KYLE SIMPSON GETIFY@GMAIL.COM

JAVASCRIPT: THE RECENT PARTS

ES6 / ES2015

- Rest/Spread Operator (...)
- Iterators + Generators

Rest/Spread Operator

ES6 / ES2015

```
function lookupRecord(id) {
    var otherParams = [].slice.call( arguments, 1 );
    otherParams.unshift(
        "people-records", id.toUpperCase()
    );
    return db.lookup.apply( null, otherParams );
}
```

spread: imperative

```
1 function lookupRecord(id,...otherParams) {
2     return db.lookup(
3         "people-records", id,...otherParams
4     );
5 }
```

spread: declarative

Iterators + Generators

ES6 / ES2015

```
1  var str = "Hello";
2  var it = str[Symbol.iterator]();
3
4  for (let v of it) {
5      console.log(v);
6  }
7  // "H" "e" "l" "l" "o"
8
9  for (let v of str) {
10     console.log(v);
11 }
12 // "H" "e" "l" "l" "o"
```

iterators: declarative iteration

```
1 var str = "Hello";
2
3 var letters = [...str];
4 letters;
5 // ["H","e","l","l","o"]
```

iterators: declarative iteration

```
var obj = {
    a: 1,
    b: 2,
    c: 3,
    *[Symbol.iterator](){
        for (let key of Object.keys(this)) {
            yield this[key];
        }
    }
};

[...obj];
// [1,2,3]
```

generator: declarative iterator

ES2016

- Array .includes(..)

Array .includes(..)

ES2016

```
var arr = [10,20,NaN,30,40,50];

arr.includes( 20 );           // true

arr.includes( 60 );           // false

arr.includes( 20, 3 );        // false

arr.includes( 10, -2 );       // false

arr.includes( 40, -2 );       // true

arr.includes( NaN );          // true
```

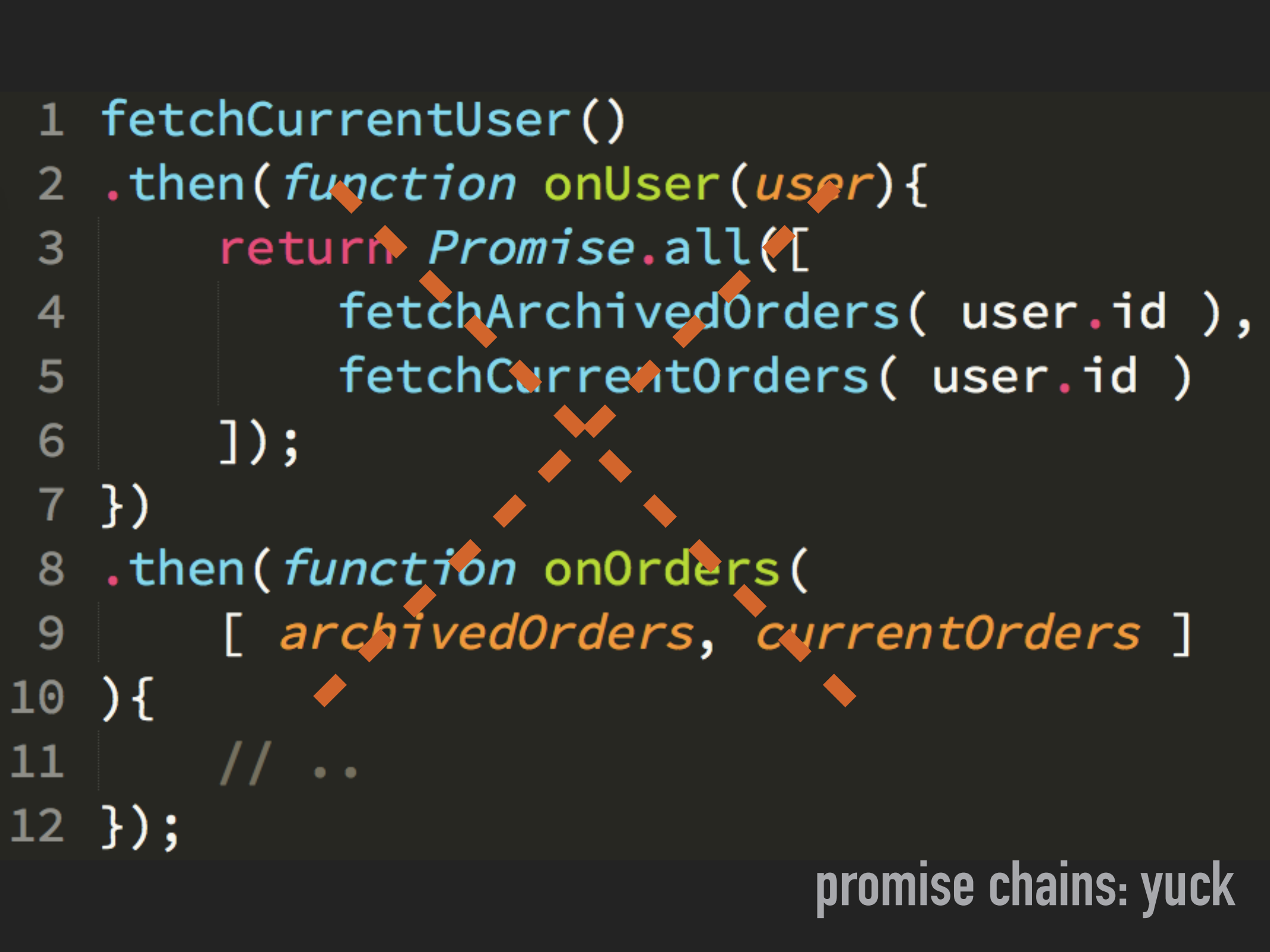
includes API > syntax

ES2017

- async .. await

async .. await

ES2017


fetchCurrentUser()
.then(function onUser(user){
return Promise.all([
fetchArchivedOrders(user.id),
fetchCurrentOrders(user.id)
]);
})
.then(function onOrders(
[archivedOrders, currentOrders]
){
// ..
});
promise chains: yuck

```
async function main() {
    var user = await fetchCurrentUser();

    var [ archivedOrders, currentOrders ] =
        await Promise.all([
            fetchArchivedOrders( user.id ),
            fetchCurrentOrders( user.id )
        ]);

    // ..
}

main();
```

async functions

- await Only Promises
- Scheduling (Starvation)
- External Cancelation

github.com/getify/CAF

```
1  var token = new CAF.cancelToken();
2
3  var main = CAF( function *main(signal,url){
4      var resp = yield fetch( url, { signal } );
5      // ..
6
7      return resp;
8  } );
9
10 main( token.signal, "http://some.tld/other" )
11 .then( onResponse, onCancelOrError );
12
13 // only wait 5 seconds for the request!
14 setTimeout( function onElapsed(){
15     token.abort( "Request took too long!" );
16 }, 5000 );
```

cancelable async functions

ES2018

- RegExp Improvements
- async* .. yield await

RegExp Improvements

ES2018

```
1 var msg = "Hello World";
2
3 msg.match(/(?<=e)(l.)/g);
4 // ["ll"]
5
6 msg.match(/(?<!e)(l.)/g);
7 // ["lo","ld"]
```

look behind

```
1  var msg = "Hello World";
2
3  msg.match(/.(l.)/);
4  // ["ell","ll"]
5
6  msg.match(/([jkl])o Wor\1/);
7  // ["lo Worl","l"]
8
9  msg.match(/(?<cap>l.)/).groups;
10 // {cap: "ll"}
11
12 msg.match(/(?<cap>[jkl])o Wor\k<cap>/);
13 // ["lo Worl","l"]
14
15 msg.replace(/(?<cap>l.)/g,"-$<cap>-");
16 // "He-ll-o Wor-ld-"
17
18 msg.replace(/(?<cap>l.)/g,function re(...args){
19     var [,,,,{ cap }] = args;
20     return cap.toUpperCase();
21 });
22 // "HeLLo WorLD"
```

named capture groups

async* .. yield await

ES2018

```
async function *fetchURLs(urls) {
    for (let url of urls) {
        let resp = await fetch( url );
        if (resp.status == 200) {
            let text = await resp.text();
            yield text.toUpperCase();
        }
        else {
            yield undefined;
        }
    }
}
```

async generators

```
async function main(favoriteSites) {
    var it = fetchURLs( favoriteSites );

    while (true) {
        let res = await it.next();
        if (res.done) break;
        let text = res.value;

        console.log( text );
    }
}
```

```
async function main(favoriteSites) {
    for await (let text of fetchURLs( favoriteSites )) {
        console.log( text );
    }
}
```

async iteration: hooray!

THANKS!!!!

KYLE SIMPSON GETIFY@GMAIL.COM

JAVASCRIPT: THE RECENT PARTS