s.t rig ger ('Once upon
a time the user did the following: ' + this.s -
tor yAr r.j oin(', '));
      // empty storyArr
      thi s.s tor yAr r.s pli -
ce(0, this.s tor yAr r.l ength);
    }
  },
  tex tUp dated: functi on( text) {
    thi s.s tor yAr r.p ush (text);
  }
});
```

Component

```
// Fairly simple view component that outputs to
console
function Consol eCo mpo nent() {
  tex tSt ore.l iste n(f uncti on(text) {
    con sol e.l og( 'text: ', text);
  });
  sta tus Sto re.l iste n( fu ncti on( st -
atus) {
    con sol e.l og( 'st atus: ',
status);
  });
  sto ryS tor e.l iste n( fun cti on( story)
{
    con sol e.l og( 'story: ', story);
  });
}
new Consol eCo mpo nent();
```



Invoke

```
// Invoking the action with arbitrary parameters
status Update(true);
textUpdate("testing", 1337, { "test": 1337
});
status Update(false);
```

/ Will output the following:**

```
*
* status: ONLINE
* text: testing
* text: 1337
* text: { test: 1337 }
* story: Once upon a time the user did the following: testing, 1337,
[object Object]
* status: OFFLINE
*/
```

Source

<https://github.com/spoike/refluxjs/blob/master/test/index.js>



By **fangjian**

cheatography.com/fangjian/

Published 12th June, 2015.

Last updated 12th May, 2016.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>