VoCode

Voice Coding Assistant

< 2018

early 2018

Launch Day + Jellyvision

NORTHERN TRUST

2018 - Jun 2019

Jellyvision®

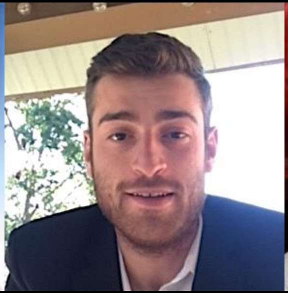
> Jun 2019

Overview

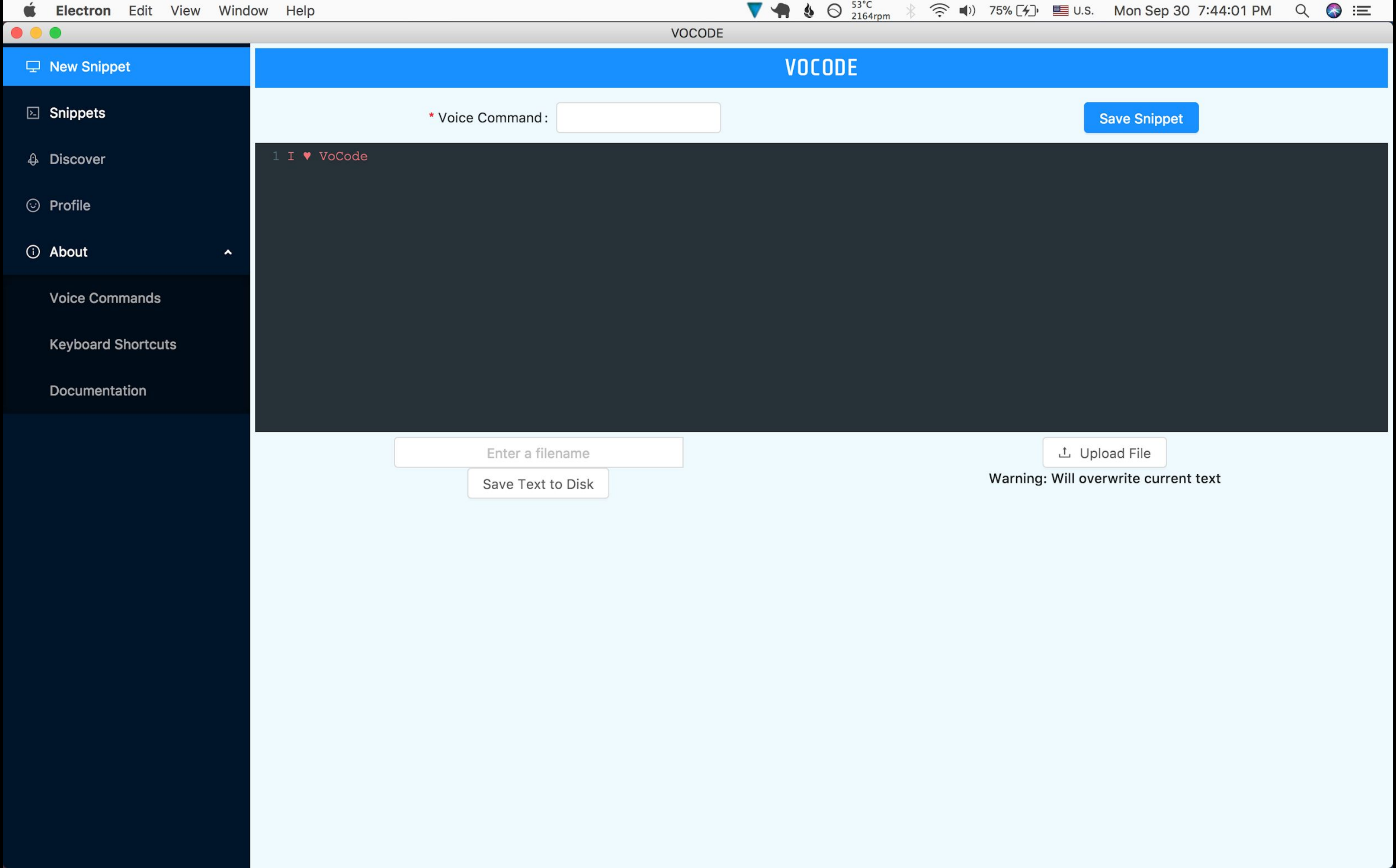
- A tour of Vocode
- What I’d change since March ‘18
- A taste of my experience at Fullstack

Dave Cohen

Kevin
Wuerdeman

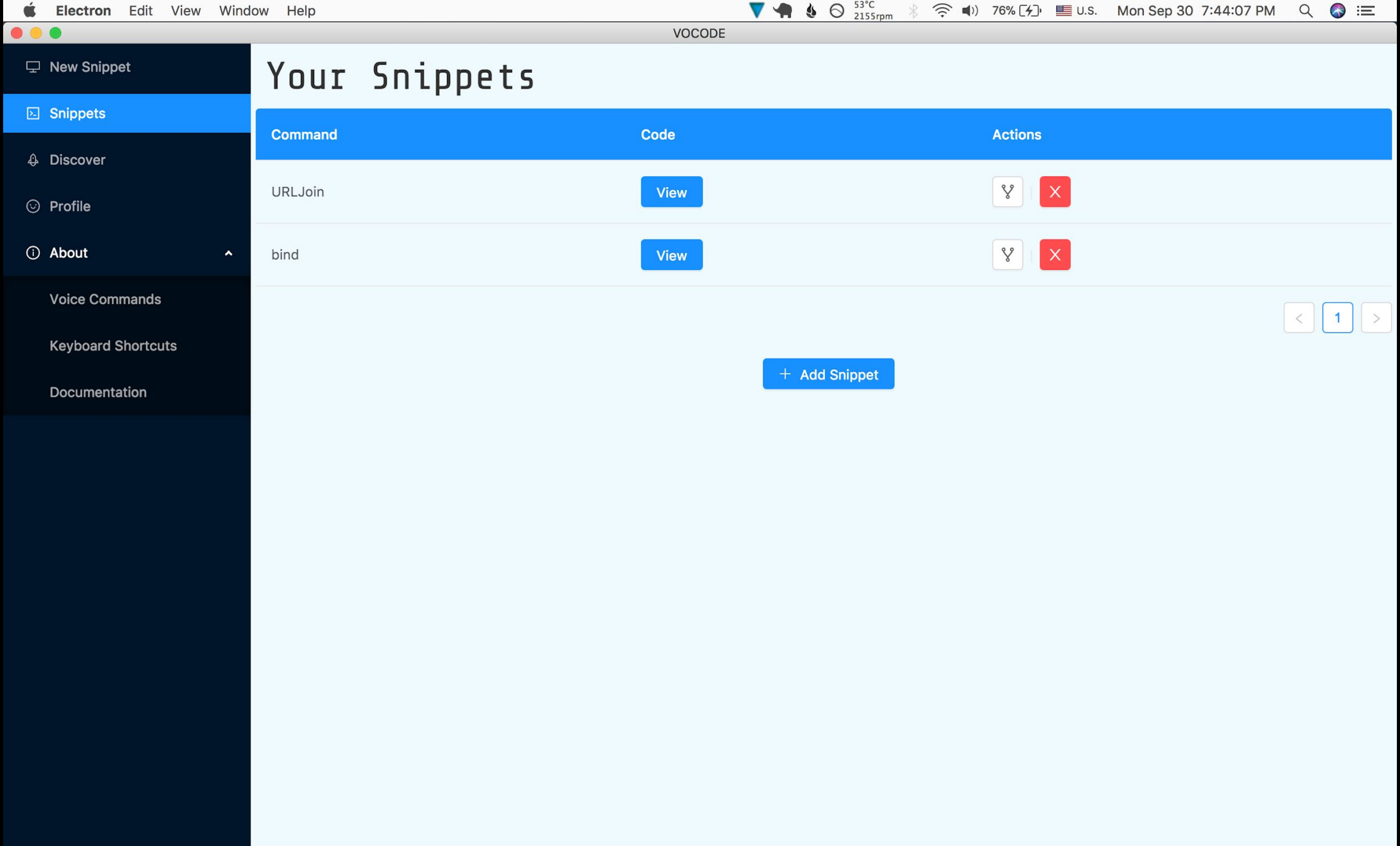

Jake Bergal

Josh Carlson



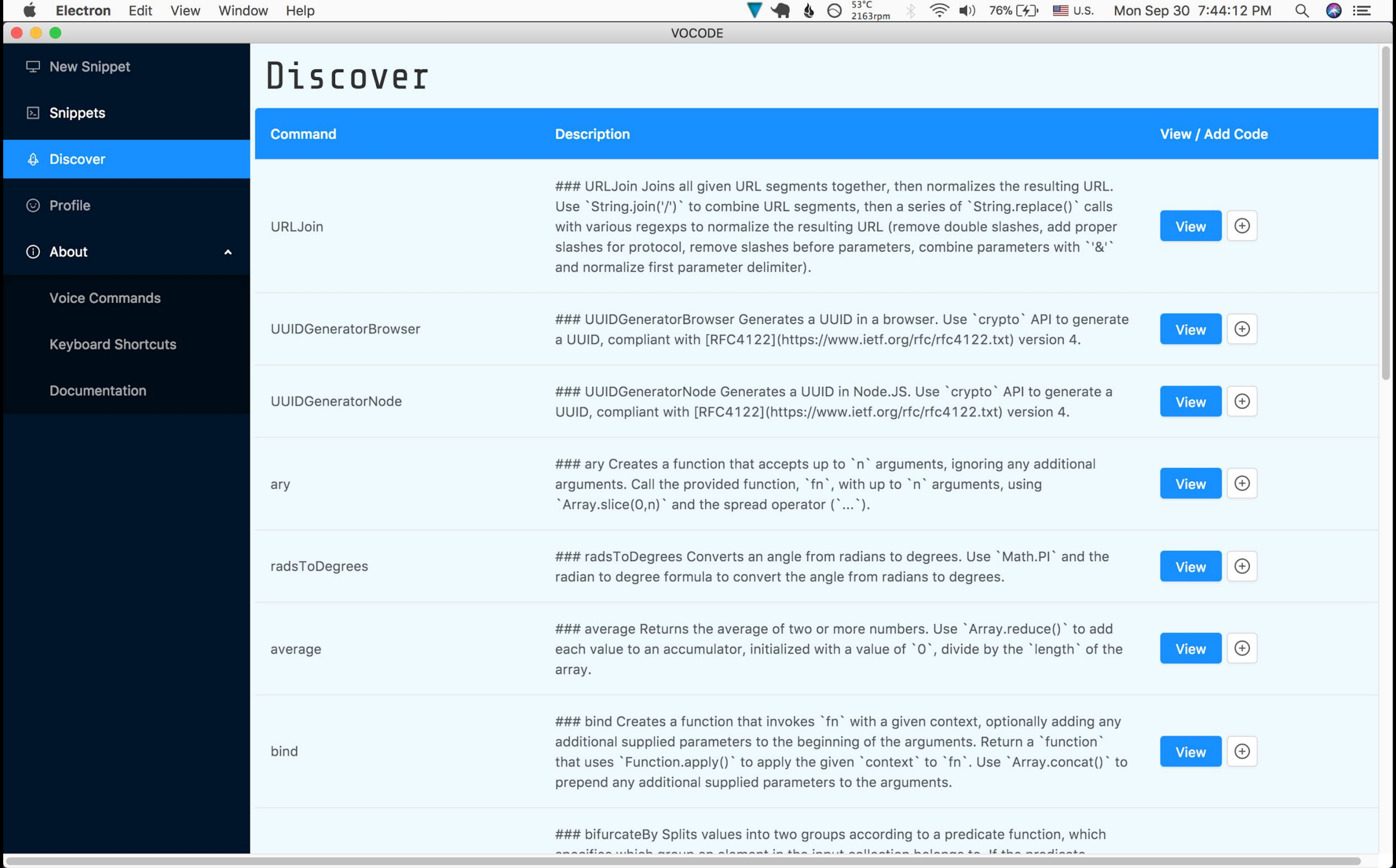
```
I ♥ VoCode
```

Warning: Will overwrite current text



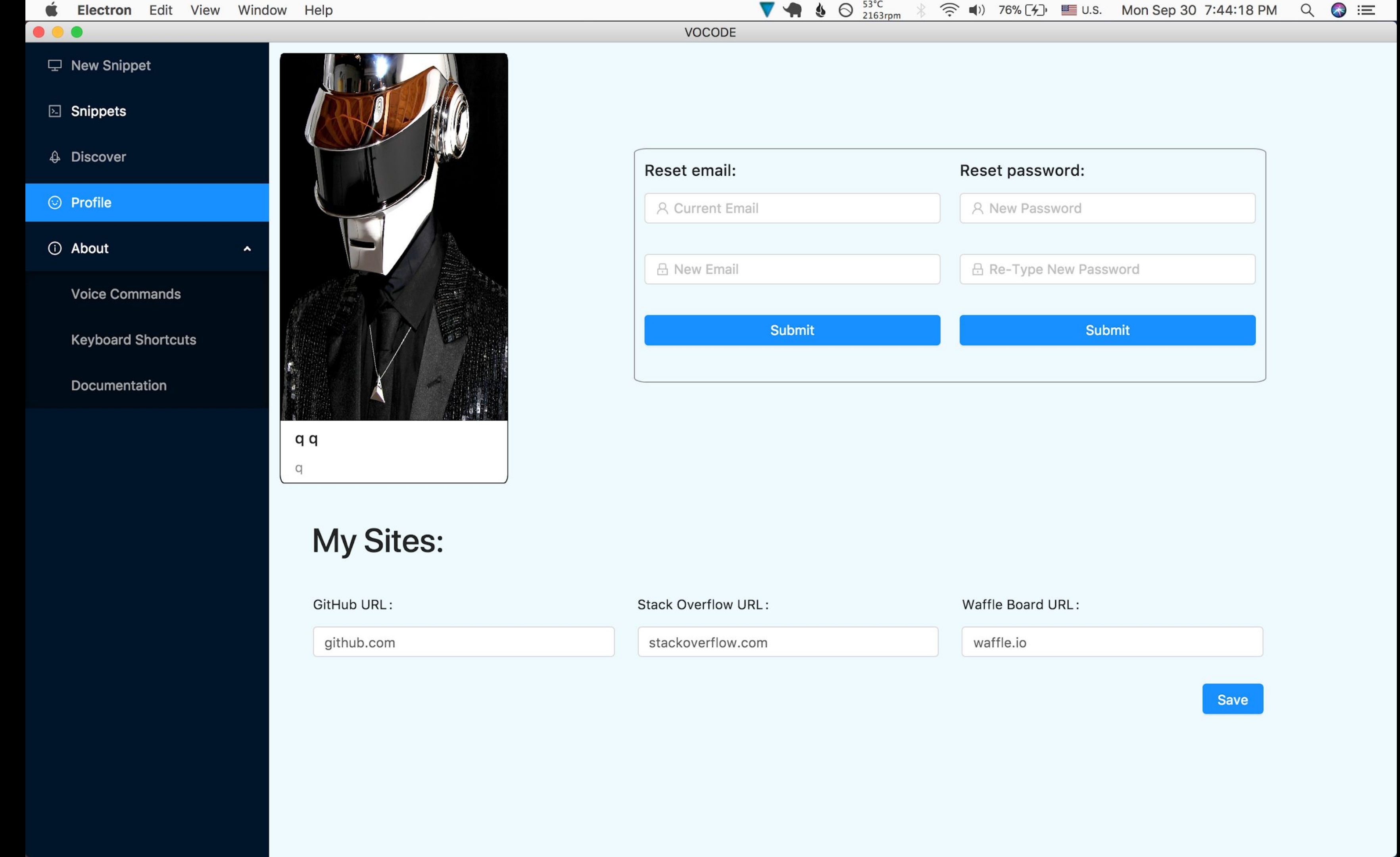
Your Snippets

Command	Code	Actions
URLJoin	View	
bind	View	



Discover

Command	Description	View / Add Code
URLJoin	### URLJoin Joins all given URL segments together, then normalizes the resulting URL. Use `String.join('/')` to combine URL segments, then a series of `String.replace()` calls with various regexps to normalize the resulting URL (remove double slashes, add proper slashes for protocol, remove slashes before parameters, combine parameters with `'&'` and normalize first parameter delimiter).	View ⊕
UUIDGeneratorBrowser	### UUIDGeneratorBrowser Generates a UUID in a browser. Use `crypto` API to generate a UUID, compliant with [RFC4122](https://www.ietf.org/rfc/rfc4122.txt) version 4.	View ⊕
UUIDGeneratorNode	### UUIDGeneratorNode Generates a UUID in Node.JS. Use `crypto` API to generate a UUID, compliant with [RFC4122](https://www.ietf.org/rfc/rfc4122.txt) version 4.	View ⊕
ary	### ary Creates a function that accepts up to `n` arguments, ignoring any additional arguments. Call the provided function, `fn`, with up to `n` arguments, using `Array.slice(0,n)` and the spread operator (`...`).	View ⊕
radsToDegrees	### radsToDegrees Converts an angle from radians to degrees. Use `Math.PI` and the radian to degree formula to convert the angle from radians to degrees.	View ⊕
average	### average Returns the average of two or more numbers. Use `Array.reduce()` to add each value to an accumulator, initialized with a value of `0`, divide by the `length` of the array.	View ⊕
bind	### bind Creates a function that invokes `fn` with a given context, optionally adding any additional supplied parameters to the beginning of the arguments. Return a `function` that uses `Function.apply()` to apply the given `context` to `fn`. Use `Array.concat()` to prepend any additional supplied parameters to the arguments.	View ⊕
	### bifurcateBy Splits values into two groups according to a predicate function, which	



q q

q

Reset email:

Current Email

New Email

Reset password:

New Password

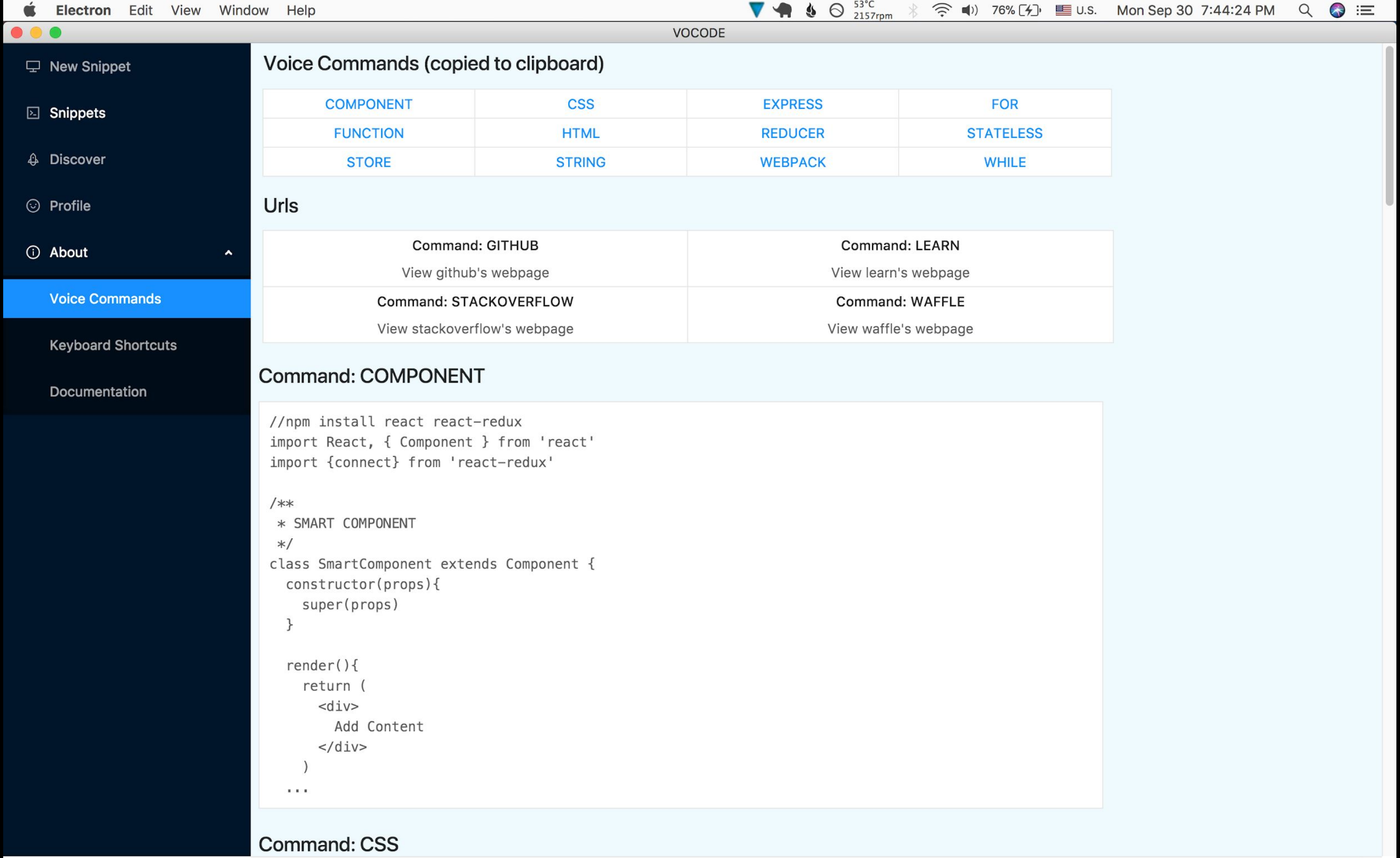
Re-Type New Password

My Sites:

GitHub URL : github.com

Stack Overflow URL : stackoverflow.com

Waffle Board URL : waffle.io



Voice Commands (copied to clipboard)

COMPONENT	CSS	EXPRESS	FOR
FUNCTION	HTML	REDUCER	STATELESS
STORE	STRING	WEBPACK	WHILE

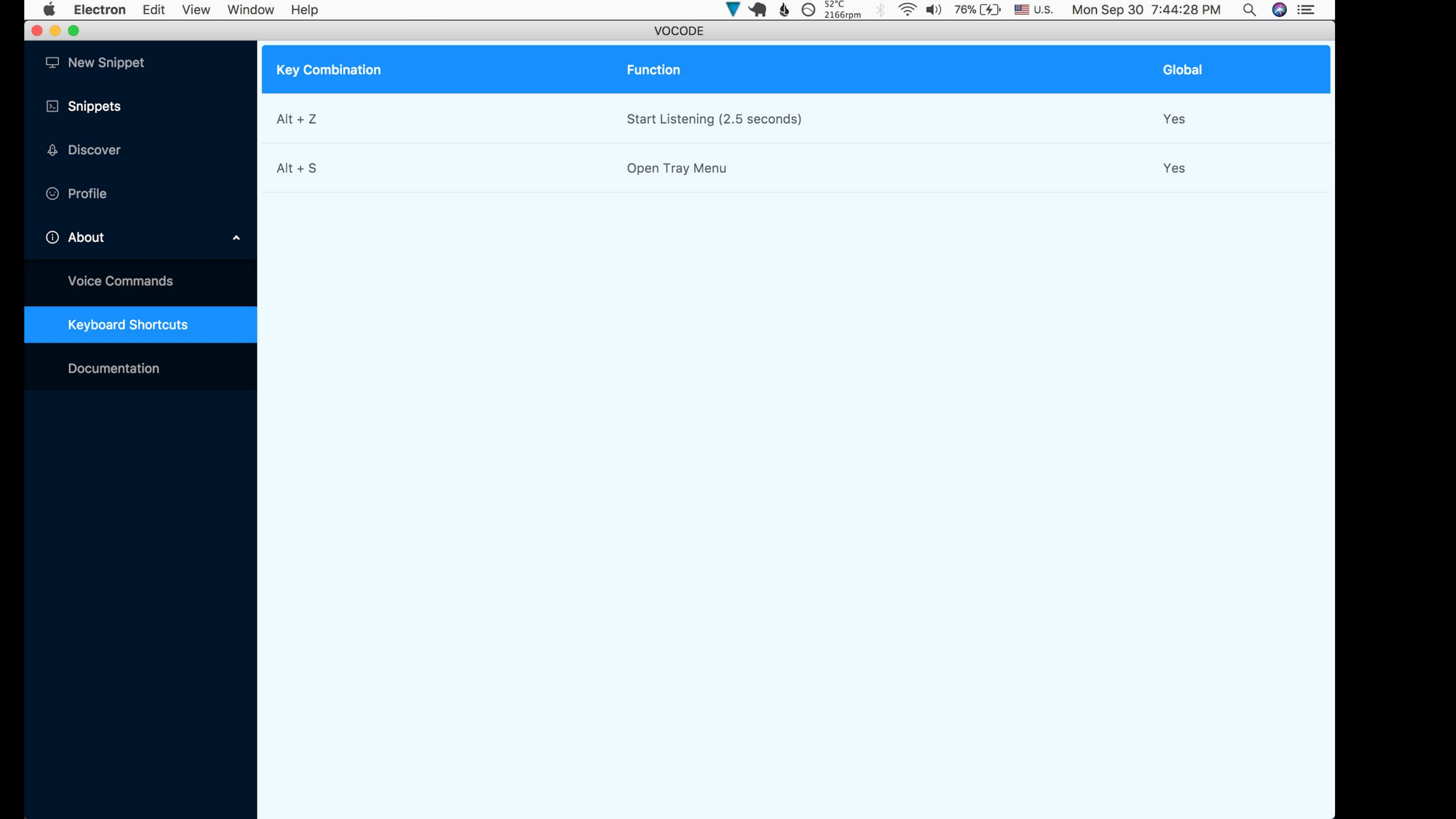
Urls

Command: GITHUB View github's webpage	**Command: LEARN** View learn's webpage
Command: STACKOVERFLOW View stackoverflow's webpage	**Command: WAFFLE** View waffle's webpage

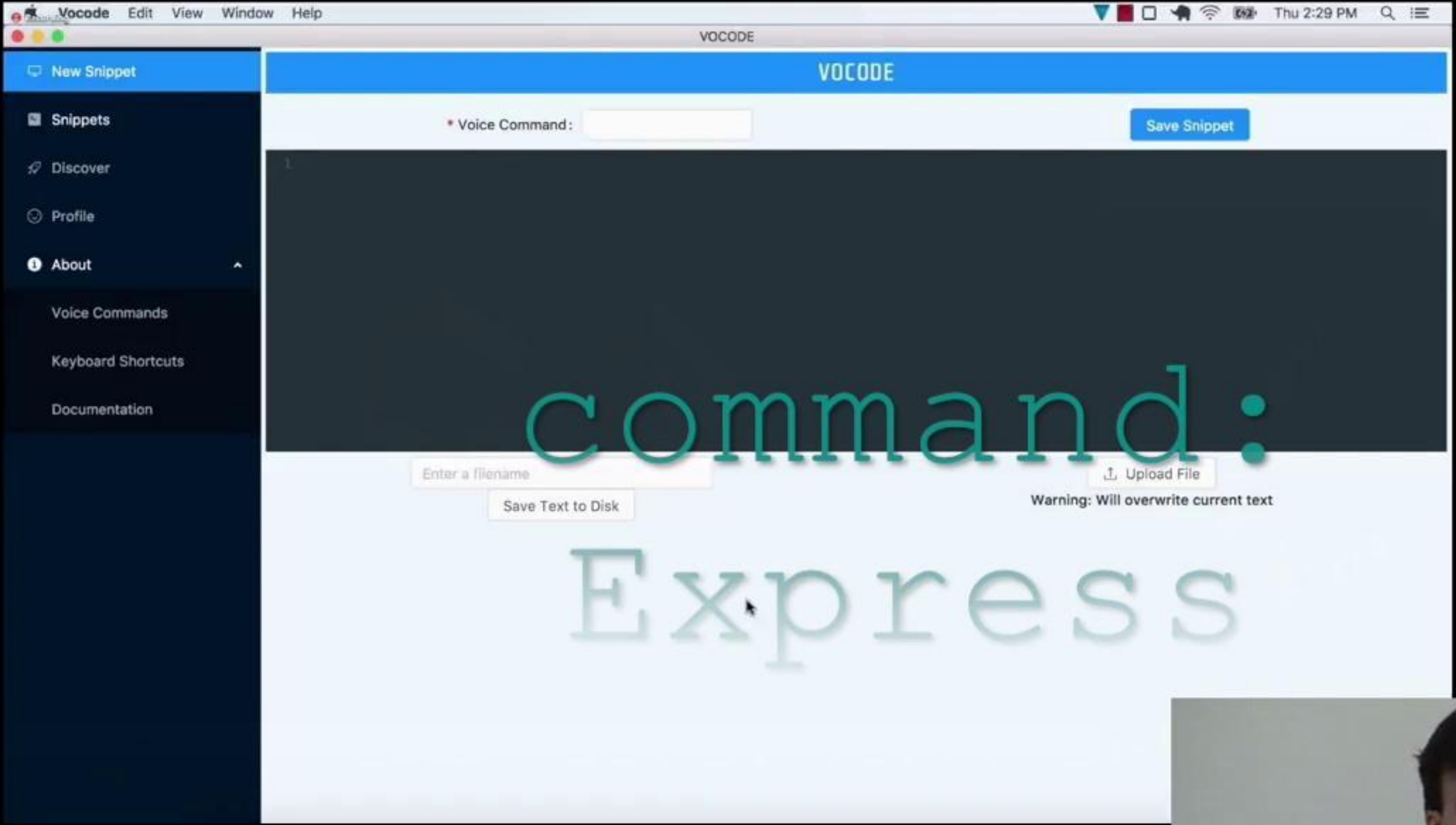
Command: COMPONENT

```
//npm install react react-redux
import React, { Component } from 'react'
import {connect} from 'react-redux'

/**
 * SMART COMPONENT
 */
class SmartComponent extends Component {
  constructor(props){
    super(props)
  }

  render(){
    return (
      <div>
        Add Content
      </div>
    )
  ...
```

Command: CSS



Key Combination	Function	Global
Alt + Z	Start Listening (2.5 seconds)	Yes
Alt + S	Open Tray Menu	Yes



command:

Express

Campus

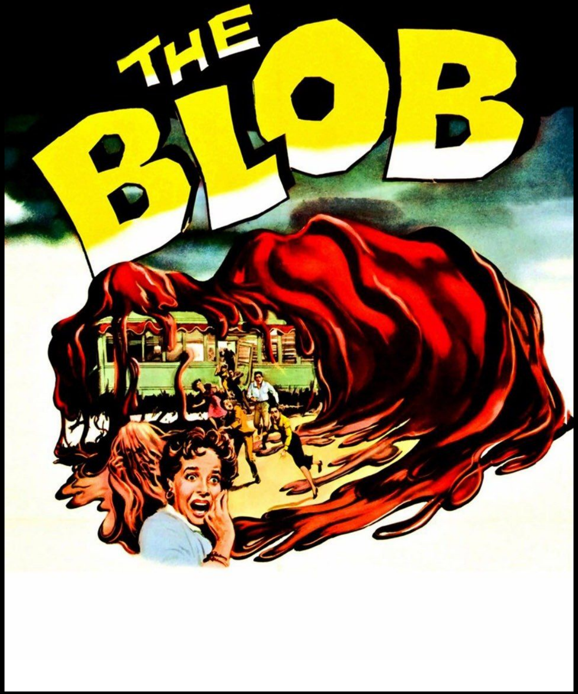
Fullstack Chicago

Vocode

Voice-controlled coding assistant

Recording Speech

$\longrightarrow$

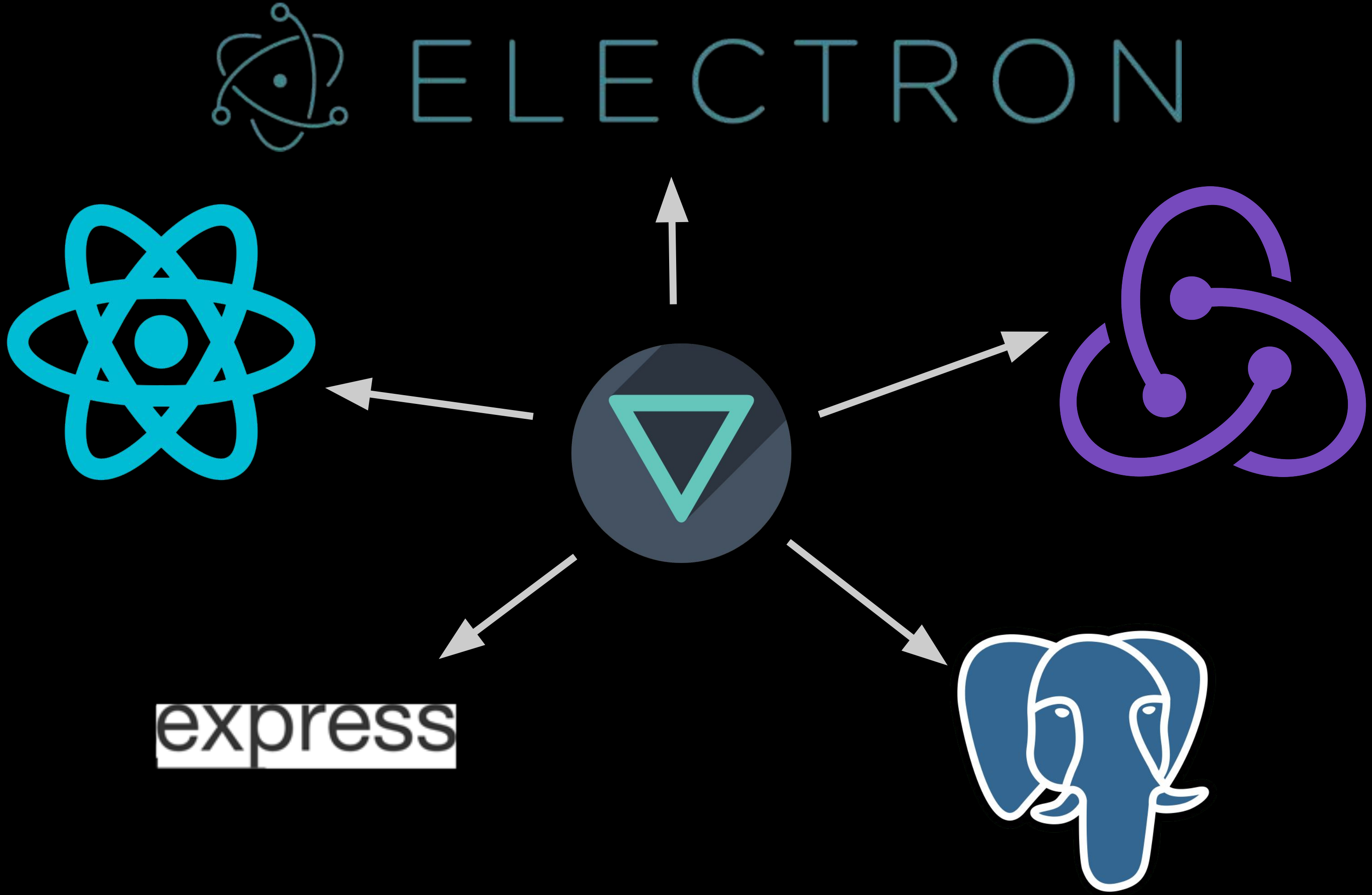


Speech Processing

Base 64 String

data:audio/wav;base64,UklGRiAABQBXQVZFZm10IBAAAAABAAIARKwAABCxAgAEABAAZGF0YQAABQAfBB8EOAQ4BI
gEiAQEAwQDIAIgAvgA+ACD/4P/DP4M/sn8yfzE/MT8pPuk+2/6b/oz+jP6TPlM+TP4M/if95/3Xfdd97P1s/X89Pz0S/VL9Q70Dv
R39Hf0PPU89eX05fQN9Q31E/YT9nD3cPfu9+73yPjI+Cz5LPle+I760fzR/Bf+F/6t/63/vgG+AVkEWQQ8BjwGlAeUBzEJMQn1
CfUJfAt8C+IL4guiC6ILMAwwDDoMOgw4DDgMpgumC1YLVgvXCtcKyArICssKywpCCkIKOwo7CgoJCgmrCKsINgg2CEwHT
AdkB2QHUQZRBuQF5AWwBLAEMAIwApEAkQBi/mL+M/wz/G/6b/pa+Fr4ffZ99rz1vPWU9JT0mPOY86nzqfNK80rzrPOs83X
zdfOT8pPyI/Mj8//y//IW8hbywPHA8bLwsvAu7y7vje6N7uTt5O2+7L7scetx6yjqKOpj6WPpE+gT6Inniec06DToA+gD6Hvoe+jm6
Obos+iz6NTp1OnY69jrVO1U7W/ub+628Lbwv/K/8t3z3fNU9lT2ufi5+KP5o/kO/A78T/5P/lT/VP/aAdoBUANQA58EnwTLBssGb
AhsCB8KHwq1C7ULBA4EDkQQRBAUEhQSkhOSE1cUVxSFFIUUuRS5FEAVQBWBFIEUTRNNE6USpRLzEPMQPQ89D3
UOdQ52DXYN7wzvDMsMywxxDHEMvgy+DIkNiQ2VDZUNyg3KDQcOBw48DTwNOA04DQwNDA1pDGkM+Av4CwoLCgstC
i0KYQhhCJ8HnwdcB1wHcwVzBasEqwSqA6oDNQI1AjIBMgGb/5v/LP4s/j78Pvwh+iH6BfgF+I31jfWT85Pz+vH68eDv4O+J7Y
nt0evR68/pz+mO6I7o/ef95/Lm8uaz57Pnbeht6FvpW+ks6yzrxevF64fth+2h76HvjfGN8VHzUfMX9Rf1DPgM+A36Dfq8+7z7vf2
9/Tn/Of8WABYA5AHkAcEDwQPEBMQE7wbvBr4IvggvCi8K2AvYC1INUg13DncObw9vD9kP2Q/8DvwOwg7CDvMN8w1sDG
wMWAtYC8wIzAhCBkIGYwRjBIMCgwLIAeUBbQFtAZYBlgHeAd4BMQExAZYBlgH3AfcBUgJSAkMDQwOuA64DWQRZBFE
FUQXHBccFhQaFBu0H7QeBCIEIfgh+CHkIeQgXCBcIywfLB+kG6QYYBhgG9QX1BTgEOAS5ArkCLQItAgQABACM/oz+3P3
c/V38XfyQ+pD65Pjk+Jn3mfde9V71XvNe8+rx6vGu767vTe5N7kDtQO037DfsiuyK7PPs8+xu7W7t+O747qzwrPCE8oTykPSQ
9J72nvbV+NX4QPtA+0z+TP4jASMBwQPBAwwGDAaSB5IHHAkcCYAKgAokDCQMZg1mDZkNmQ0CDQINYQxhDGYMZgy
hC6ELeAt4C2ELYQuVCZUJ3QjdCJwInAh/B38HMAYwBosFiwVyBXIFgASABNcD1wNSA1IDvQK9Am0CbQJPAk8CegJ6Ao
kCiQI2AzYDXgNeA00ETQRVBlUGKQcpBzoJOgmtC60L7QztDAUPBQ/GEMYQBxIHEmUTZROaE5oT+RP5E/ET8RMNEw0
TnBKcEiQRJBEmDyYPIQ0hDfUK9QrSCNIIjQaNBg8FDwUqAyoDdQF1AYQAhABJ/0n/Kf8p/+T+5P75/fn9M/4z/oH+gf5M/kz+
Pf49/kb+Rv4i/iL+Jf4l/if+J/7V/dX9T/1P/bz8vPwJ/Qn9Sv1K/Z39nf3u/u7+V/9X/2n/af/7//v/HgAeABYAFgAJAAkAc/9z/+z+7P42/j
b+p/yn/Nn72fuw+rD6yvjK+Ef3R/dB9UH19fP18//y//Jj8mPyFPlU8qzwrPCz8LPwjfCN8OHv4e8b8RvxPvE+8VbxVvGE8oTyyfL
J8iPzI/Oe857z2/Pb88b0xvSq9ar1AfYB9v/2//bb99v3UPhQ+L/5v/ks+iz6kfqR+tv82/wl/gj+Q/9D/4IBggErAysDgQSBBBN8F3wX5
BvkG0QbRBiMHIwdLB0sH7gbuBhEHEQeeBp4GuwbuBvEG8QYQYtBy0HeAd4B3wHfAchCCEI0AjQCBEJEQlxCXEJYApgCvcQ
K9AqBC4ELnAucC1gLWAt2C2C3YL/Qr9CsYKxgpkCmQKuAm4CSAJIAlLCEsIkweTB6oGqga/Bb8FkwSTBDwDPAlAuUCDAE
MAW//b/+i/qL+0vzS/Cn7KftZ+ln6mvia+Aj3CPdPY9T1GfQZ9JXylfKe8Z7xgfCB8PDA8O7v7u8e7p7ynvWe5Z7hLuEu4U7hTuD+4P7h
nuGe5j7mPuYu5i7hbuFu7K7sruOO8479Xv1e/L8Mvw4vDi8N/x3/HH8sfyPvM+8+82/PbP0/QUbT1RPVE9XX1dfWC9oL2Lfgt+Iv4i/4/
gc+Rz5tvm2+Sn6KfpK/Eb9w/8H/0f/R/9f/0vvF/rL/R/Mj6Ob8/ncn9+f/qX/Fn9lj5Sp/iX4/qk/ziqr9ZH7v+lrnR5NH6MjNqbvWMlTdjQBW+j+wP5+jv9rT9nu3eQt/fc
AIsApECkQK4ArgCjgOOA4wQEBAQ+BD4EIQQhBGMEYwQ+BT4FHwUfBfBfgE+AS1BLUE/gP+A1MDUwPuAu4C8wHzAS4BL
gEPAQ8B9//3/7r/uv87/zv/ZP5k/rj+uP6k/aT9WfxZ/Pf79/ub+5v7uB+4H7qvqq+qX5pflK+Ur5sfix+Pj3+Pfa99r3A/cD9071TvH0ffR9
9I7zjvO18rXyFPIU8qbypvpK48rjy3vLe8jzzzPPNQ81DznvOe89jz2POG9Ib0i/SL9Kz0rPQK9Ar0gfOB893z3fNT81PzU/NT80/zT/
Na81rzLvMu81nzWfP68/rz1fPV80L0L0QvTF9MX0c/Vz9Yb2hvYR+BH4BH4SvlK+aX5pfmx+rH61PvU+4j8iPwh/iH+YwBjAEsBSwG+
Ab4B1gLWAtQD1ANoBGgEpwSnBPUE9QQQGBQYFsgSyBEEEQQQQMBAwEKgQqBOoE6gQqBSoF/wT/BAAGAAapBqkGQ
gdCB5EIkQhCCCUIJ1AnUCVMKUwpoCmgKwQrBCqkKqQqmCqYKaAtoC68LrwvBC8ELzwvPC6oLqgvSC9ILPQw9DNcL1wv
FCsUKuwm7CV8IXwigB6AH9wb3Bu4F7gV/BX8FcQVxBboEugScA5wDxgPGAzYDNgPVAdUBYgFiAXUAdQCl/6X/Sv9K/xr/
Gv+0/rT+qf6p/hb/Fv/B/sH+3P7c/hz/HP+M/4z/JwAnAHYAdgBpAWkBNQE1AQYBBgHoAegBlAGUAZ8BnwEGAgYCswGzAd
YB1gHMAswCSQNJAw0DDQOeA54DQwNDA6sCqwL3AvcCRQJFAo8BjwF+AX4BLgIuAg8DDwMyAzID3wPfAwgFCAUbB

Google Cloud Speech API

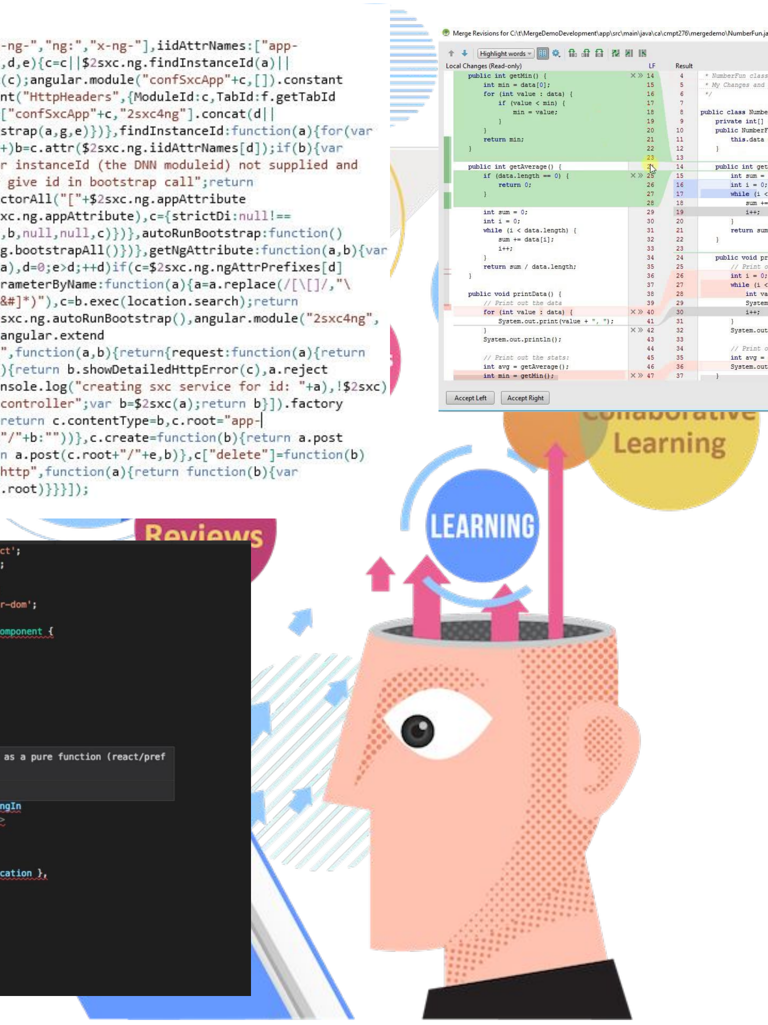

ELECTRON
express

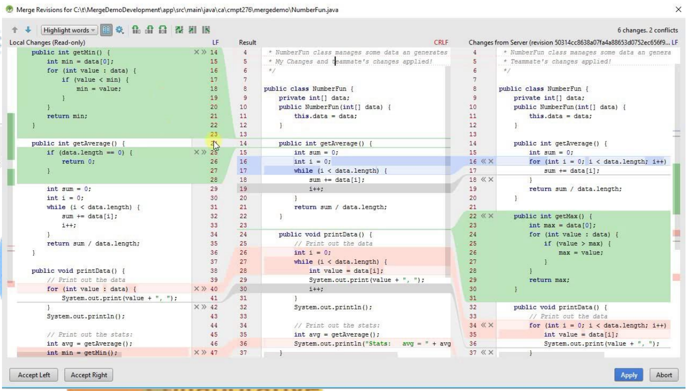
Fullstack Academy

no regrets

- Teamwork: Building something cool in 2 weeks
- Attention from Jellyvision

```
/*! ToSic_ToSxc 2015-05-17 */

$2sxc.ng={appAttribute:"sxc-app",ngAttrPrefixes:["ng-","data-ng-","ng:","x-ng-"],iidAttrNames:["app-
instanceid","data-instanceid","id"],bootstrap:function(a,b,c,d,e){c=c||$2sxc.ng.findInstanceId(a)||
$2sxc.ng.getParameterByName("mid");var f=$.ServicesFramework(c);angular.module("confSxcApp"+c,[]).constant
("AppInstanceId",c).constant("AppServiceFramework",f).constant("HttpHeaders",{ModuleId:c,TabId:f.getTabId
(),RequestVerificationToken:f.getAntiForgeryValue()});var g=["confSxcApp"+c,"2sxc4ng"].concat(d||
[b]);angular.element(document).ready(function(){angular.bootstrap(a,g,e)})},findInstanceId:function(a){for(var
b,c=angular.element(a),d=0;d<$2sxc.ng.iidAttrNames.length;d++)b=c.attr($2sxc.ng.iidAttrNames[d]);if(b){var
e=parseInt(b.toString().replace(/\D/g,""));if(!e)throw"iid or instanceId (the DNN moduleid) not supplied and
automatic lookup failed. Please set app-tag attribute iid or give id in bootstrap call";return
e}},bootstrapAll:function(a){a=a||document;var b=a.querySelectorAll("["+$2sxc.ng.appAttribute
+"]");angular.forEach(b,function(a){var b=a.getAttribute($2sxc.ng.appAttribute),c={strictDi:null!==
$2sxc.ng.getNgAttribute(a,"strict-di")};$2sxc.ng.bootstrap(a,b,null,null,c)})},autoRunBootstrap:function()
{angular&&angular.element(document).ready(function(){$2sxc.ng.bootstrapAll()})},getNgAttribute:function(a,b){var
 c,d,e=$2sxc.ng.ngAttrPrefixes.length;for(a=angular.element(a),d=0;e>d;++d)if(c=$2sxc.ng.ngAttrPrefixes[d]
+b,"string"==typeof(c=a.attr(c)))return c;return null},getParameterByName:function(a){a=a.replace(/[\[]/,"\
\[").replace(/[\]]/,"\\]");var b=new RegExp("[\\?&]"+a+"=([^&#]*)"),c=b.exec(location.search);return
null===c?"":decodeURIComponent(c[1].replace(/\+/g," "))}},$2sxc.ng.autoRunBootstrap(),angular.module("2sxc4ng",
["ng"]).config(["$httpProvider","HttpHeaders",function(a,b){angular.extend
(a.defaults.headers.common,b),a.interceptors.push(["$q","sxc",function(a,b){return{request:function(a){return
a.url=b.resolveServiceUrl(a.url),a},responseError:function(c){return b.showDetailedHttpError(c),a.reject
(c)}}}])}]).factory("sxc",["AppInstanceId",function(a){if(console.log("creating sxc service for id: "+a),!$2sxc)
throw"the Angular service 'sxc' can't find the global $2sxc controller";var b=$2sxc(a);return b}]).factory
("content",["$http",function(a){return function(b){var c={};return c.contentType=b,c.root="app-
content/"+b,c.get=c.read=function(b){return a.get(c.root+(b?"/"+b:""))},c.create=function(b){return a.post
(c.root,b)},c.update=function(b,d){var e=d||b.Id||b.id;return a.post(c.root+"/"+e,b)},c["delete"]=function(b)
{return a["delete"](c.root+"/"+b)},c}}]).factory("query",["$http",function(a){return function(b){var
c=this;c.root="app-query/"+b,c.get=function(){return a.get(c.root)}}}]);
//# sourceMappingURL=2sxc4ng.min.js.map
```




Merge Revisions for C:\t\MergeDemoDevelopment\app\src\main\java\ca\cmpt276\mergedemo\NumberFun.java
Highlight words
6 changes. 2 conflicts
Accept Left
Accept Right
Apply
Abort

Local Changes (Read-only) LF

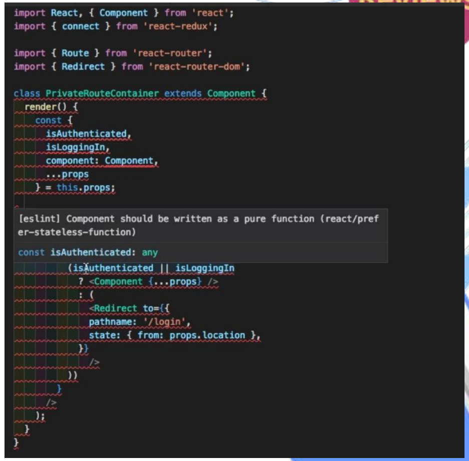
```
    public int getMin() {
        int min = data[0];
        for (int value : data) {
            if (value < min) {
                min = value;
            }
        }
        return min;
    }

    public int getAverage() {
        if (data.length == 0) {
            return 0;
        }

        int sum = 0;
        int i = 0;
        while (i < data.length) {
            sum += data[i];
            i++;
        }
        return sum / data.length;
    }

    public void printData() {
        // Print out the data
        for (int value : data) {
            System.out.print(value + ", ");
        }
        System.out.println();

        // Print out the stats:
        int avg = getAverage();
        int min = getMin();
```

Result CRLF

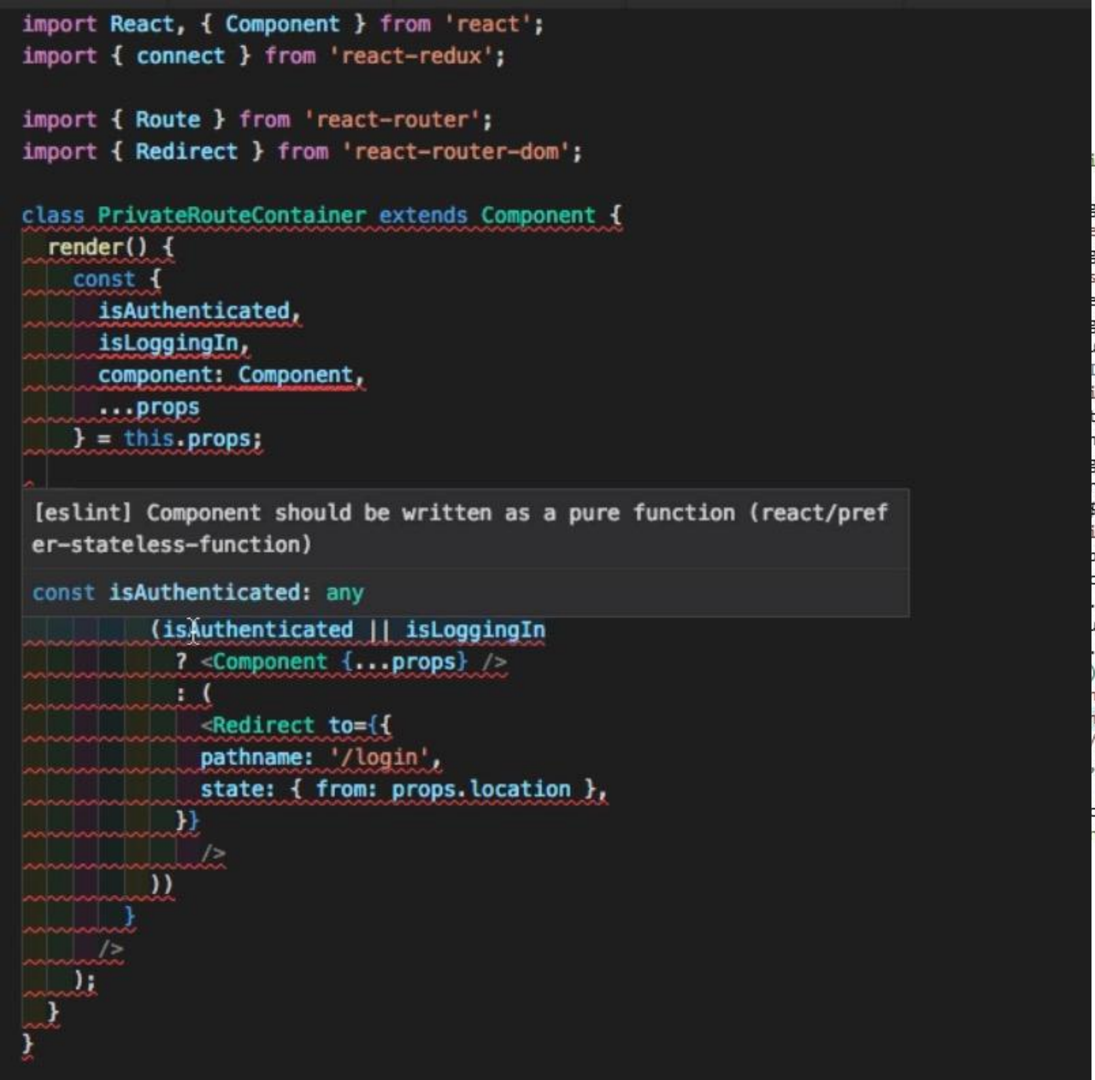
```
 * NumberFun class manages some data an generates
 * My Changes and teammate's changes applied!
 */

public class NumberFun {
    private int[] data;
    public NumberFun(int[] data) {
        this.data = data;
    }

    public int getAverage() {
        int sum = 0;
        int i = 0;
        while (i < data.length) {
            sum += data[i];
            i++;
        }
        return sum / data.length;
    }

    public void printData() {
        // Print out the data
        int i = 0;
        while (i < data.length) {
            int value = data[i];
            System.out.print(value + ", ");
            i++;
        }
        System.out.println();

        // Print out the stats:
        int avg = getAverage();
        System.out.println("Stats:   avg = " + avg
    }
```

Changes from Server (revision 50314cc8638a07fa4a88653d0752ec656f9... LF

```
 * NumberFun class manages some data an genera
 * Teammate's changes applied!
 */

public class NumberFun {
    private int[] data;
    public NumberFun(int[] data) {
        this.data = data;
    }

    public int getAverage() {
        int sum = 0;
        for (int i = 0; i < data.length; i++)
            sum += data[i];
        }
        return sum / data.length;
    }

    public int getMax() {
        int max = data[0];
        for (int value : data) {
            if (value > max) {
                max = value;
            }
        }
        return max;
    }

    public void printData() {
        // Print out the data
        for (int i = 0; i < data.length; i++)
            int value = data[i];
            System.out.print(value + ", ");
        }
```

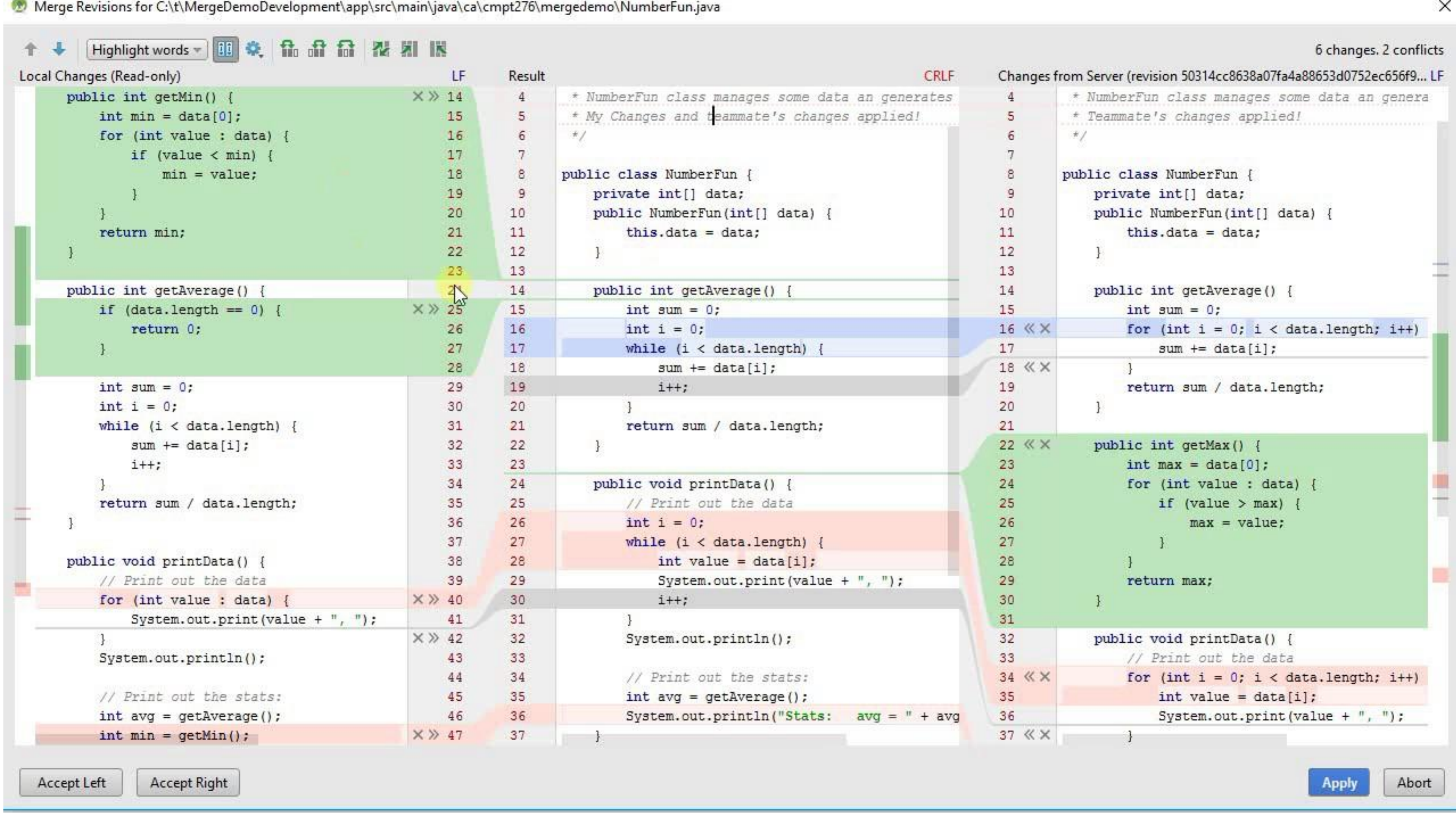


```
import React, { Component } from 'react';
import { connect } from 'react-redux';

import { Route } from 'react-router';
import { Redirect } from 'react-router-dom';

class PrivateRouteContainer extends Component {
  render() {
    const {
      isAuthenticated,
      isLoggingIn,
      component: Component,
      ...props
    } = this.props;

[eslint] Component should be written as a pure function (react/pref
er-stateless-function)

const isAuthenticated: any
          (isAuthenticated || isLoggingIn
            ? <Component {...props} />
            : (
              <Redirect to={{
              pathname: '/login',
              state: { from: props.location },
            }}
              />
          ))
        }
      />
    );
  }
}
```



```
import React, { Component } from 'react';
import { connect } from 'react-redux';

import { Route } from 'react-router';
import { Redirect } from 'react-router-dom';

class PrivateRouteContainer extends Component {
  render() {
    const {
      isAuthenticated,
      isLoggingIn,
      component: Component,
      ...props
    } = this.props;

[eslint] Component should be written as a pure function (react/prefer-stateless-function)

const isAuthenticated: any
          (isAuthenticated || isLoggingIn
            ? <Component {...props} />
            : (
              <Redirect to={{
              pathname: '/login',
              state: { from: props.location },
            }}
              />
          ))
        }
      />
    );
  }
}
```

```
ic_ToSxc 2015-05-17 */

g={appAttribute:"sxc-app",ngAttrPrefixes:["ng-","data-ng-","ng:","x-ng-"],iidAttrN
eid","data-instanceid","id"],bootstrap:function(a,b,c,d,e){c=c||$2sxc.ng.findInsta
g.getParameterByName("mid");var f=$.ServicesFramework(c);angular.module("confSxcAp
stanceId",c).constant("AppServiceFramework",f).constant("HttpHeaders",{ModuleId:c,
estVerificationToken:f.getAntiForgeryValue()});var g=["confSxcApp"+c,"2sxc4ng"].co
gular.element(document).ready(function(){angular.bootstrap(a,g,e)})},findInstanceI
ular.element(a),d=0;d<$2sxc.ng.iidAttrNames.length;d++)b=c.attr($2sxc.ng.iidAttrNa
Int(b.toString().replace(/\D/g,""));if(!e)throw"iid or instanceId (the DNN modulei
ic lookup failed. Please set app-tag attribute iid or give id in bootstrap call";r
tstrapAll:function(a){a=a||document;var b=a.querySelectorAll("["+$2sxc.ng.appAttri
ngular.forEach(b,function(a){var b=a.getAttribute($2sxc.ng.appAttribute),c={strict
g.getNgAttribute(a,"strict-di")};$2sxc.ng.bootstrap(a,b,null,null,c)})},autoRunBoo
r&&angular.element(document).ready(function(){$2sxc.ng.bootstrapAll()})},getNgAttr
$2sxc.ng.ngAttrPrefixes.length;for(a=angular.element(a),d=0;e>d;++d)if(c=$2sxc.ng.
ing"==typeof(c=a.attr(c)))return c;return null},getParameterByName:function(a){a=a
place(/[\]]/,"\\]");var b=new RegExp("[\\?&]"+a+"=([^&#]*)"),c=b.exec(location.sea
:?"":decodeURIComponent(c[1].replace(/\+/g," "))}},$2sxc.ng.autoRunBootstrap(),ang
.config(["$httpProvider","HttpHeaders",function(a,b){angular.extend
ults.headers.common,b),a.interceptors.push(["$q","sxc",function(a,b){return{reques
.resolveServiceUrl(a.url),a},responseError:function(c){return b.showDetailedHttpEr
)}]).factory("sxc",["AppInstanceId",function(a){if(console.log("creating sxc servi
he Angular service 'sxc' can't find the global $2sxc controller";var b=$2sxc(a);re
nt",["$http",function(a){return function(b){var c={};return c.contentType=b,c.root
/"+b,c.get=c.read=function(b){return a.get(c.root+(b?"/"+b:""))},c.create=function
,b)},c.update=function(b,d){var e=d||b.Id||b.id;return a.post(c.root+"/"+e,b)},c["
 a["delete"](c.root+"/"+b)},c}}]).factory("query",["$http",function(a){return func
:.root="app-query/"+b,c.get=function(){return a.get(c.root)}}}]);
rceMappingURL=2sxc4ng.min.js.map
```



Merge Revisions for C:\t\MergeDemoDevelopment\app\src\main\java\ca\cmpt276\mergedemo\NumberFun.java

Highlight words

6 changes. 2 conflicts

Local Changes (Read-only) LF

```
    public int getMin() {
        int min = data[0];
        for (int value : data) {
            if (value < min) {
                min = value;
            }
        }
        return min;
    }

    public int getAverage() {
        if (data.length == 0) {
            return 0;
        }

        int sum = 0;
        int i = 0;
        while (i < data.length) {
            sum += data[i];
            i++;
        }
        return sum / data.length;
    }

    public void printData() {
        // Print out the data
        for (int value : data) {
            System.out.print(value + ", ");
        }
        System.out.println();

        // Print out the stats:
        int avg = getAverage();
        int min = getMin();
```

Result CRLF

```
 * NumberFun class manages some data an generates
 * My Changes and teammate's changes applied!
 */

public class NumberFun {
    private int[] data;
    public NumberFun(int[] data) {
        this.data = data;
    }

    public int getAverage() {
        int sum = 0;
        int i = 0;
        while (i < data.length) {
            sum += data[i];
            i++;
        }
        return sum / data.length;
    }

    public void printData() {
        // Print out the data
        int i = 0;
        while (i < data.length) {
            int value = data[i];
            System.out.print(value + ", ");
            i++;
        }
        System.out.println();

        // Print out the stats:
        int avg = getAverage();
        System.out.println("Stats:   avg = " + avg
    }
```

Changes from Server (revision 50314cc8638a07fa4a88653d0752ec656f9... LF

```
 * NumberFun class manages some data an genera
 * Teammate's changes applied!
 */

public class NumberFun {
    private int[] data;
    public NumberFun(int[] data) {
        this.data = data;
    }

    public int getAverage() {
        int sum = 0;
        for (int i = 0; i < data.length; i++)
            sum += data[i];
        }
        return sum / data.length;
    }

    public int getMax() {
        int max = data[0];
        for (int value : data) {
            if (value > max) {
                max = value;
            }
        }
        return max;
    }

    public void printData() {
        // Print out the data
        for (int i = 0; i < data.length; i++)
            int value = data[i];
            System.out.print(value + ", ");
        }
```

Accept Left | Accept Right | Apply | Abort

This code's a mess

...but I had a pretty good excuse

- *console.log*...everywhere
- Silly function names like `*resetCurrDiv*`
- Code areas that cover multiple domains (spaghetti code)

Time Crunch Sacrifices

- Documentation
- Interfaces
- Tests
- Browsability
- Performance
- Security
- Error-handling
- A “product” perspective

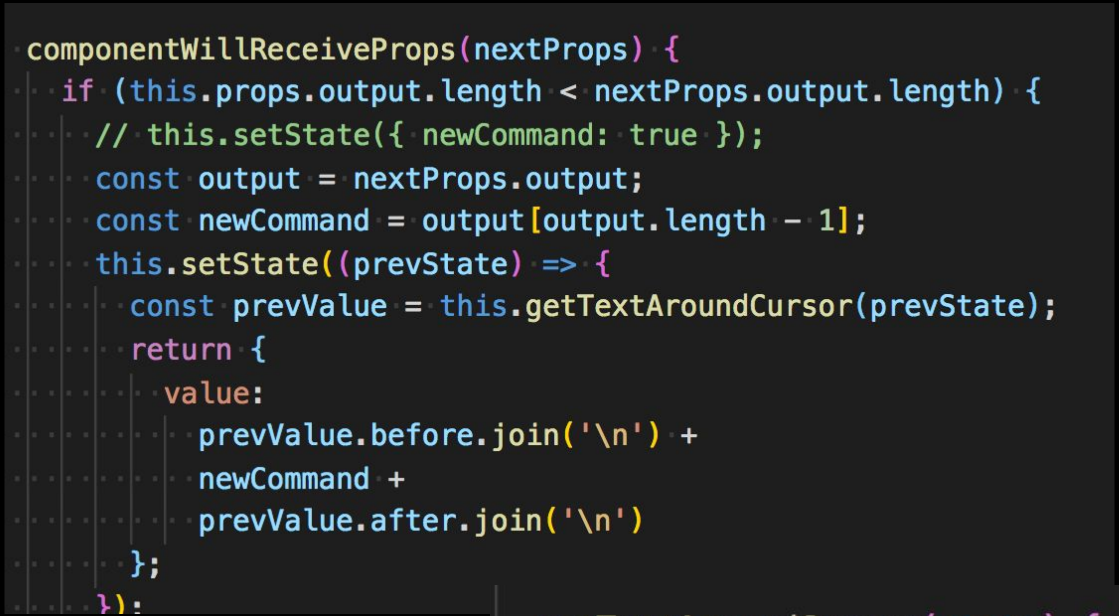
Documentation

Problems

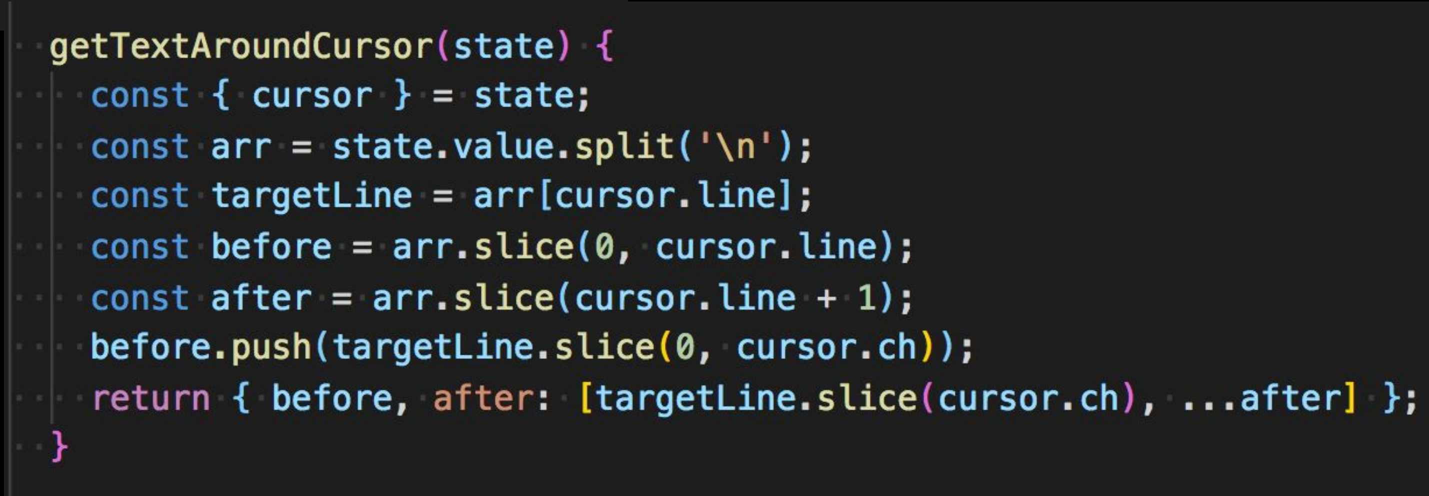
- The mindset of *"no one else is working in this file"*
- That thing I couldn’t remember in <Code> Editor

Solutions

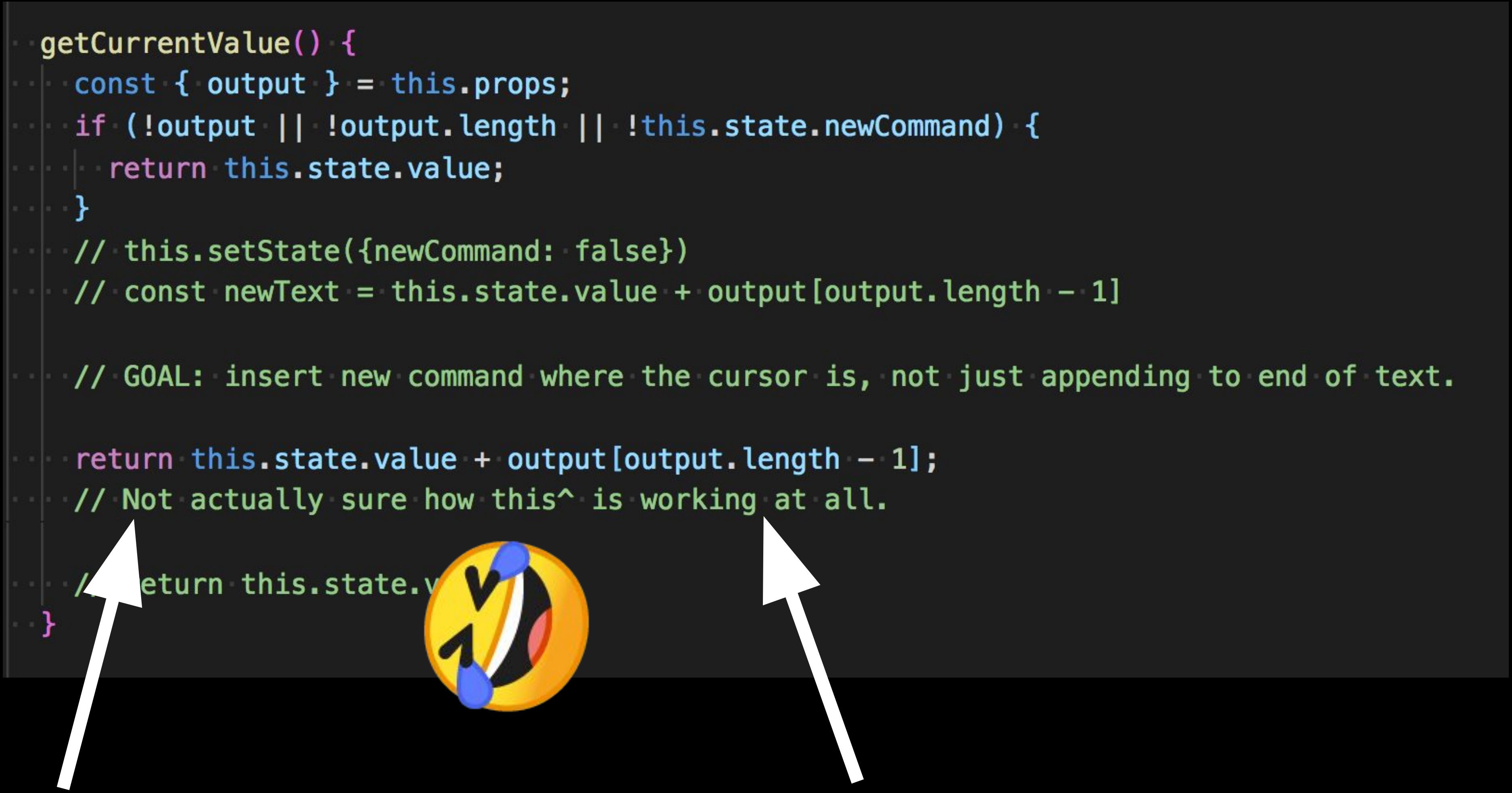
- Clearer comments
- Better variable names
- JSDocs on functions: description, `@param`s, `@returns`



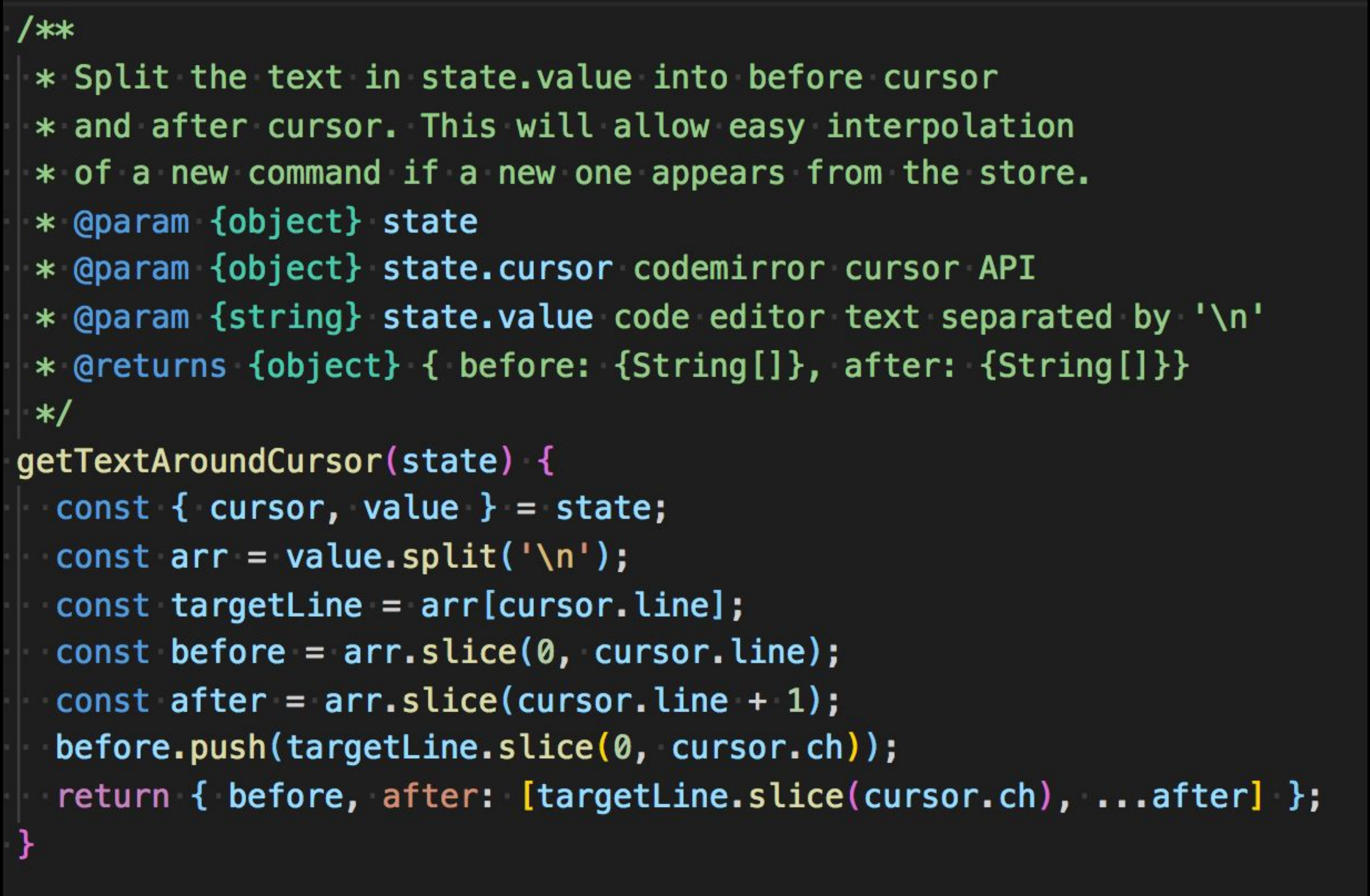
```
componentWillReceiveProps(nextProps) {
  if (this.props.output.length < nextProps.output.length) {
    // this.setState({ newCommand: true });
    const output = nextProps.output;
    const newCommand = output[output.length - 1];
    this.setState((prevState) => {
      const prevValue = this.getTextAroundCursor(prevState);
      return {
        value:
          prevValue.before.join('\n') +
          newCommand +
          prevValue.after.join('\n')
      };
    });
```



```
getTextAroundCursor(state) {
  const { cursor } = state;
  const arr = state.value.split('\n');
  const targetLine = arr[cursor.line];
  const before = arr.slice(0, cursor.line);
  const after = arr.slice(cursor.line + 1);
  before.push(targetLine.slice(0, cursor.ch));
  return { before, after: [targetLine.slice(cursor.ch), ...after] };
}
```



```
getCurrentValue() {
  const { output } = this.props;
  if (!output || !output.length || !this.state.newCommand) {
    return this.state.value;
  }
  // this.setState({newCommand: false})
  // const newText = this.state.value + output[output.length - 1]

  // GOAL: insert new command where the cursor is, not just appending to end of text.

  return this.state.value + output[output.length - 1];
  // Not actually sure how this^ is working at all.

  // return this.state.v
}
```



```
/**
 * Split the text in state.value into before cursor
 * and after cursor. This will allow easy interpolation
 * of a new command if a new one appears from the store.
 * @param {object} state
 * @param {object} state.cursor codemirror cursor API
 * @param {string} state.value code editor text separated by '\n'
 * @returns {object} { before: {String[]}, after: {String[]}}
 */
getTextAroundCursor(state) {
  const { cursor, value } = state;
  const arr = value.split('\n');
  const targetLine = arr[cursor.line];
  const before = arr.slice(0, cursor.line);
  const after = arr.slice(cursor.line + 1);
  before.push(targetLine.slice(0, cursor.ch));
  return { before, after: [targetLine.slice(cursor.ch), ...after] };
}
```

Interfaces

Problems

- Lack of consistency = higher cognitive load
- APIs I could’ve abstracted out (word methods, etc)
- Dictionary functions doing too much (validation, etc)

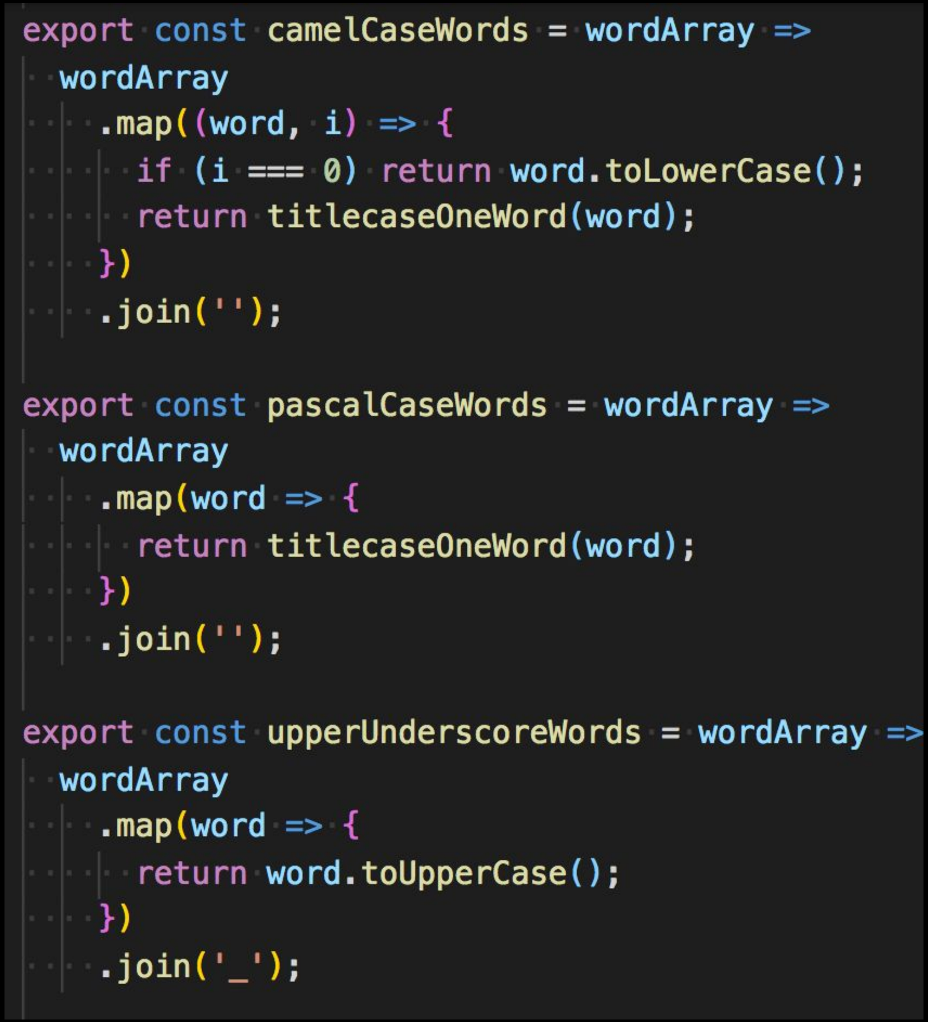
Solutions

- Word methods have a transformWords wrapper
- Making the Dictionary have only one job - to supply unique values for each template

Old

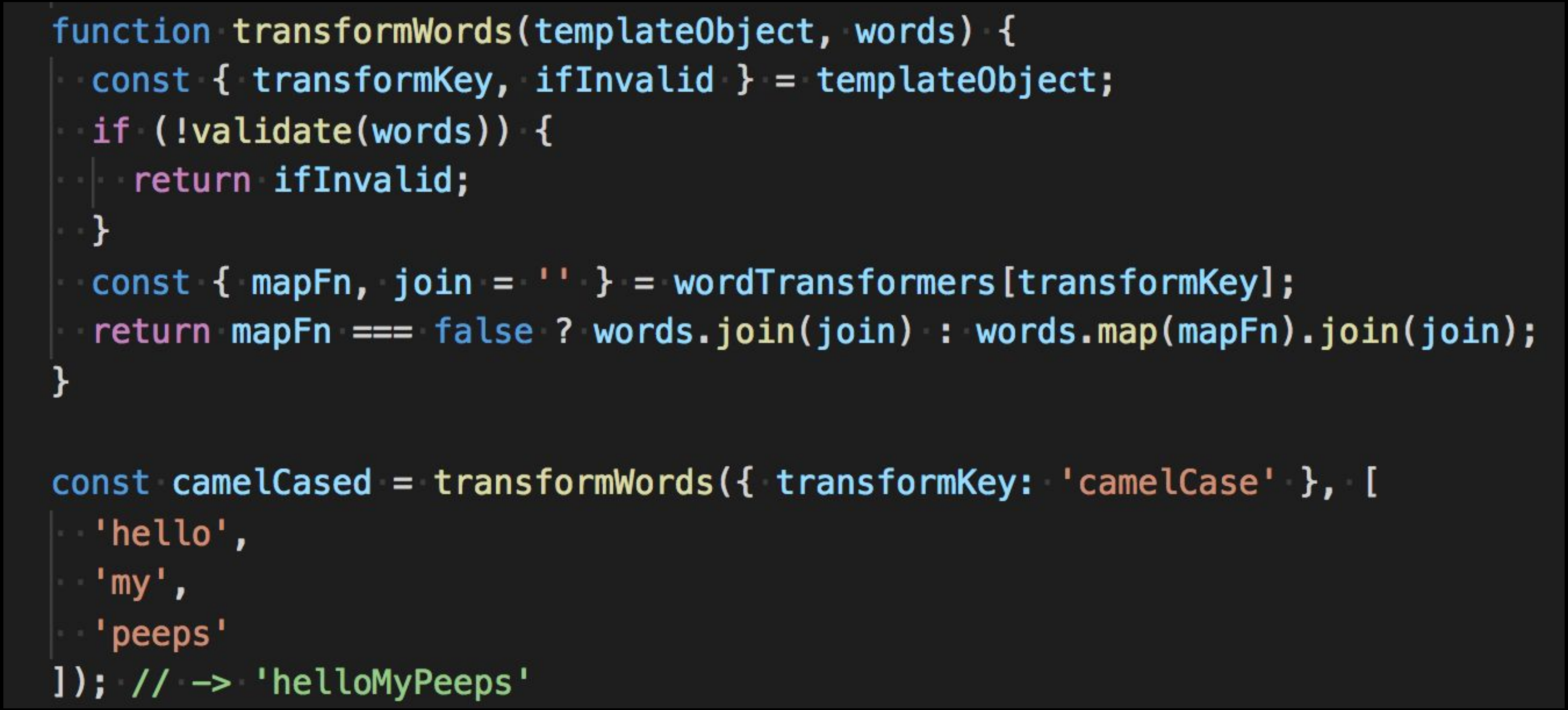
- each word method takes the same type of input, maps and joins it

For reference:
camelCase, PascalCase, SCREAMING_SNAKE_CASE (upperUnderscoreWords)



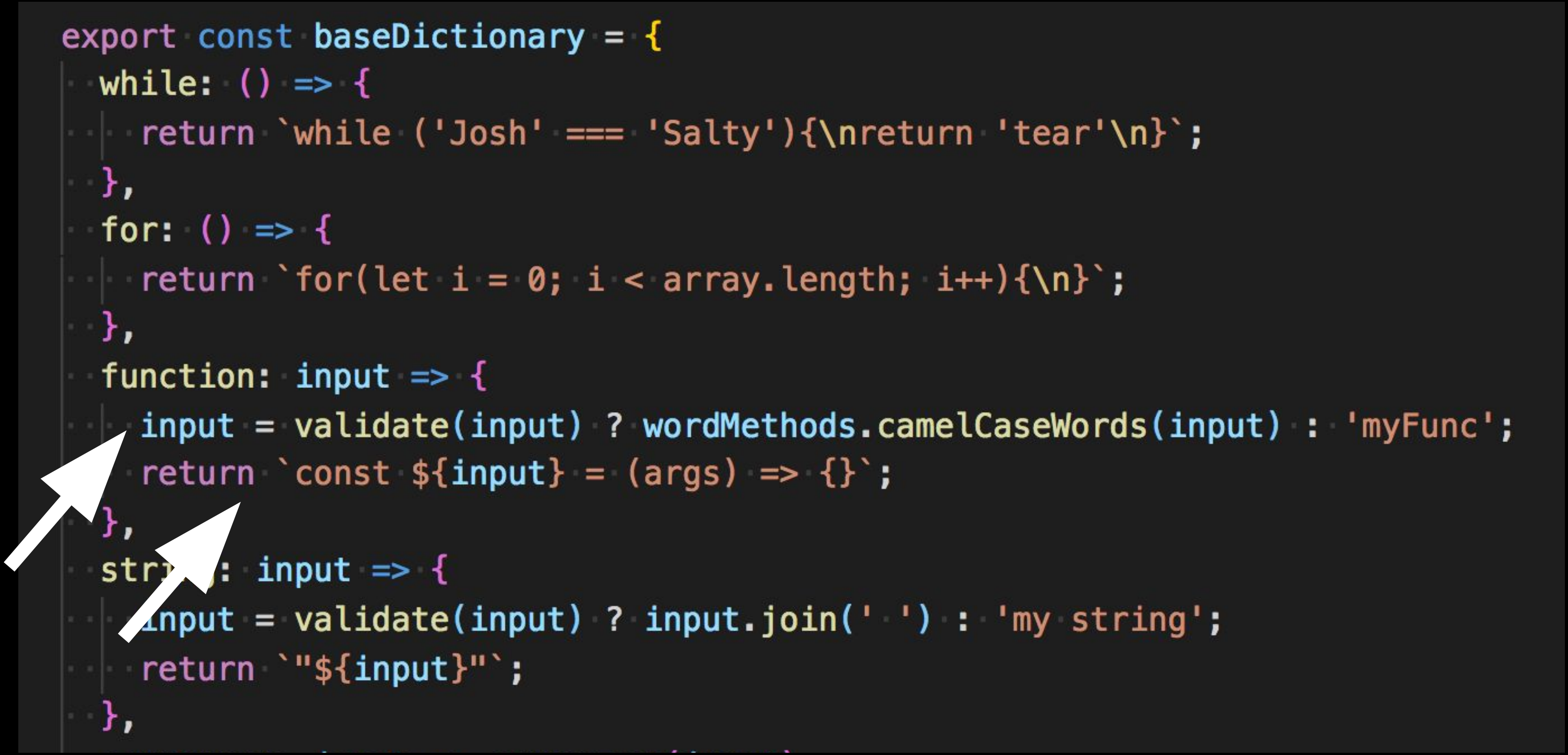
```
export const camelCaseWords = wordArray =>
  wordArray
    .map((word, i) => {
      if (i === 0) return word.toLowerCase();
      return titlecaseOneWord(word);
    })
    .join('');

export const pascalCaseWords = wordArray =>
  wordArray
    .map(word => {
      return titlecaseOneWord(word);
    })
    .join('');

export const upperUnderscoreWords = wordArray =>
  wordArray
    .map(word => {
      return word.toUpperCase();
    })
    .join('_');
```

Better: a new *transformWords* function that does the word transformation



```javascript
function transformWords(templateObject, words) {
  const { transformKey, ifInvalid } = templateObject;
  if (!validate(words)) {
    return ifInvalid;
  }
  const { mapFn, join = '' } = wordTransformers[transformKey];
  return mapFn === false ? words.join(join) : words.map(mapFn).join(join);
}

const camelCased = transformWords({ transformKey: 'camelCase' }, [
  'hello',
  'my',
  'peeps'
]); // -> 'helloMyPeeps'
```

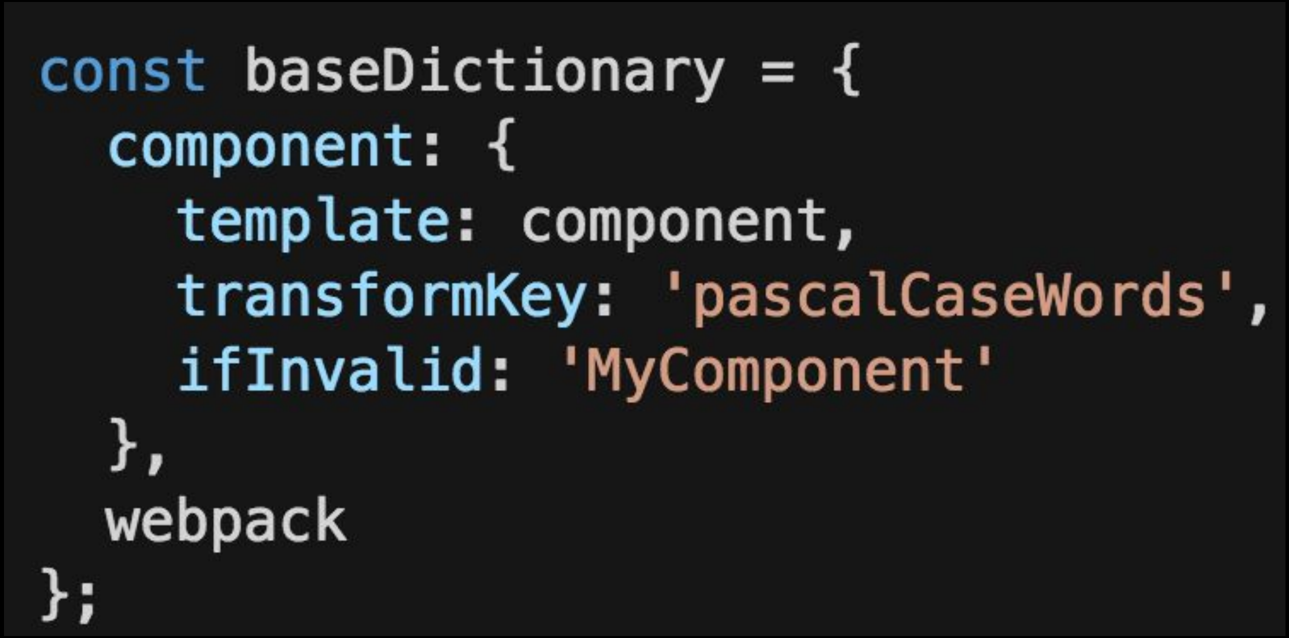
Old - Dictionary included templates and validation (too many jobs)



```
export const baseDictionary = {
  while: () => {
    return `while ('Josh' === 'Salty'){\nreturn 'tear'\n}`;
  },
  for: () => {
    return `for(let i = 0; i < array.length; i++){\n}`;
  },
  function: input => {
    input = validate(input) ? wordMethods.camelCaseWords(input) : 'myFunc';
    return `const ${input} = (args) => {}`;
  },
  string: input => {
    input = validate(input) ? input.join(' ') : 'my string';
    return `"${input}"`;
  },
```

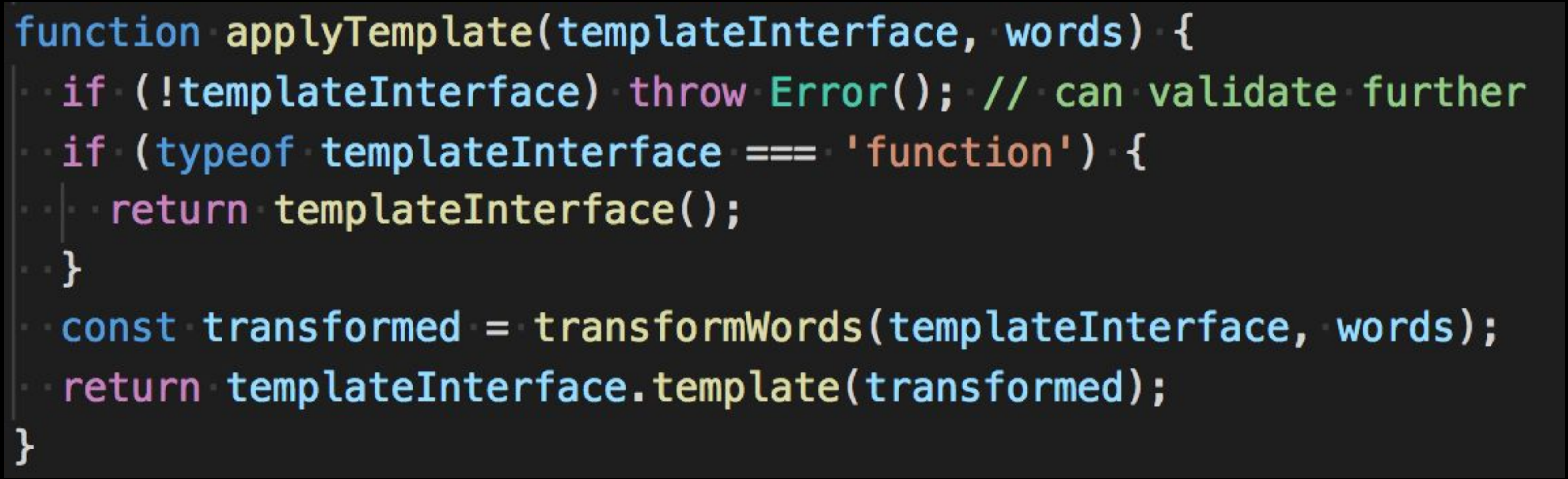
New

- Dictionary has a consistent API with expected object properties
- Puts templates, *ifInvalid*, and word transformers together (if needed)

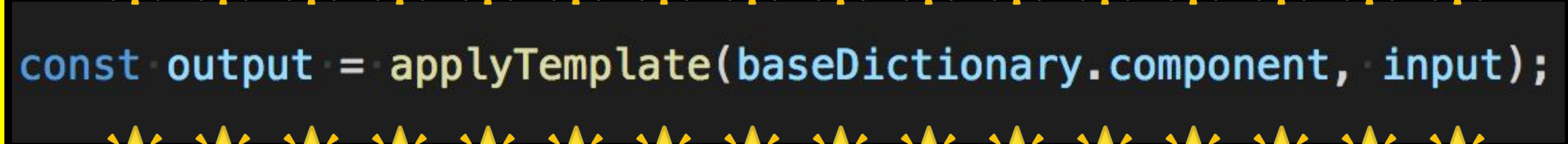


```
const baseDictionary = {
  component: {
    template: component,
    transformKey: 'pascalCaseWords',
    ifInvalid: 'MyComponent'
  },
  webpack
};
```

Now it’s easy to apply a template!



```
function applyTemplate(templateInterface, words) {
  if (!templateInterface) throw Error(); // can validate further
  if (typeof templateInterface === 'function') {
    return templateInterface();
  }
  const transformed = transformWords(templateInterface, words);
  return templateInterface.template(transformed);
}
```



```
const output = applyTemplate(baseDictionary.component, input);
```

Tests

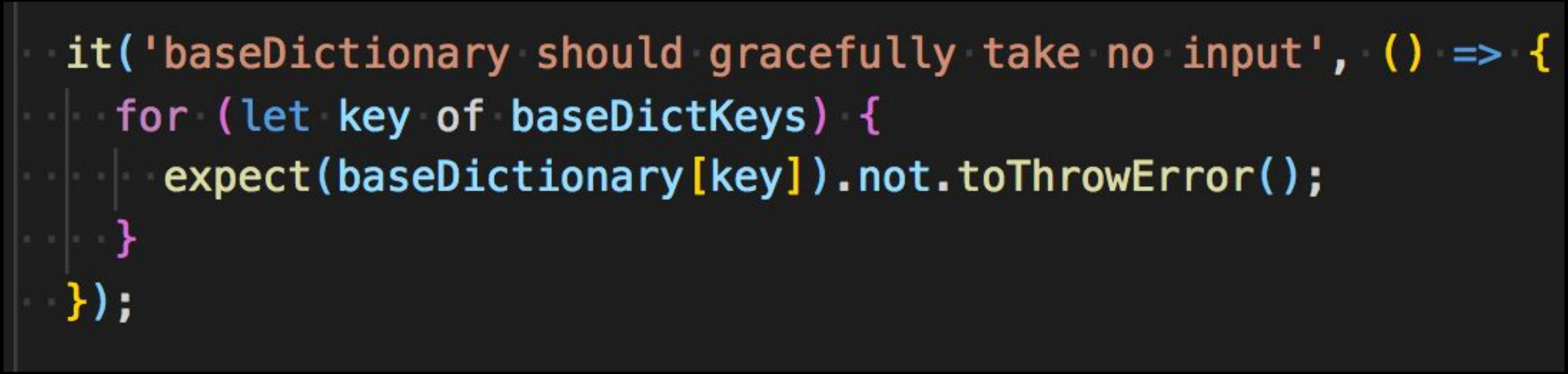
Problems

- Tests are too abstract and have unhelpful assertions
- Definitive tests are missing

Solutions

- Test for exact expected output
- TDD: Write a complete set of test specs ("*it*" blocks)

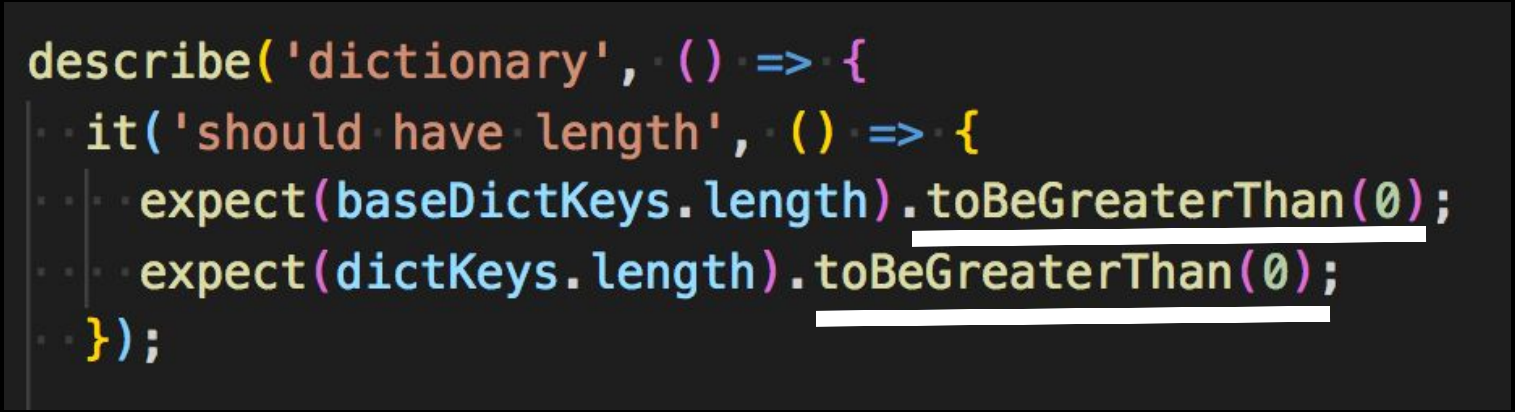
Good...



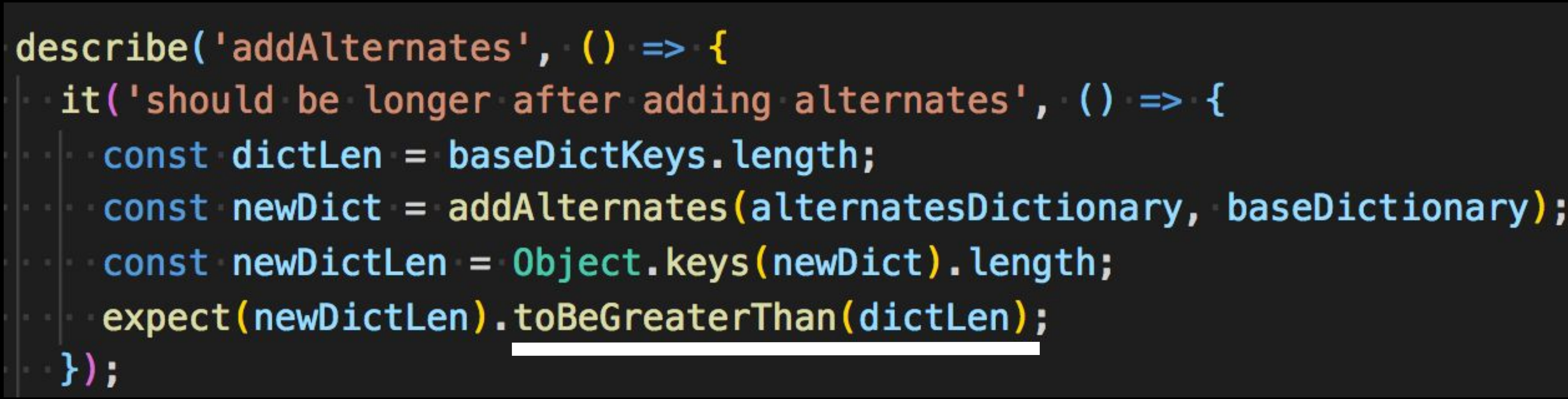
```
  it('baseDictionary should gracefully take no input', () => {
    for (let key of baseDictKeys) {
      expect(baseDictionary[key]).not.toThrowError();
    }
  });
```

...but what about expected output?

bad



```
describe('dictionary', () => {
  it('should have length', () => {
    expect(baseDictKeys.length).toBeGreaterThan(0);
    expect(dictKeys.length).toBeGreaterThan(0);
  });
```



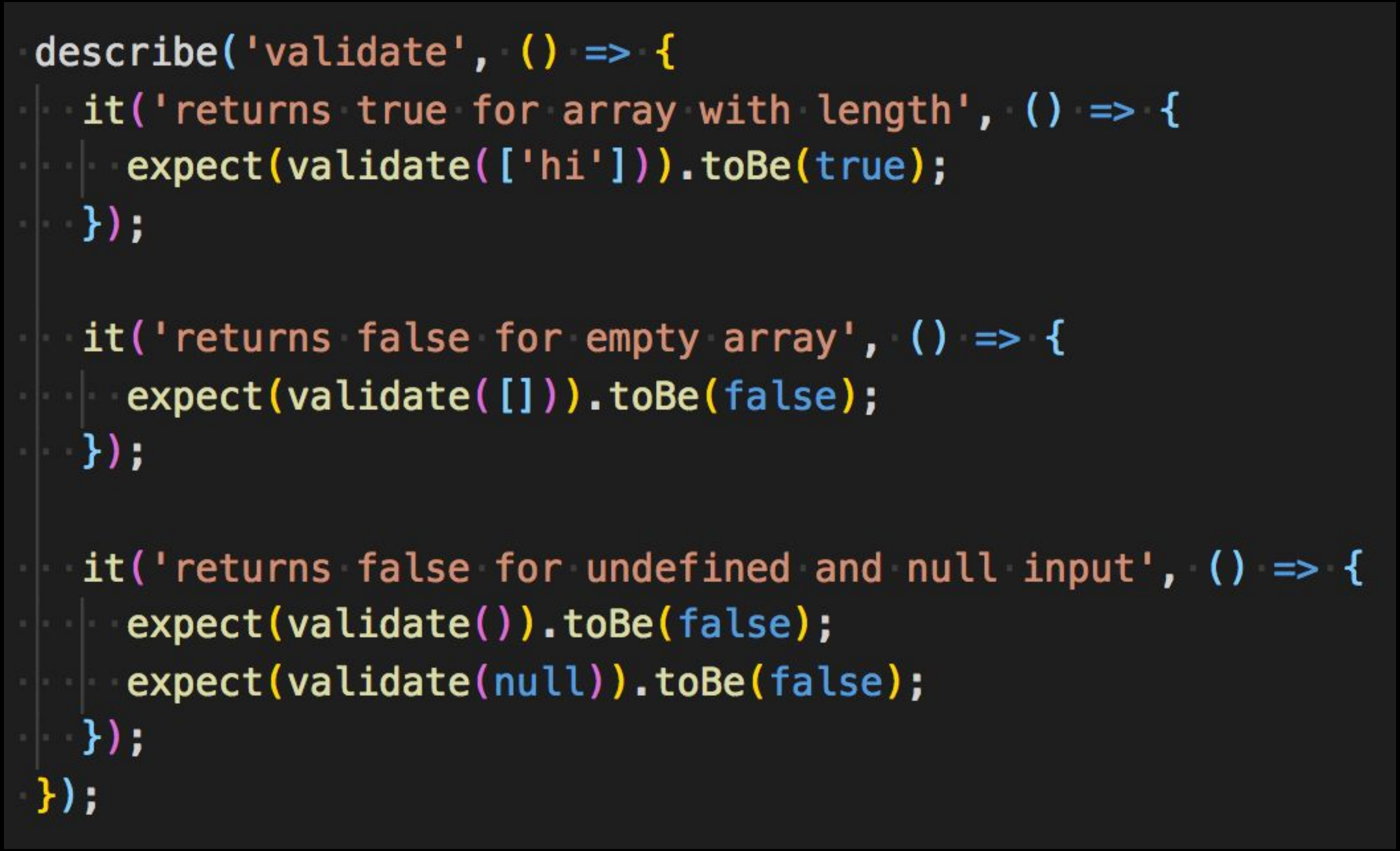
```
describe('addAlternates', () => {
  it('should be longer after adding alternates', () => {
    const dictLen = baseDictKeys.length;
    const newDict = addAlternates(alternatesDictionary, baseDictionary);
    const newDictLen = Object.keys(newDict).length;
    expect(newDictLen).toBeGreaterThan(dictLen);
  });
```

Verify all dictionary keys ✔️

describe(‘dictionary’)

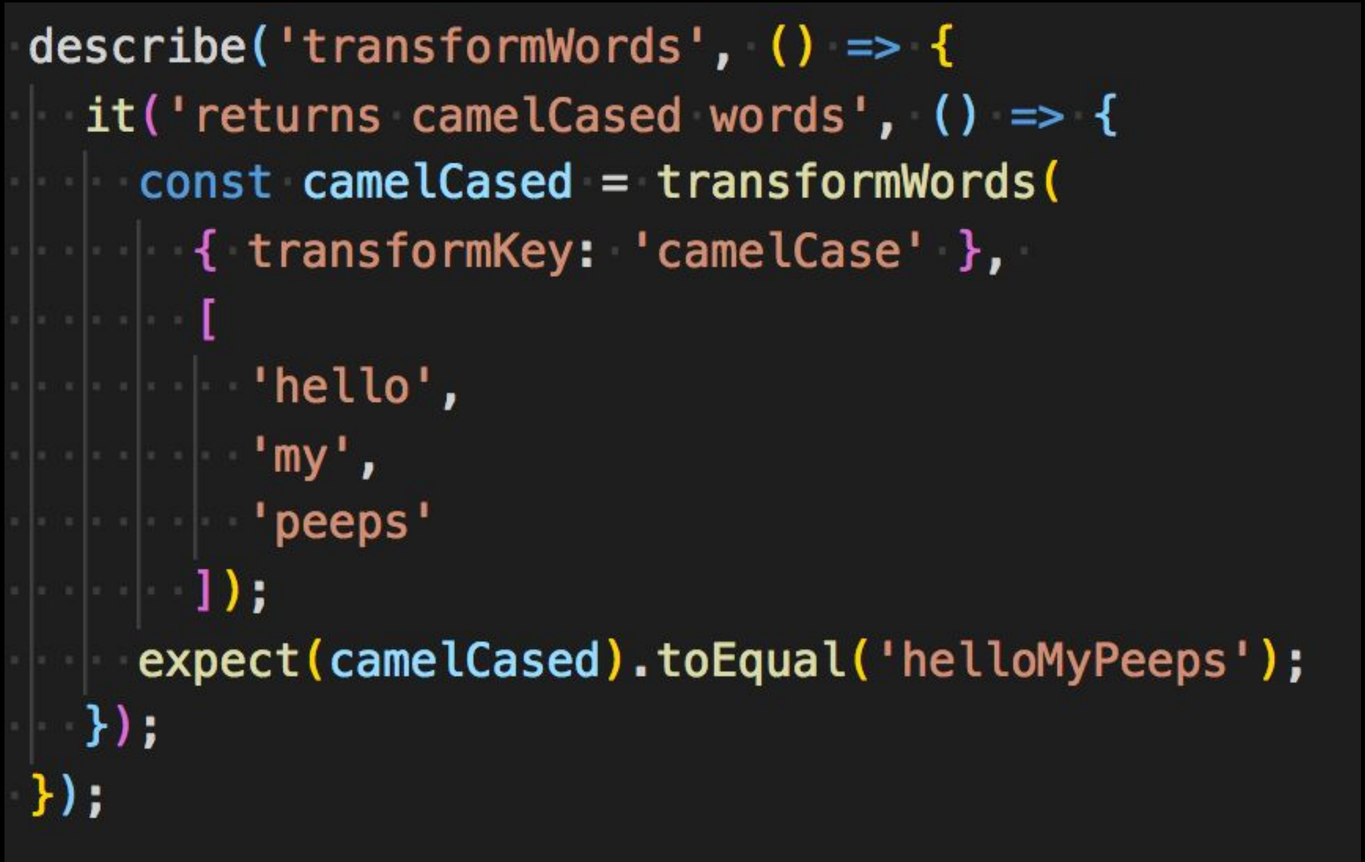
- it('should have 22 commands')
- it('should contain the baseDictionary’)
- it('should contain urls commands with functions as values')
- it('should contain alternates')
- it('each command should have the expected function or object API')
    - Each value is either a function or an object
    - Each object should have the expected keys 'template', 'ifInvalid', 'transformKey'
    - Keys should by types ‘function’, ‘string’, and ‘string’, respectively

test *validate* ✔️



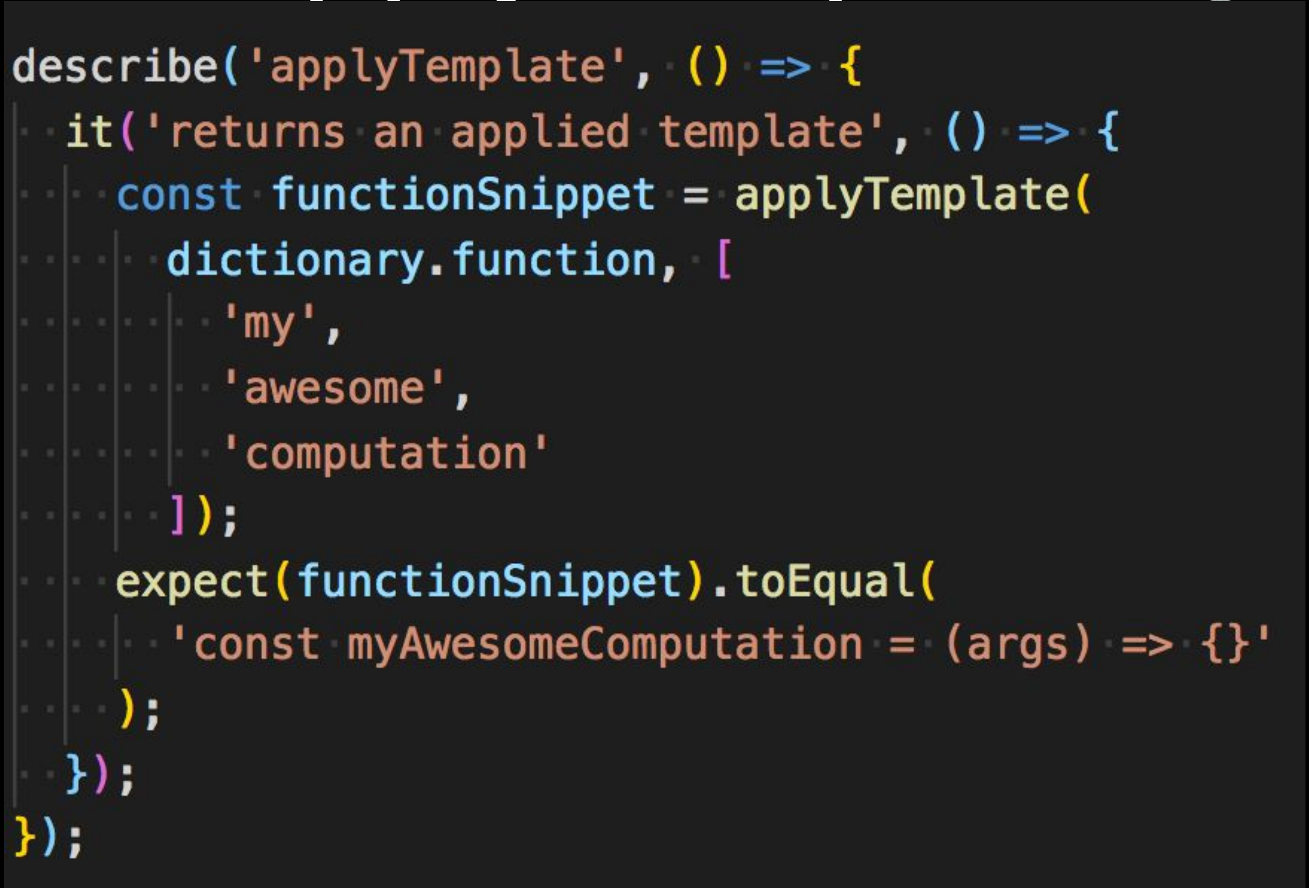
```
describe('validate', () => {
  it('returns true for array with length', () => {
    expect(validate(['hi'])).toBe(true);
  });

  it('returns false for empty array', () => {
    expect(validate([])).toBe(false);
  });

  it('returns false for undefined and null input', () => {
    expect(validate()).toBe(false);
    expect(validate(null)).toBe(false);
  });
});
```

test *transformWords* ✔️



```
describe('transformWords', () => {
  it('returns camelCased words', () => {
    const camelCased = transformWords(
      { transformKey: 'camelCase' },
      [
        'hello',
        'my',
        'peeps'
      ]);
    expect(camelCased).toEqual('helloMyPeeps');
  });
});
```

test *applyTemplate* ✔️



```
describe('applyTemplate', () => {
  it('returns an applied template', () => {
    const functionSnippet = applyTemplate(
      dictionary.function, [
        'my',
        'awesome',
        'computation'
      ]);
    expect(functionSnippet).toEqual(
      'const myAwesomeComputation = (args) => {}'
    );
  });
});
```

Browsability and Performance

Problems

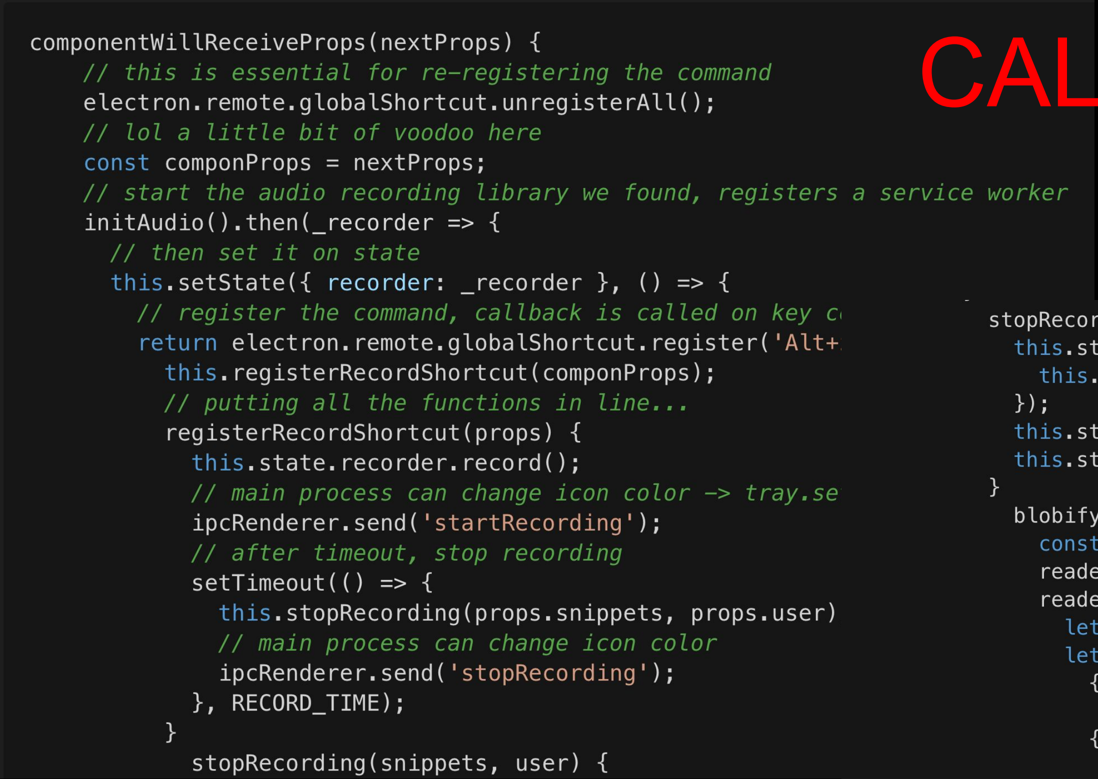
- Callback hell in <*Mic* />
- Missed performance optimization opportunities

Solutions

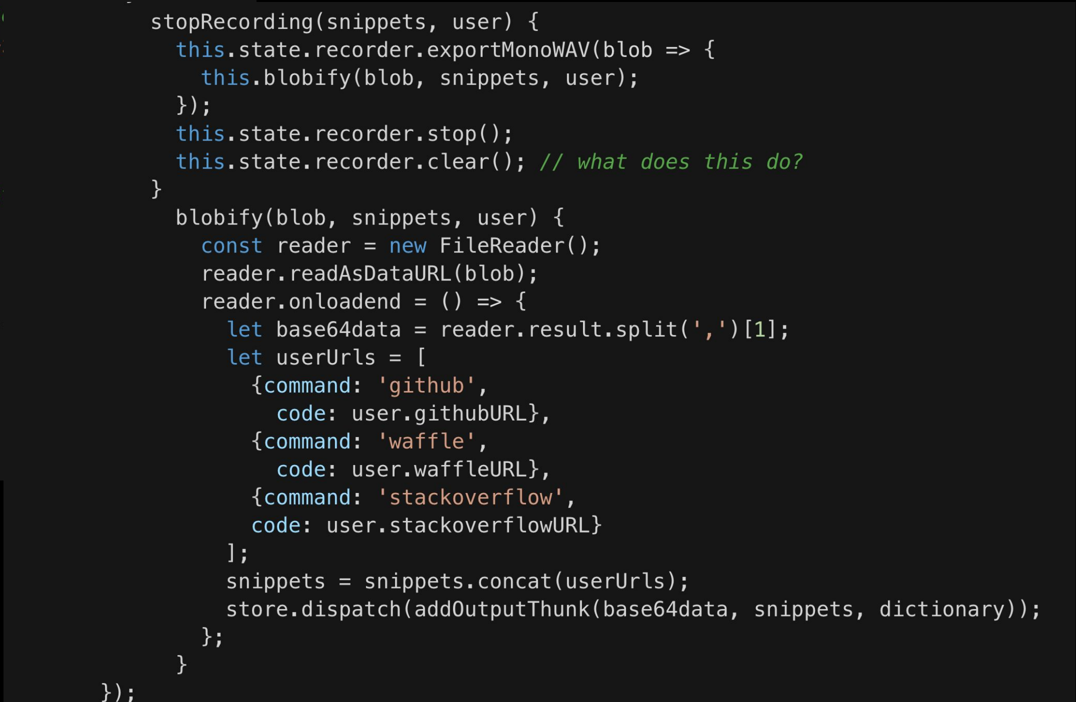
- Modularize where possible
- Check if global or component state *needs* to update

Mic

CALLBACK 🔥 HELL



```
componentWillReceiveProps(nextProps) {
    // this is essential for re-registering the command
    electron.remote.globalShortcut.unregisterAll();
    // lol a little bit of voodoo here
    const componProps = nextProps;
    // start the audio recording library we found, registers a service worker
    initAudio().then(_recorder => {
      // then set it on state
      this.setState({ recorder: _recorder }, () => {
        // register the command, callback is called on key c
        return electron.remote.globalShortcut.register('Alt+
          this.registerRecordShortcut(componProps);
          // putting all the functions in line...
          registerRecordShortcut(props) {
            this.state.recorder.record();
            // main process can change icon color -> tray.se
            ipcRenderer.send('startRecording');
            // after timeout, stop recording
            setTimeout(() => {
              this.stopRecording(props.snippets, props.user)
              // main process can change icon color
              ipcRenderer.send('stopRecording');
            }, RECORD_TIME);
          }
            stopRecording(snippets, user) {
```



```
  stopRecording(snippets, user) {
    this.state.recorder.exportMonoWAV(blob => {
      this.blobify(blob, snippets, user);
    });
    this.state.recorder.stop();
    this.state.recorder.clear(); // what does this do?
  }
    blobify(blob, snippets, user) {
      const reader = new FileReader();
      reader.readAsDataURL(blob);
      reader.onloadend = () => {
        let base64data = reader.result.split(',')[1];
        let userUrls = [
          {command: 'github',
            code: user.githubURL},
          {command: 'waffle',
            code: user.waffleURL},
          {command: 'stackoverflow',
          code: user.stackoverflowURL}
        ];
        snippets = snippets.concat(userUrls);
        store.dispatch(addOutputThunk(base64data, snippets, dictionary));
      };
    }
});
```

1. Unregister all app shortcuts
2. Initialize an audio recorder
3. if successful, set it on state
4. Register alt+z
5. call Start Recording on recorder and send message to main process
6. Set timeout for recording time
7. When complete, create WAV file and send message to main process
8. Stop recorder and clear recorder
9. Read WAV file as blob
10. Convert WAV file to base64
11. Combine hardcoded url snippets with props.snippets
12. Dispatch thunk which calls Google Speech API

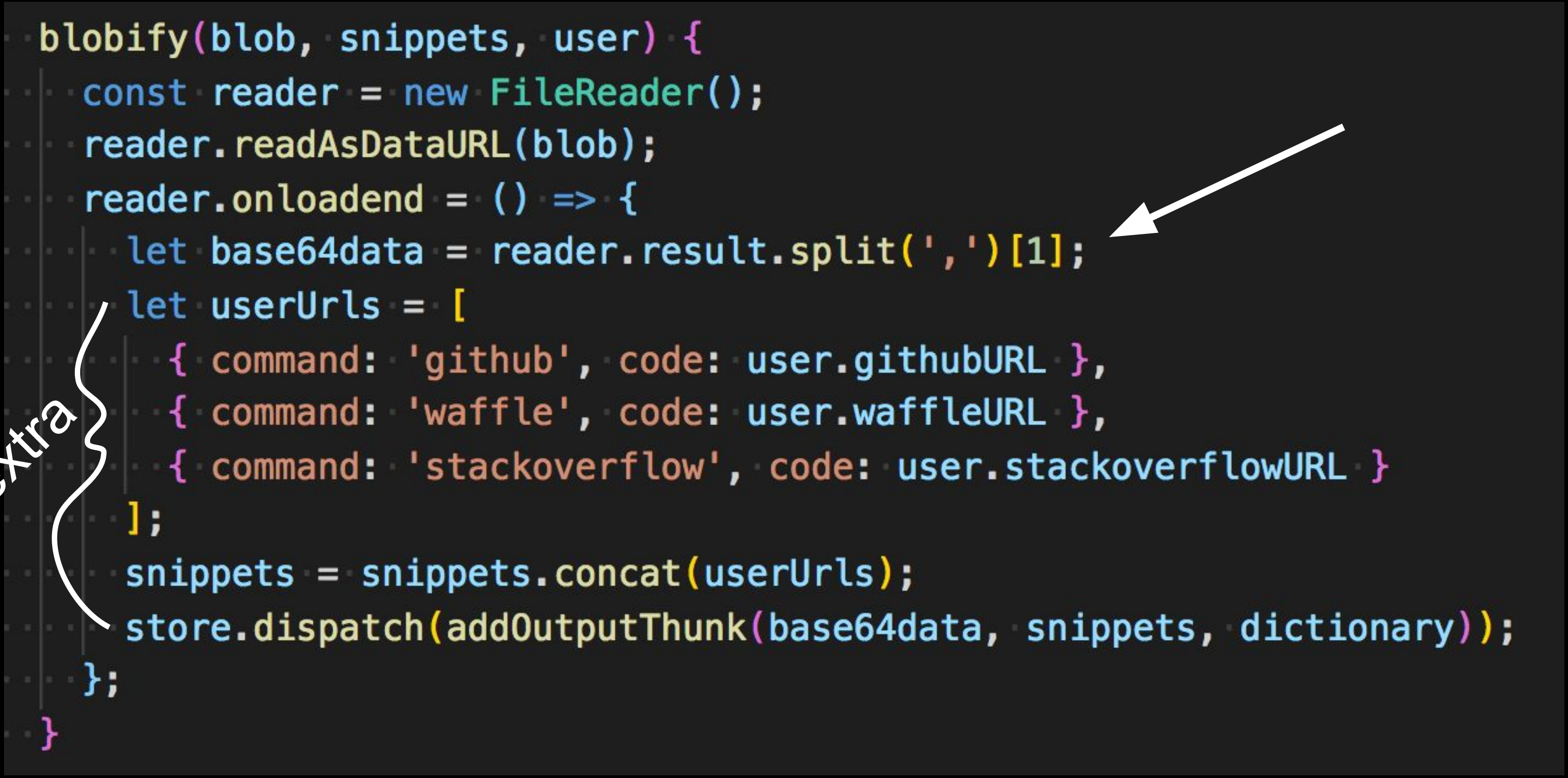
Missed performance improvements

- We didn't do a props comparison - what if there was no need to run all these functions again?
- unregister ALL global Electron shortcuts?

Browsability improvements and code cleanup opportunities

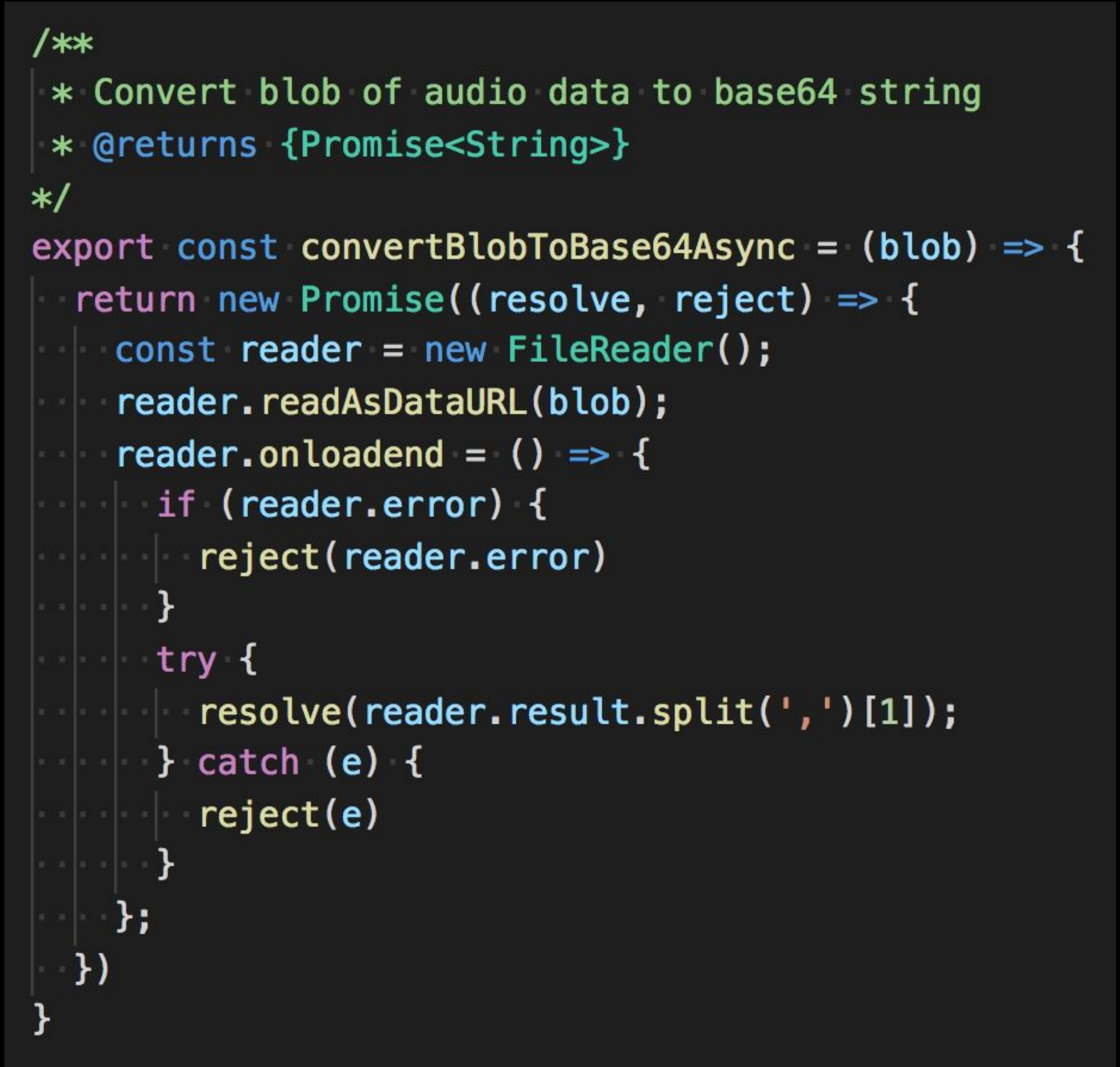
- could've done this.recorder instead of setting state (probably only had to initialize it once)
- there's no .catch if initAudio fails!
- Hard-coded in userUrls (not sure why I didn't reuse the defined dictionary)
- **could've pulled out "pure" functions to reduce the complexity and add unit tests**

Let’s fix `blobify`

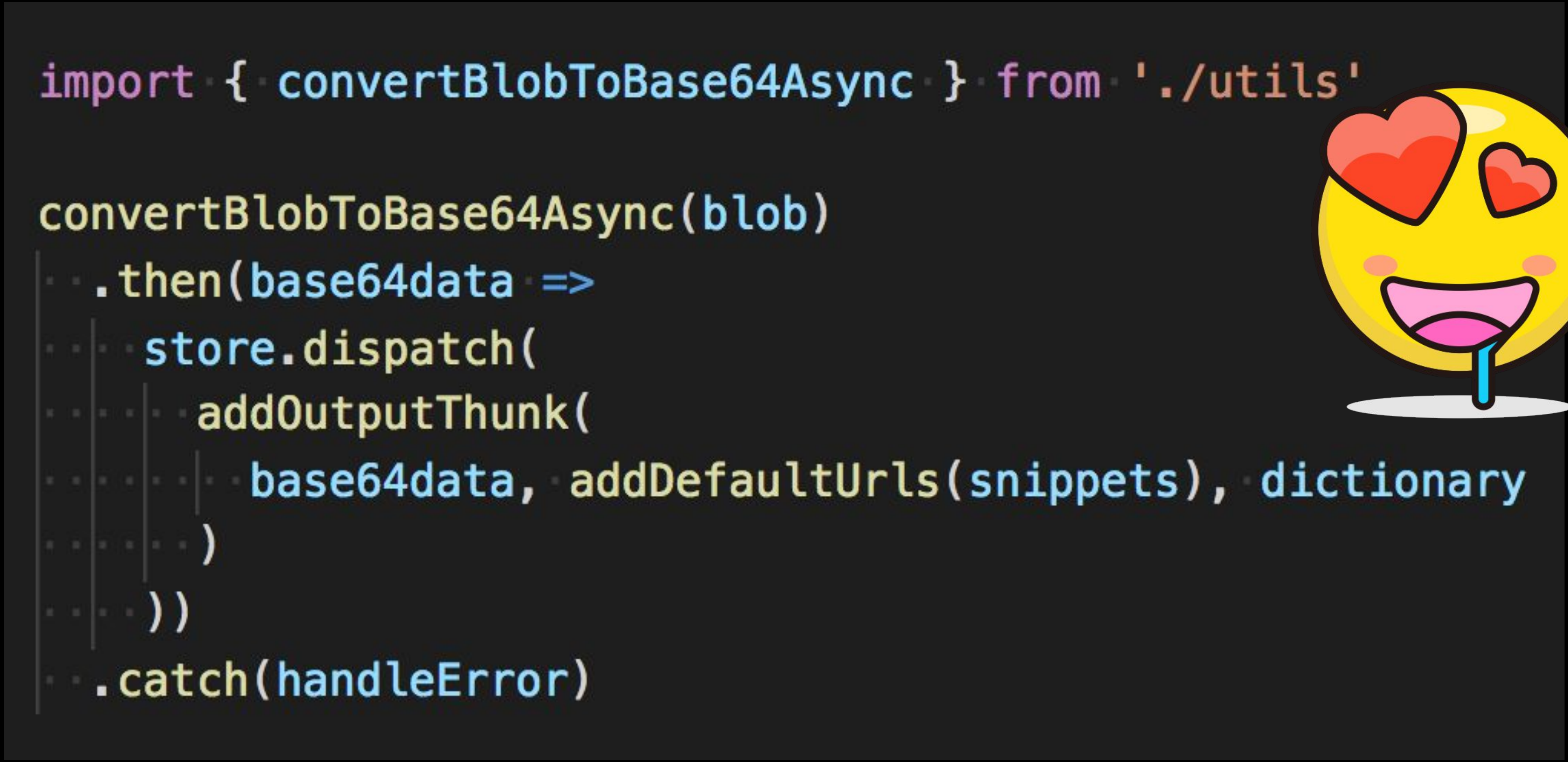


```
blobify(blob, snippets, user) {
  const reader = new FileReader();
  reader.readAsDataURL(blob);
  reader.onloadend = () => {
    let base64data = reader.result.split(',')[1];
    let userUrls = [
      { command: 'github', code: user.githubURL },
      { command: 'waffle', code: user.waffleURL },
      { command: 'stackoverflow', code: user.stackoverflowURL }
    ];
    snippets = snippets.concat(userUrls);
    store.dispatch(addOutputThunk(base64data, snippets, dictionary));
  };
}
```

After



```javascript
/**
 * Convert blob of audio data to base64 string
 * @returns {Promise<String>}
*/
export const convertBlobToBase64Async = (blob) => {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.readAsDataURL(blob);
    reader.onloadend = () => {
      if (reader.error) {
        reject(reader.error)
      }
      try {
        resolve(reader.result.split(',')[1]);
      } catch (e) {
        reject(e)
      }
    };
  })
}
```



```
import { convertBlobToBase64Async } from './utils'

convertBlobToBase64Async(blob)
  .then(base64data =>
    store.dispatch(
      addOutputThunk(
        base64data, addDefaultUrls(snippets), dictionary
      )
    ))
  .catch(handleError)
```

Time Crunch Sacrifices

- Documentation
- Interfaces
- Tests
- Browsability
- Performance
- Security: sanitize inputs server-side
- Error-handling: React error boundary and reporting services
- A “product” perspective: analytics, user testing

http://vocode.herokuapp.com