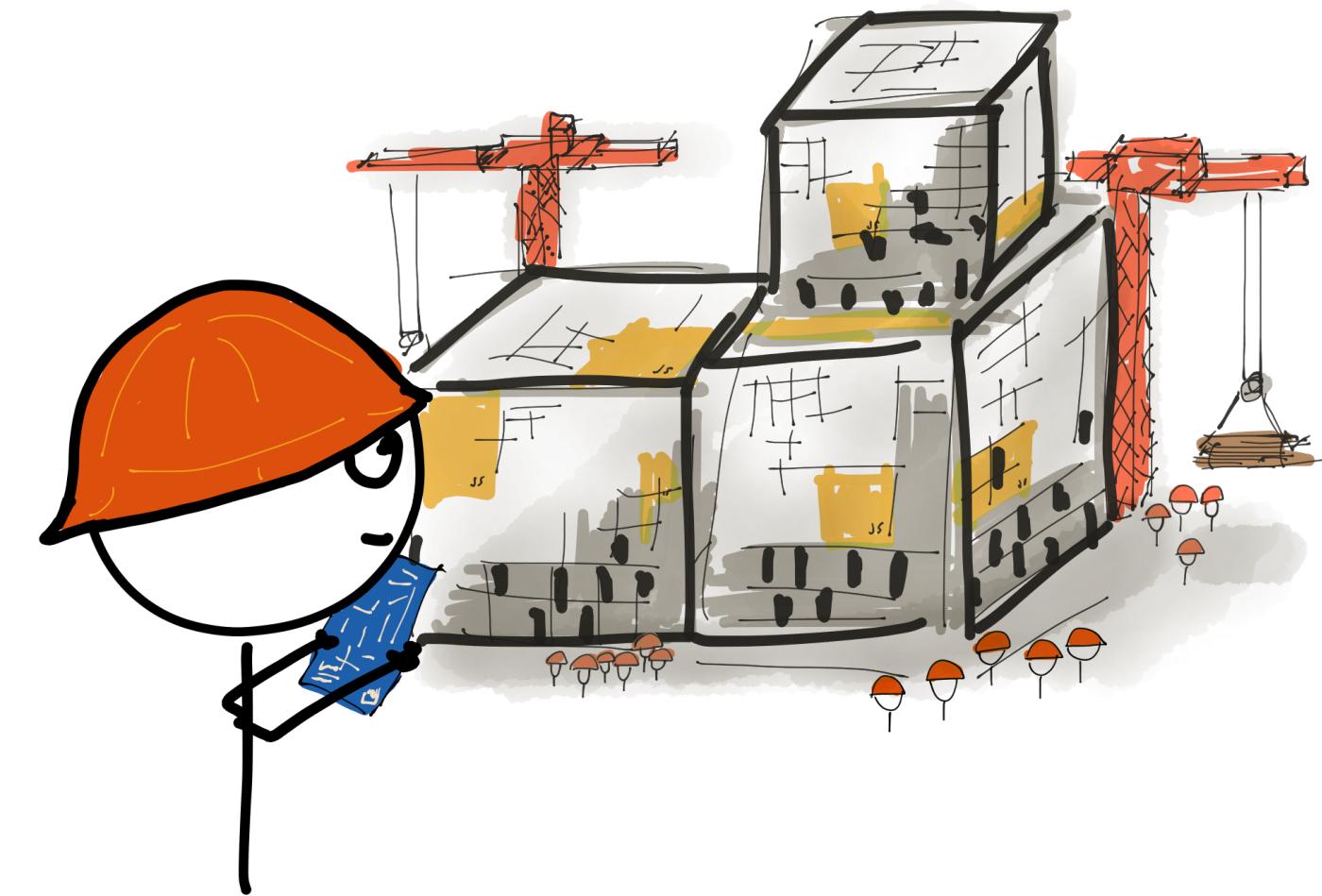
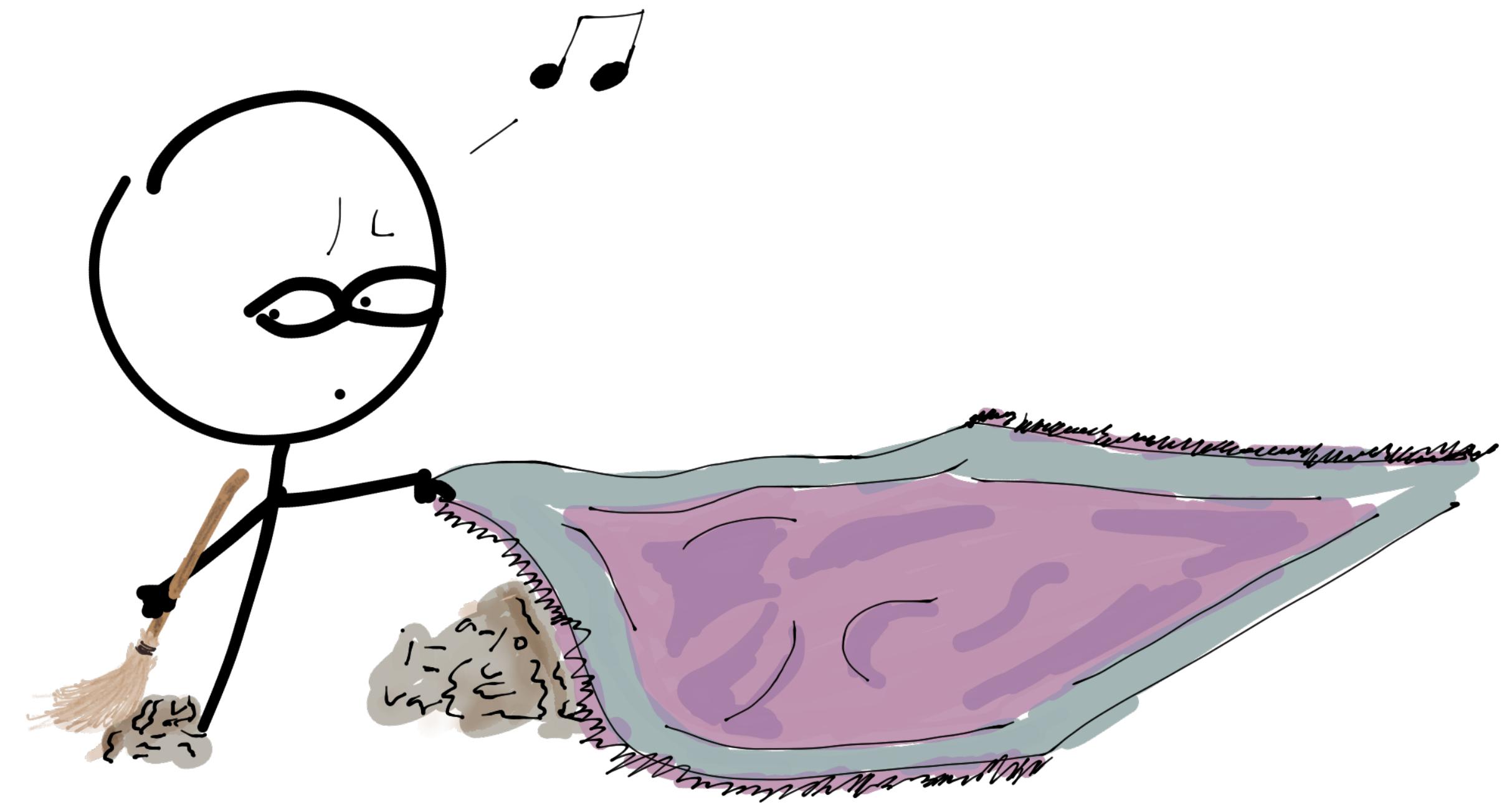
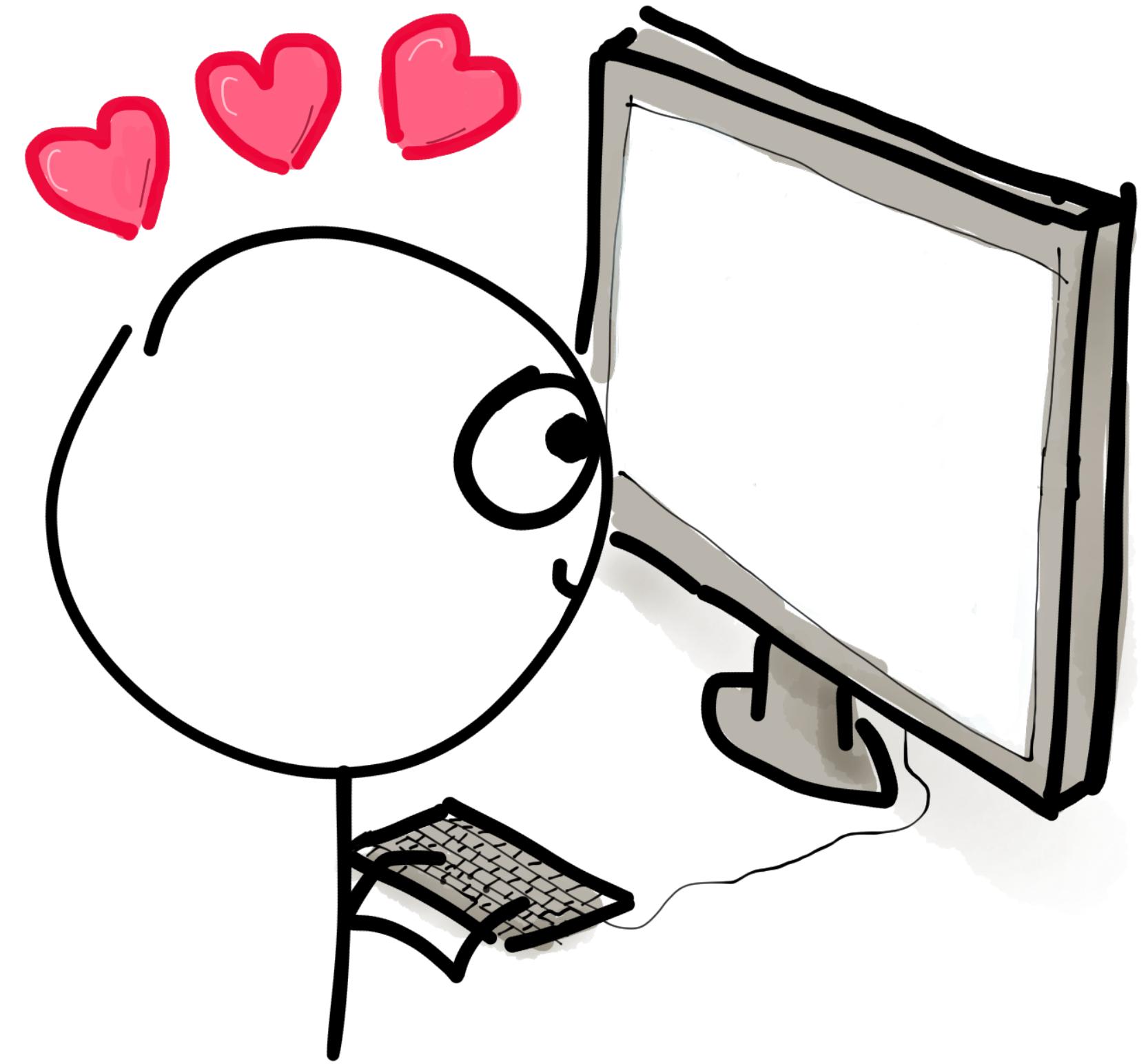
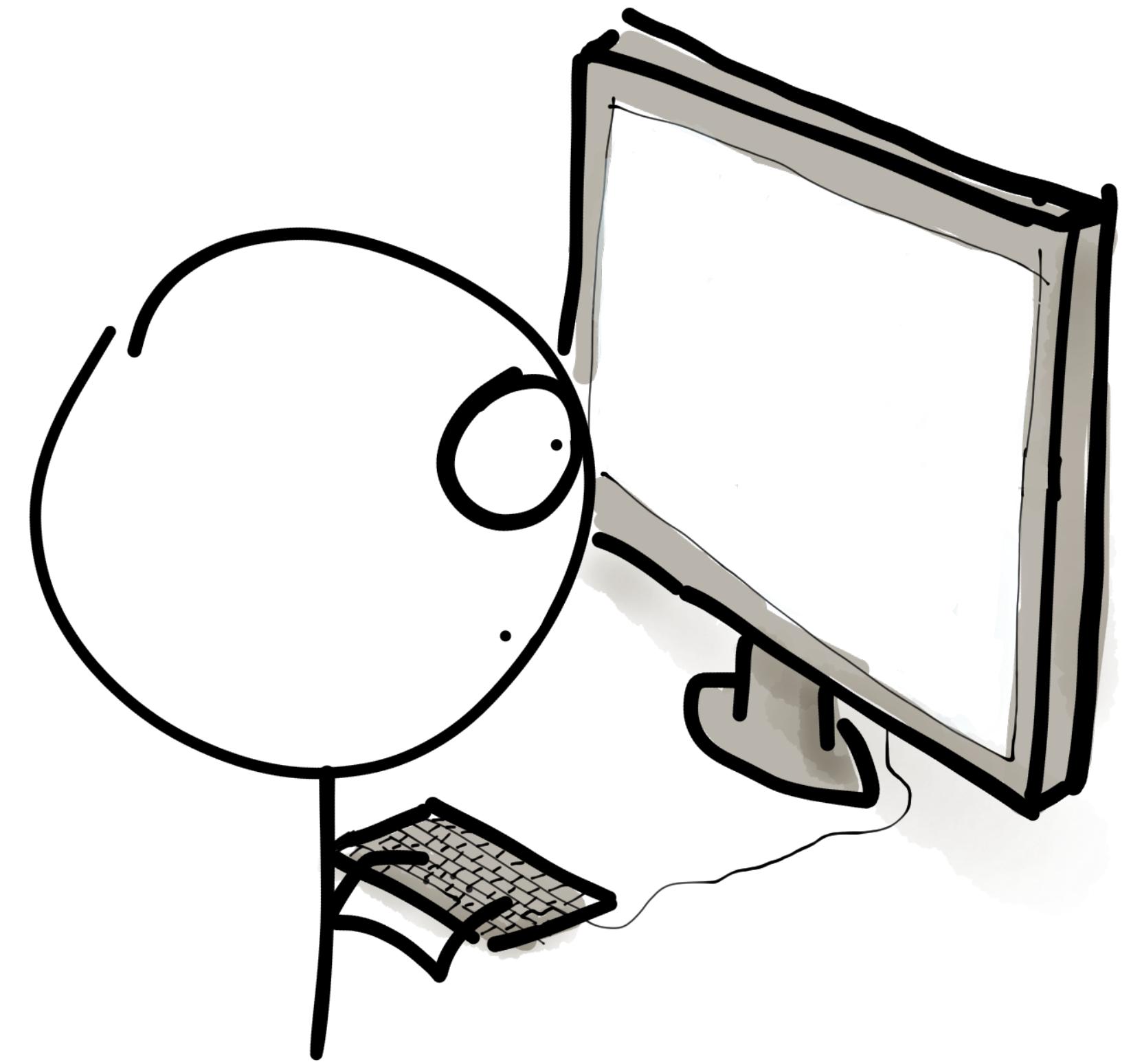


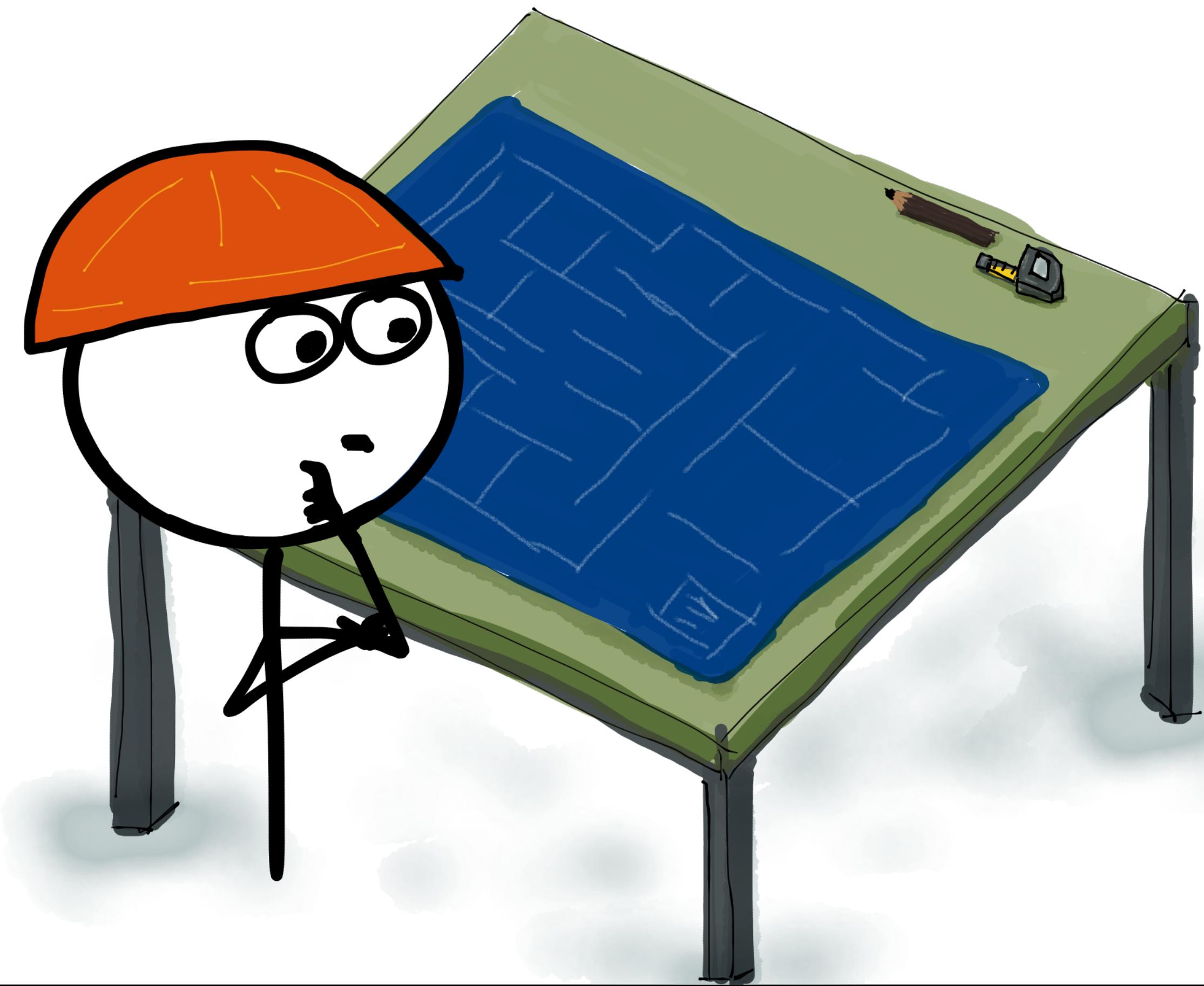
Growing Express Architecture



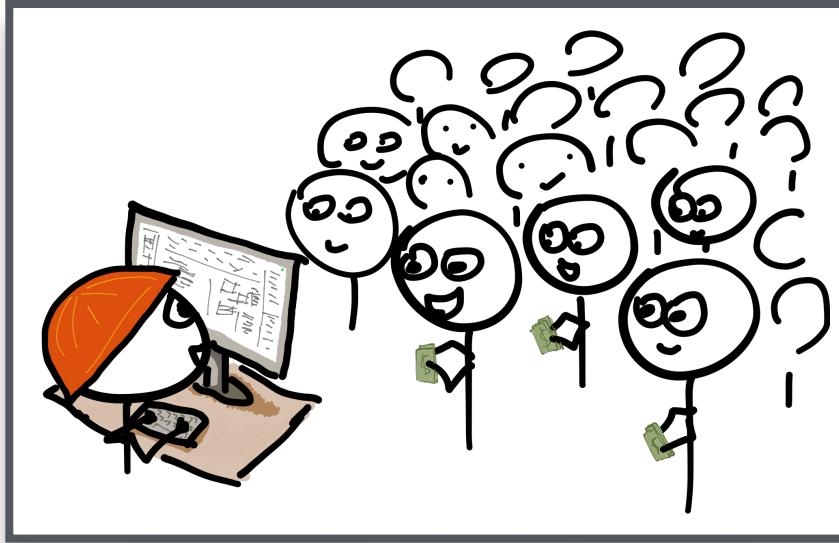




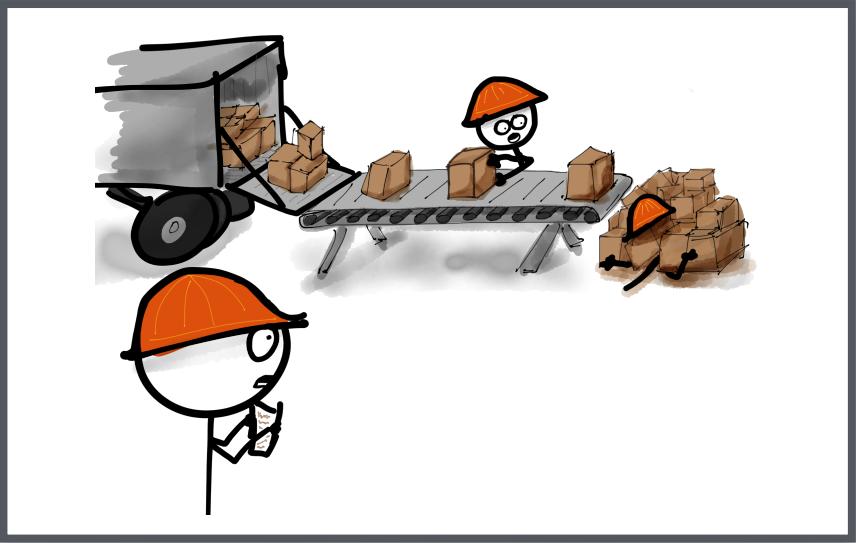




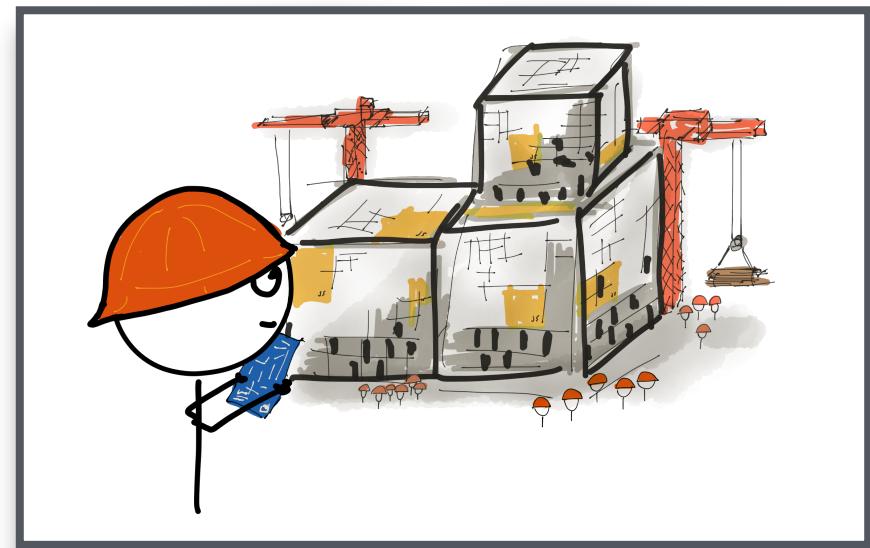
1. Users



2. Features



3. Teams





" Press ? for help

.. (up a dir)
</episode-10/
► env/
► initializers/
▼ lib/
 ▼ media/
 file.js
 index.js
 ► media/
 ► migrootions/
 ► node_modules/
 ► scripts/
 gruntfile.js
 migroose.js
 nodemon.json
 package.json

~
~
~
~
~
~
~
~
~
~

file.js (~/dev/screencasts/architecting-express-apps/episode-10/lib/media) - VIM

```
19      this.findOne(query, cb);
20  });
21 }
22 // File Methods
23 // -----
24
25
26 FileSchema.method("getDownloadUrl", function(cb){
27     var options = {
28         Bucket: epa.get("s3").bucket,
29         Key: this.name
30     };
31
32     var s3 = new AWS.S3();
33
34     s3.getSignedUrl("getObject", options, cb);
35 }
36
37 FileSchema.method("getInfo", function(cb){
38     var options = {
39         Bucket: epa.get("s3").bucket,
40         Key: this.name
41     };
42 }
```

<ecting-express-apps/episode-10 [1] lib/media/file.js [javascript] [Git(master)] 28,1-1 56%

-- VISUAL LINE -- 1

" Press ? for help

.. (up a dir)
</episode-5/
► env/
▼ initializers/
 aws.js
 mongoose.js
► lib/
▼ media/
 ▼ bin/
 www*
 ► public/
 ► routes/
 ► views/
 app.js
► node_modules/
► scripts/
 gruntfile.js
 nodemon.json
 package.json

~
~
~

www (~/dev/screencasts/architecting-express-apps/episode-5/media/bin) - VIM

```
1 #!/usr/bin/env node
2
3 var app = require('../app');
4 var debug = require('debug')('media:server');
5 var http = require('http');
6 var nanit = require("nanit");
7
8 var port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 nanit.initialize(function(err){
12     if (err) { throw err; }
13
14     var server = http.createServer(app);
15
16     server.listen(port);
17     server.on('error', onError);
18     server.on('listening', () => onListening(server));
19 });
20
21 function normalizePort(val) {
22     var port = parseInt(val, 10);
23
24     if (isNaN(port)) {
```

<tecting-express-apps/episode-5 [3] media/bin/www [javascript] [Git(master)] 20,0-1 Top
-- VISUAL LINE -- 10

" Press ? for help

.. (up a dir)
</66-cleanup-route-methods/
► env/
► lib/
▼ web/
 ► bin/
 ► public/
 ▼ routes/
 home.js
 index.js
 post-author.js
 posts.js
 users.js
 ► views/
 app.js
 npm-debug.log

~
~
~
~
~
~

posts.js (~/dev/screencasts/express/66-cleanup-route-methods/web/routes) - VIM

```
1 var express = require("express");
2 var author = require("./post-author");
3 var Posts = require("posts");
4
5 var router = new express.Router();
6
7 router.get("/", list);
8 router.get("/:id", captureView, getPostsByAuthor, view);
9 router.get("/:id/edit", edit);
10
11 router.use("/:id/author", author);
12
13 router.param(":id", function(req, res, next, id){
14   Posts.loadById(id, function(err, post){
15     if (err) { return next(err); }
16
17     req.appData = {
18       post: post
19     };
20
21     next();
22   });
23 });
24
```

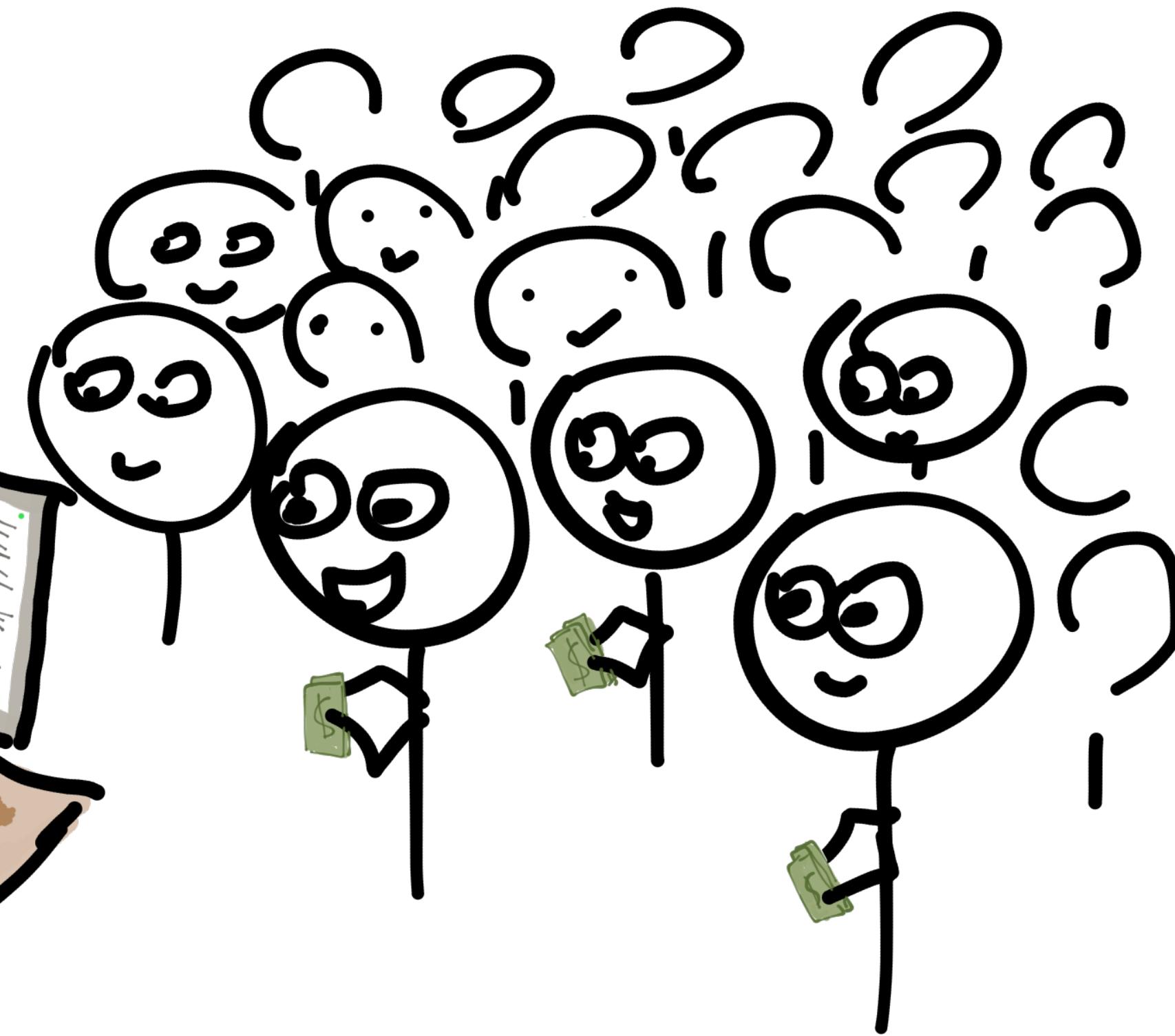
<press/66-cleanup-route-methods [4] web/routes/posts.js [javascript] [Git(master)] 1,1-1 Top
"web/routes/posts.js" 81L, 1499C

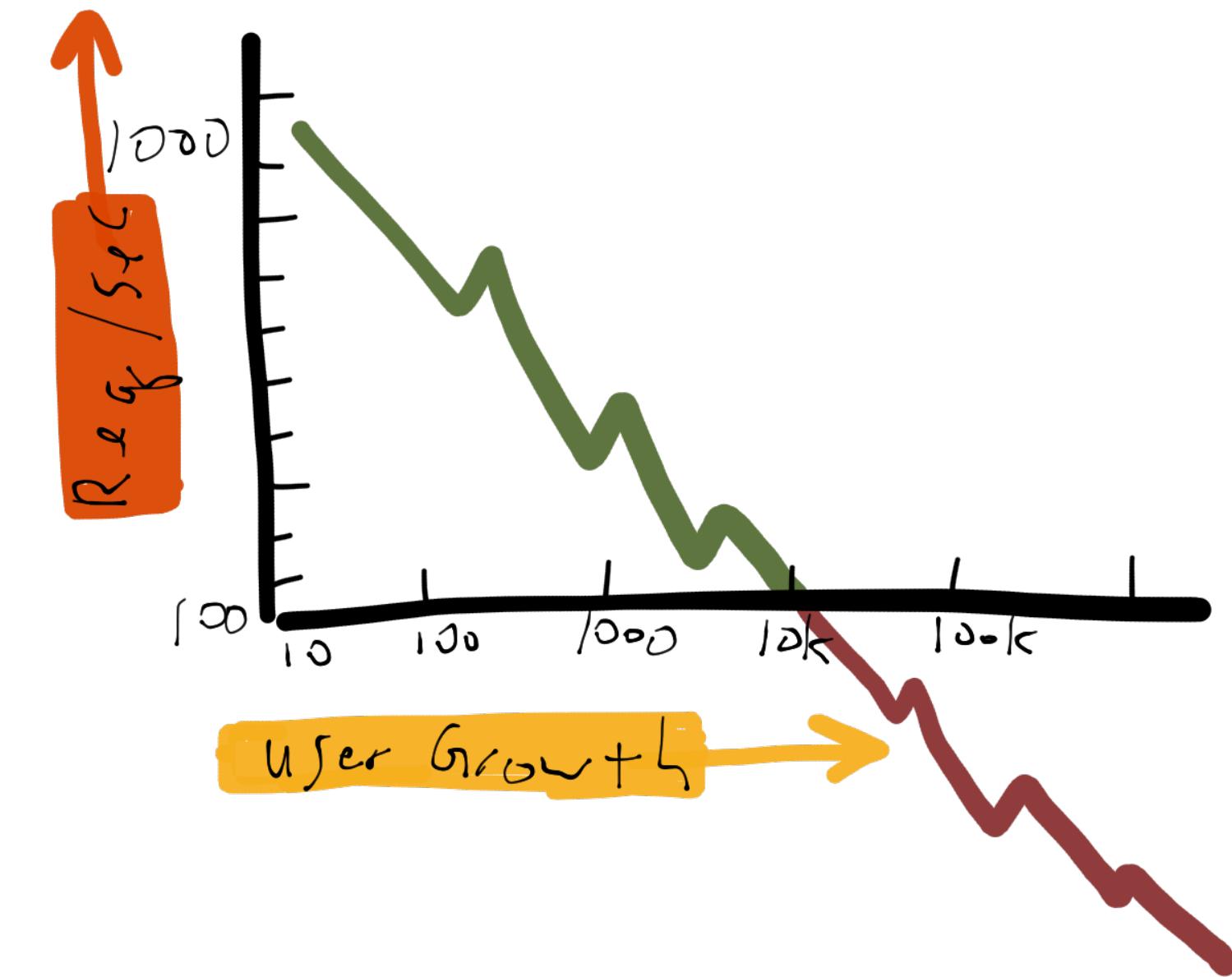
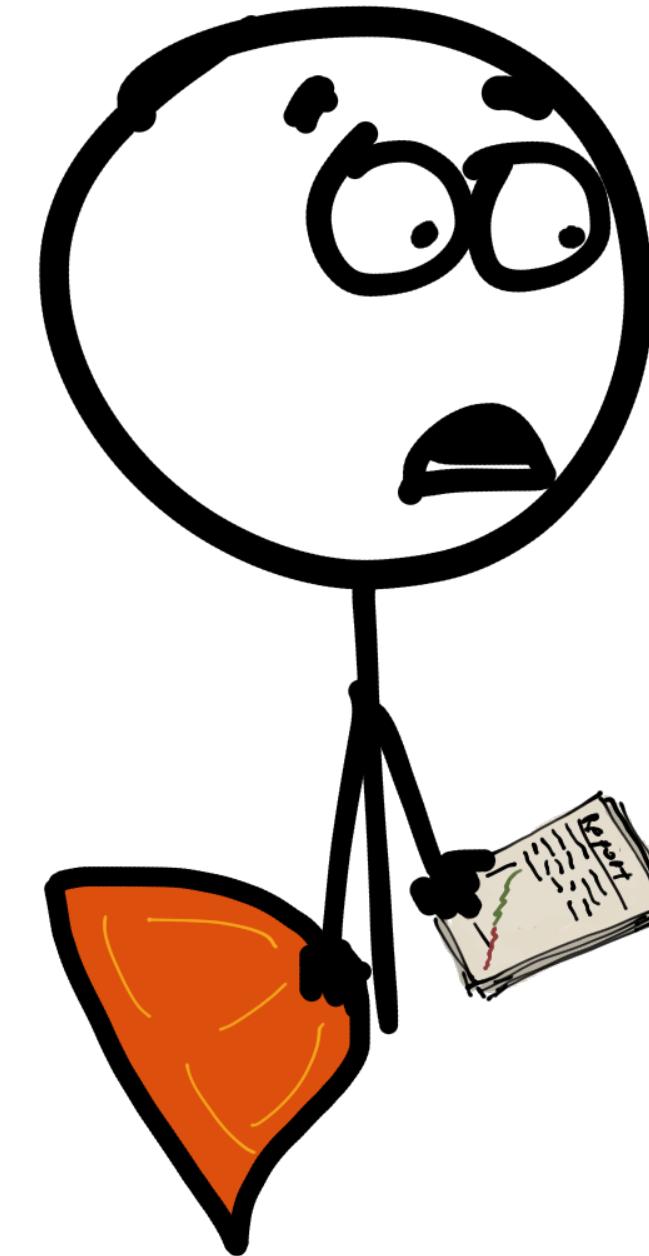
```
router.get("/:id", captureView, getPostsByAuthor, view);
```

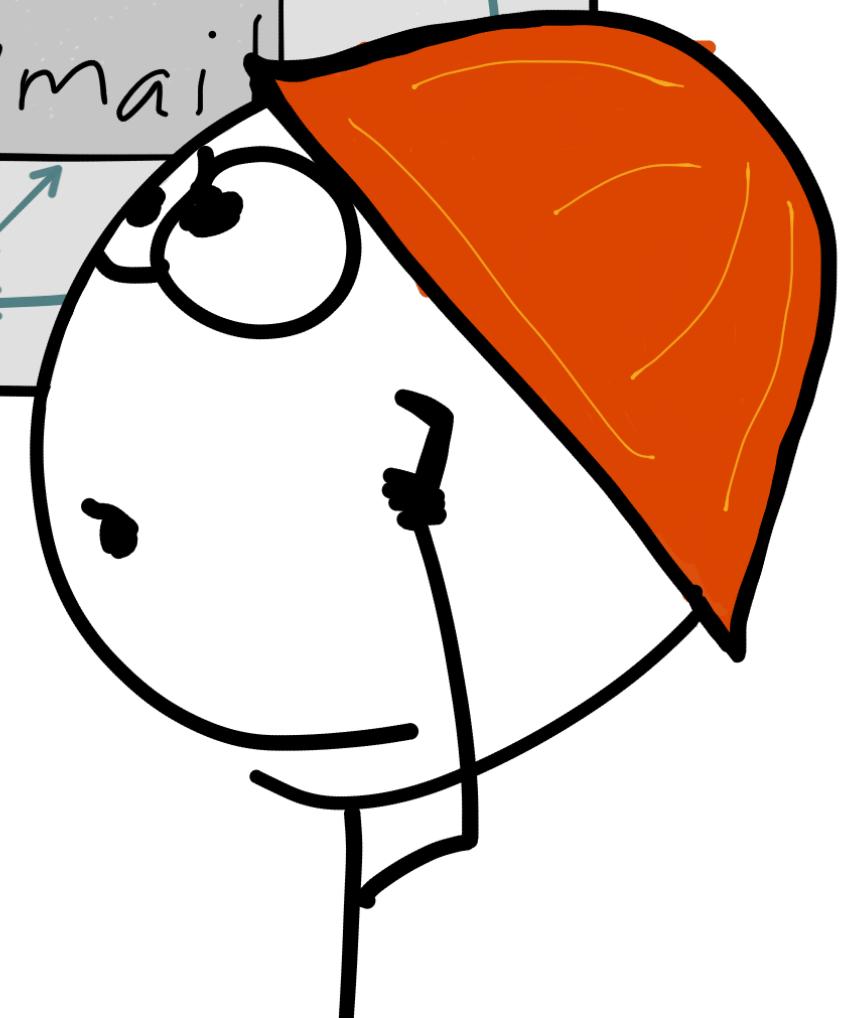
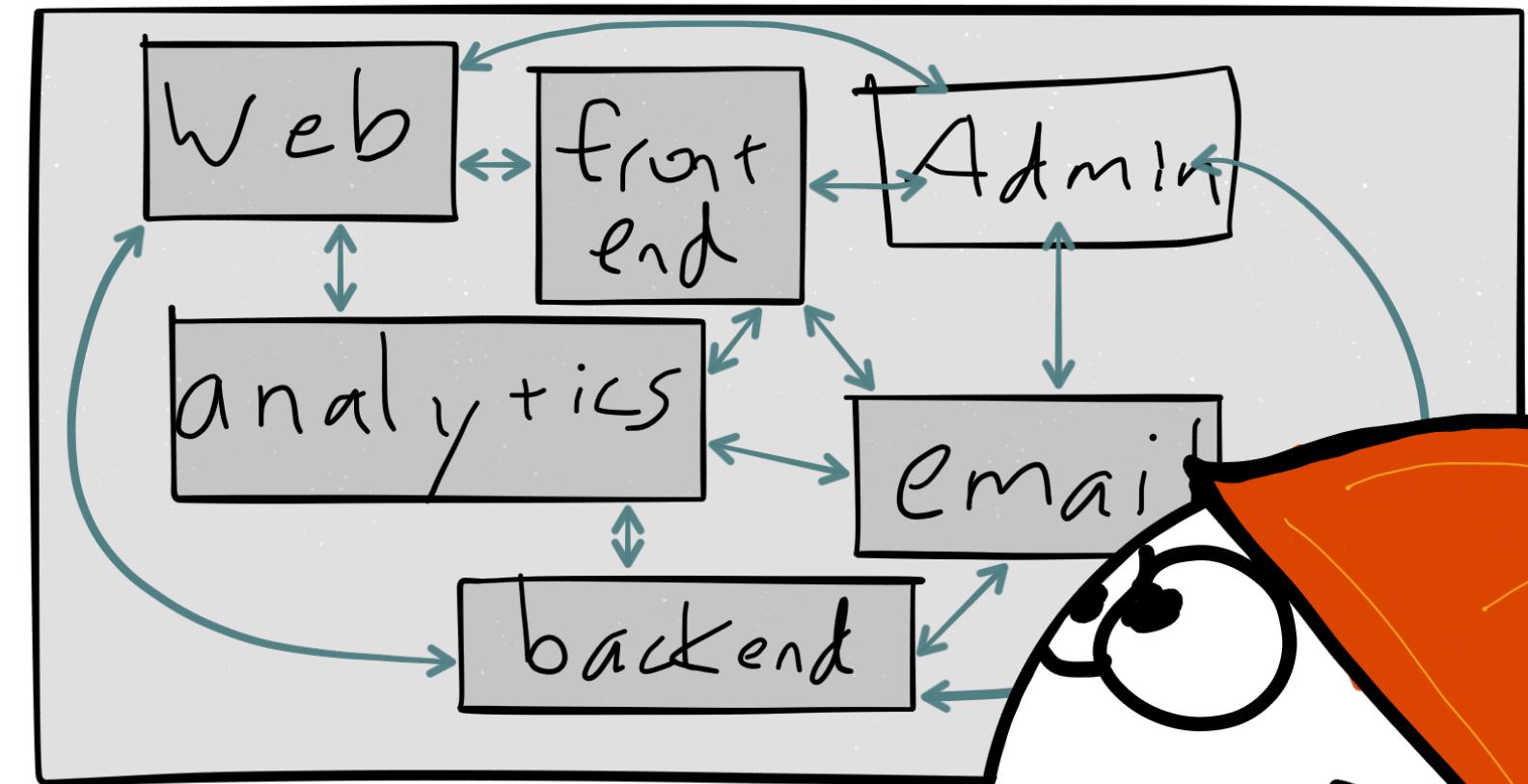
```
function captureView(req, res, next){  
  var post = req.appData.post;  
  
  if (!req.session.viewedPosts){  
    req.session.viewedPosts = [];  
  }  
  req.session.viewedPosts.push({  
    id: post.id,  
    title: post.title  
  });  
  
  next();  
}
```

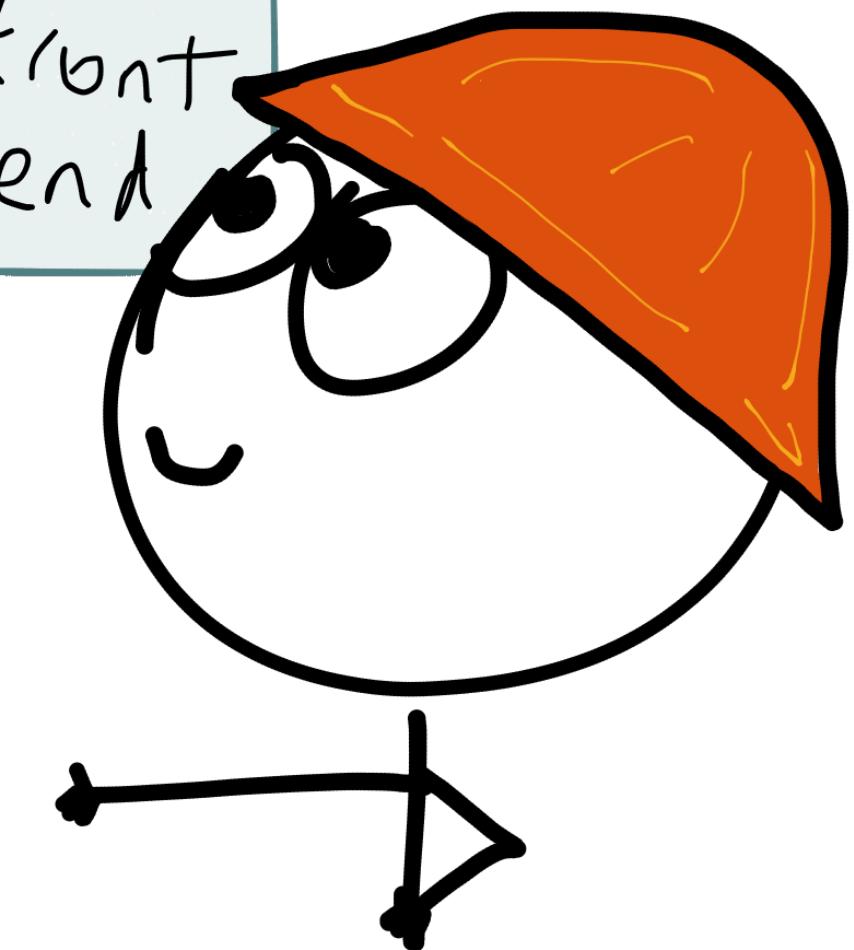
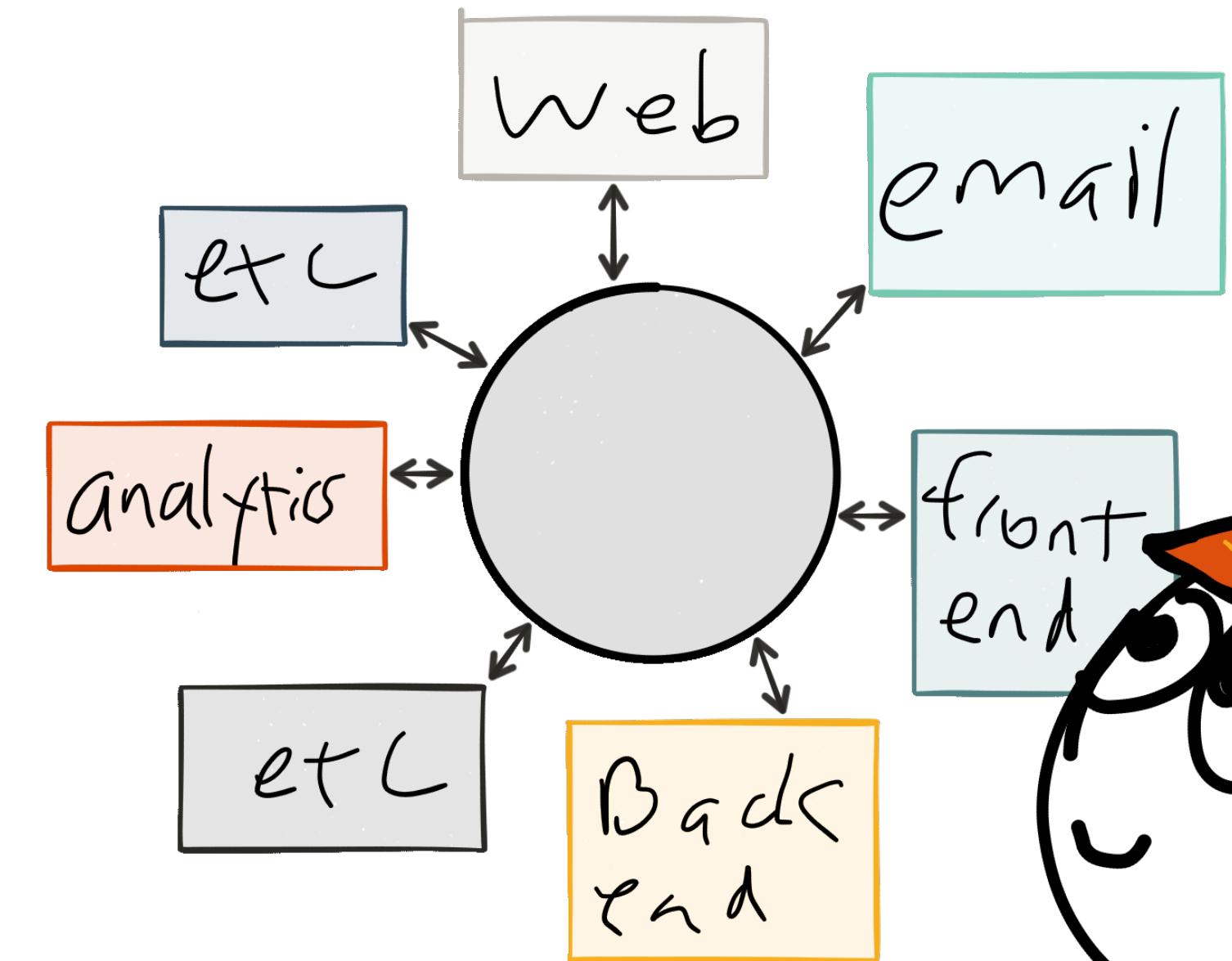
```
function getPostsByAuthor(req, res, next){  
  var post = req.appData.post;  
  var authorId = post.author.id;  
  
  Posts.findById(authorId, function(err, posts){  
    if (err) { return next(err); }  
  
    req.appData.postsByAuthor = posts;  
    next();  
  });  
}
```

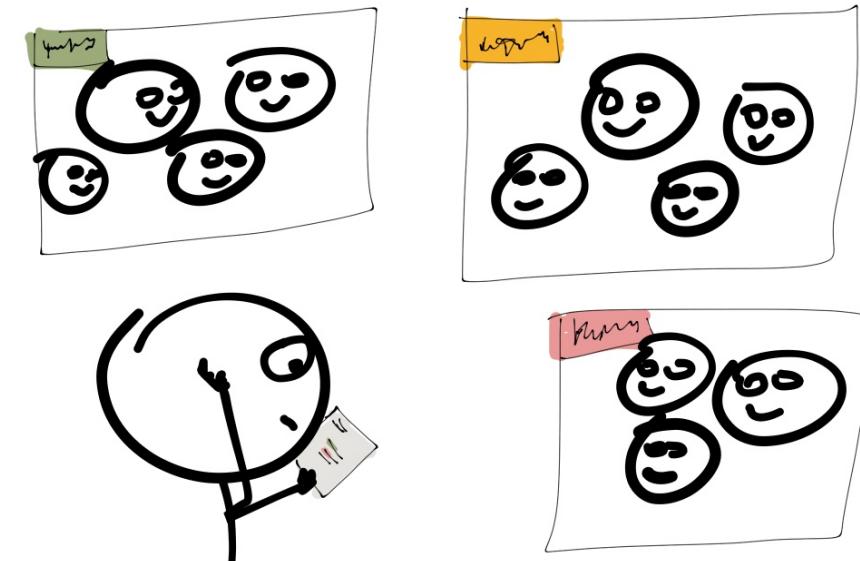
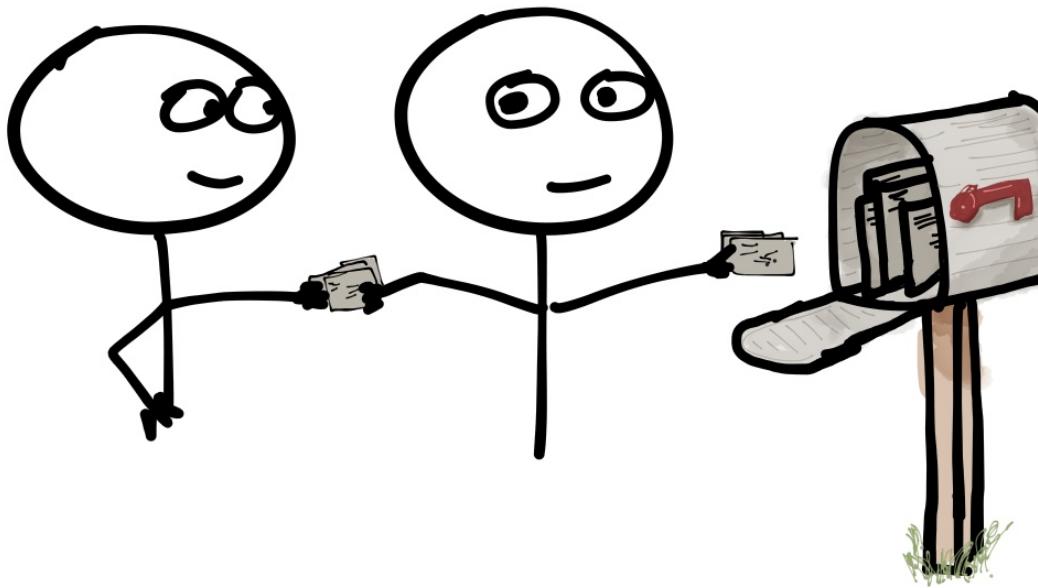
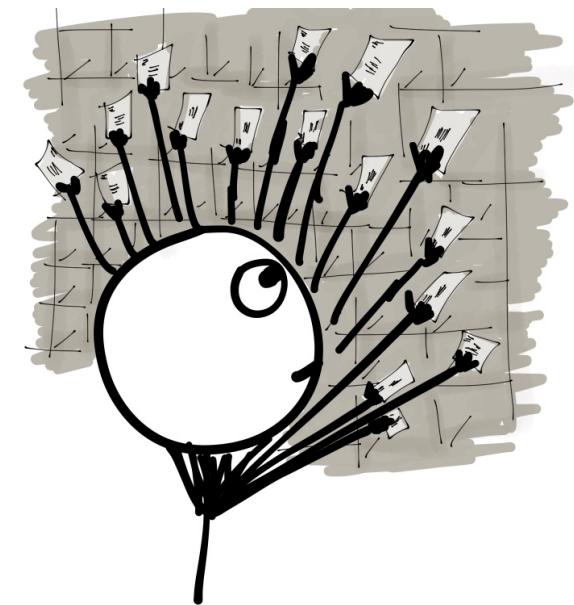
```
function view(req, res, next){  
  var post = req.appData.post;  
  var postsByAuthor = req.appData.postsByAuthor;  
  
  res.render("post", {  
    post: post,  
    postsByAuthor: postsByAuthor  
  });  
}
```





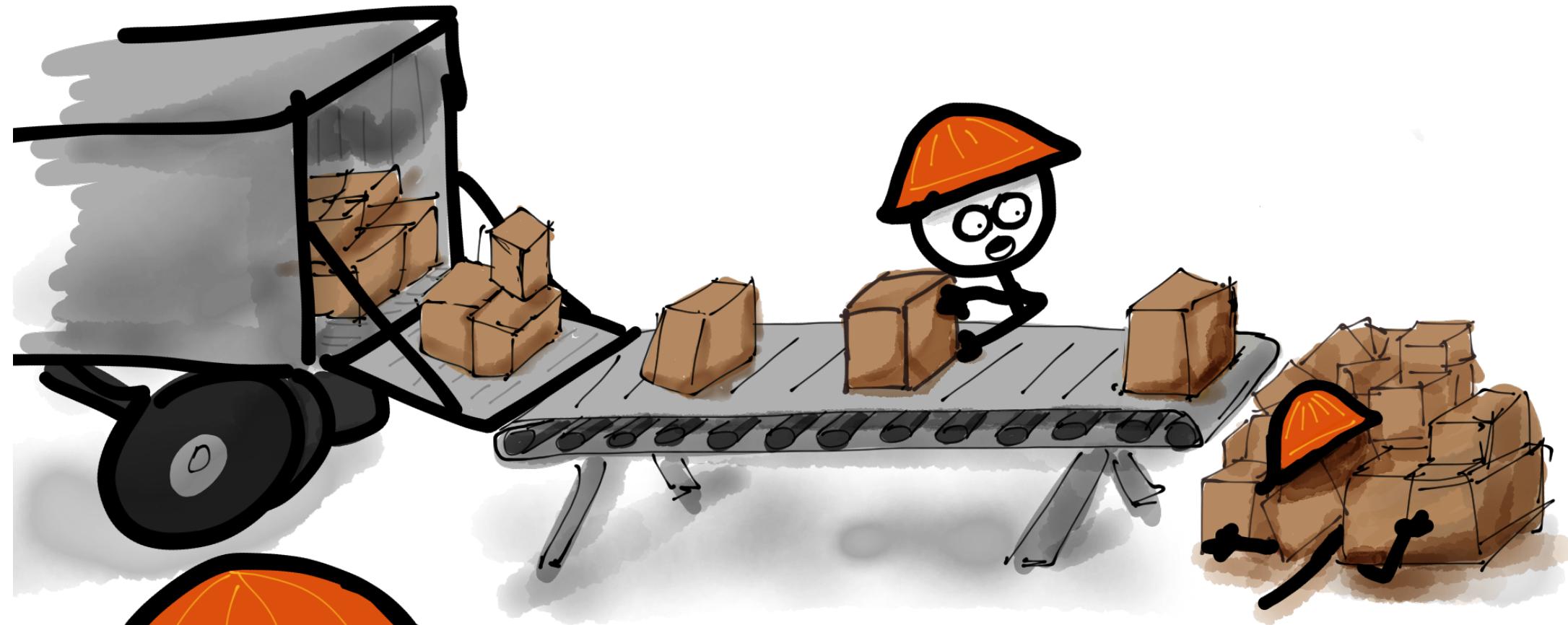












" Press ? for help

.. (up a dir)
</66-cleanup-route-methods/

► env/

▼ lib/

 ▼ posts/

 comment.js

 index.js

 ▼ status/

 index.js

 ▼ users/

 index.js

▼ web/

 ► bin/

 ► public/

 ▼ routes/

 home.js

 index.js

 post-author.js

 posts.js

 users.js

 ► views/

 app.js

 npm-debug.log

<press/66-cleanup-route-methods [4] web/routes/posts.js [javascript] [Git(master)] 14,1-1 Top

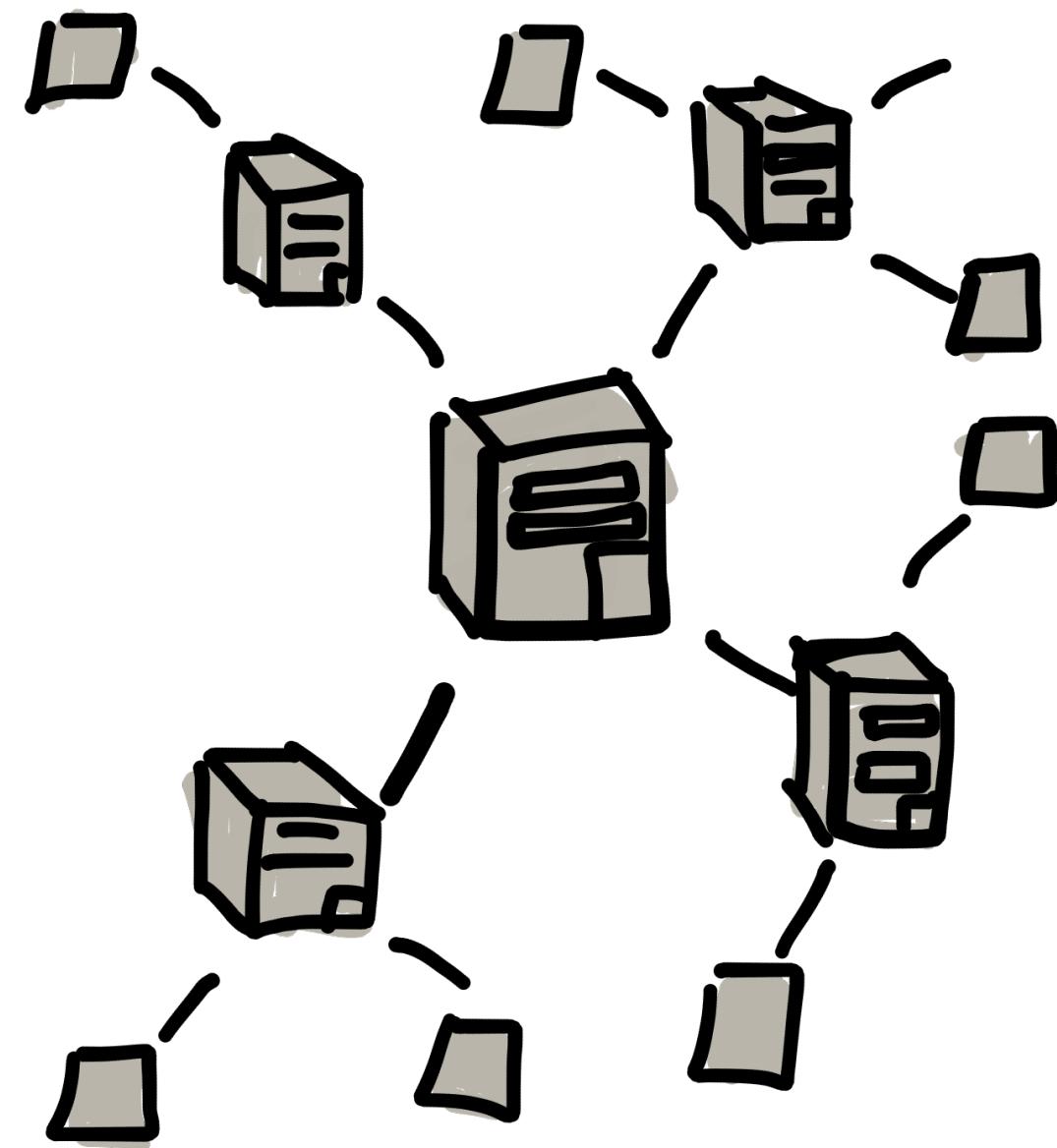
-- VISUAL LINE --

```
1 var express = require("express");
2 var author = require("./post-author");
3 var Posts = require("posts");
4
5 var router = new express.Router();
6
7 router.get("/", list);
8 router.get("/:id", captureView, getPostsByAuthor, view);
9 router.get("/:id/edit", edit);
10
11 router.use("/:id/author", author);
12
13 router.param(":id", function(req, res, next, id){
14   Posts.loadById(id, function(err, post){
15     if (err) { return next(err); }
16
17     req.appData = {
18       post: post
19     };
20
21     next();
22   });
23 });
24
25 function list(req, res, next){
```

```
▼ web/
  ▶ bin/
  ▶ lib/
  ▶ public/
  ▶ routes/
  ▼ views/
    ▶ login/
    ▶ notifications/
    ▼ schedule-templates/
      ▶ agent-types/
      ▶ dependency-types/
      ▶ job-steps/
      ▶ jobs/
      ▶ non-concurrence/
      ▶ schedules/
      ▶ step-families/
    ▶ schedules/
    ▶ templates/
      _errorList.jade
      _navbar.jade
      error.jade
      index.jade
      layout.jade
    app.js
```



```
agents/
assets/
docs/
env/
initializers/
lib/
  account/
  agentsmith/
  db/
  enum/
  errors/
  es6-poly/
  loadScheduleSender/
  logging/
  notifications/
  onCollectionEvent/
  qdot/
  registry/
  require-folder/
  resolver/
  scheduling/
  seafoam/
  viewModel/
logs/
mastercontrol/
migrootions/
node_modules/
queries/
scheduleLoadTimer/
schedulePersist/
specs/
tron/
web/
  deploy_production*
  gen-id*
  gruntfile.js
  migroose.js
  package.json
```



```
1 var express = require("express");
2 var messageBus = require("bus");
3
4 // routes
5 // -----
6
7 var router = new express.Router();
8 router.post("/", stripeHook); // post to '/api/stripe'
9
10 // route handlers
11 // -----
12
13 function stripeHook(req, res, next){
14     var event = req.body;
15
16     var hookData = {
17         eventId: event.id,
18         eventType: event.type,
19         dateReceived: new Date()
20     };
21
22     messageBus.sendStripeHook(hookData, function(err){
23         if (err) { return next(err); }
24
25         res.status(200);
26         res.send({});
```

```
27     });
28 };
29
```

" Press ? for help

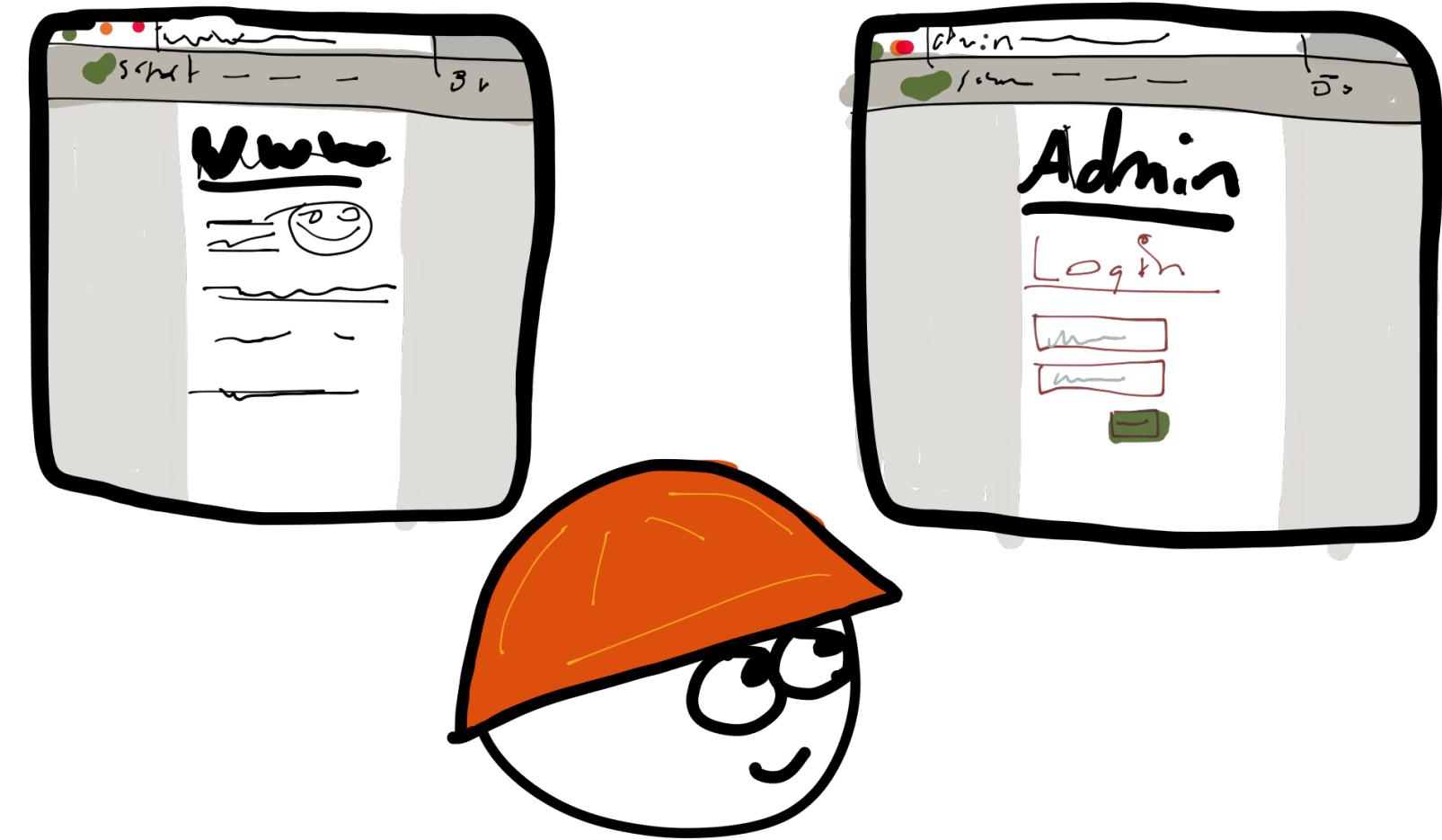
.. (up a dir)
<rabbitmq/78-errors/
► env/
▼ longProcess/
 index.js
 jobRequestReceiver.js
 jobStatusSender.js
► node_modules/
► web/
package.json

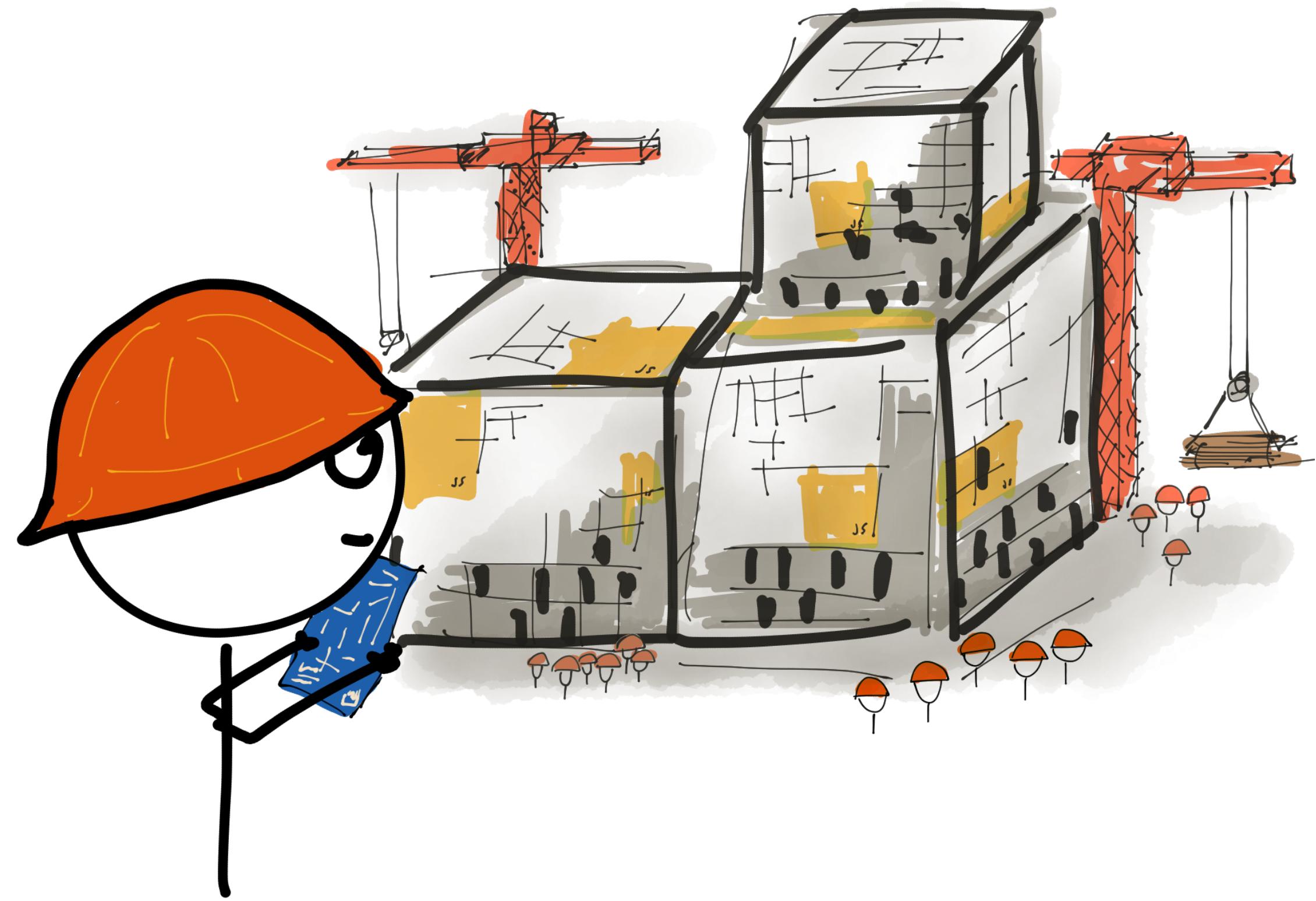
index.js

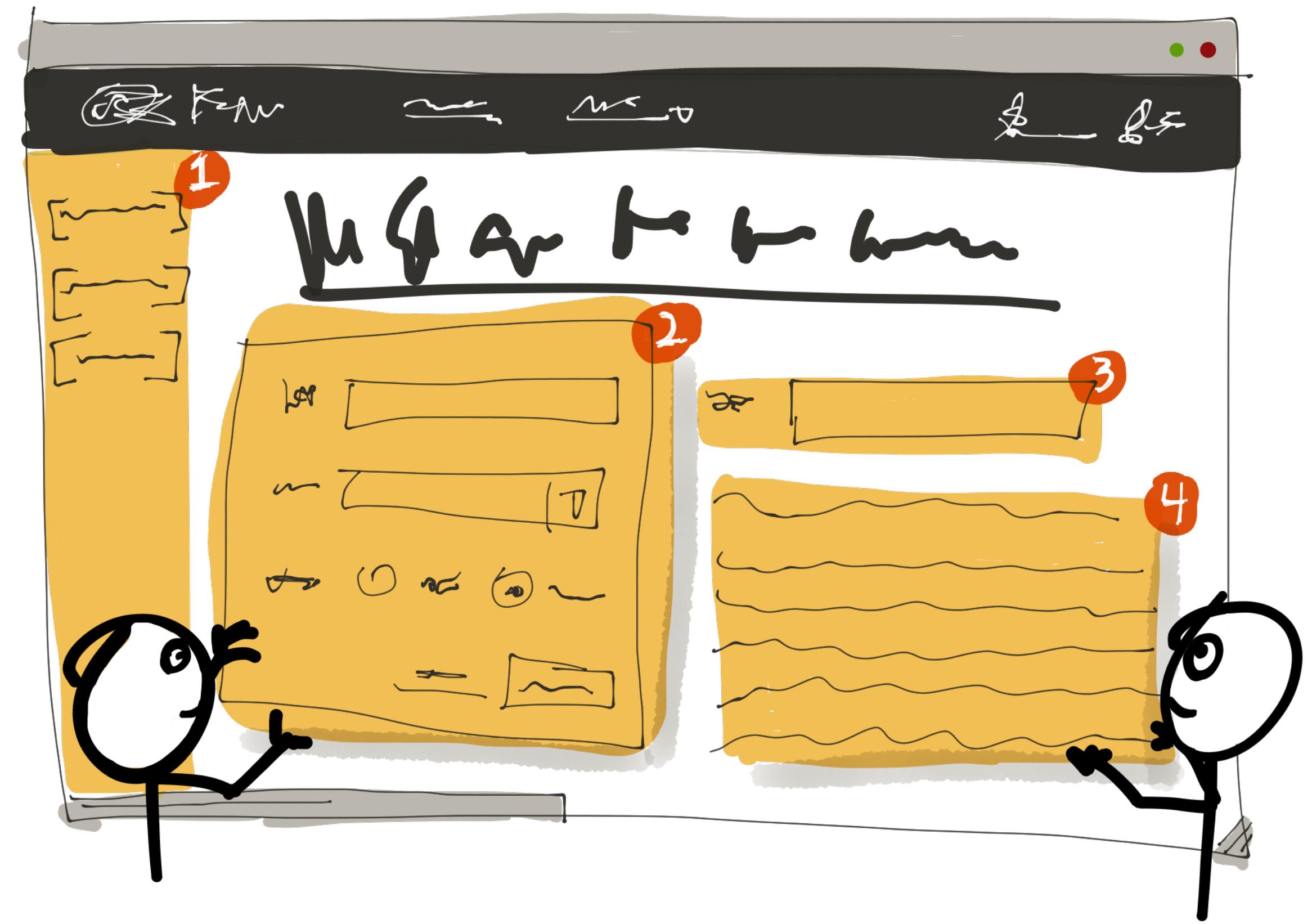
```
7 // rabbit config and start up
8 // -----
9
10 var rabbitConfig = epa.get("rabbitmq");
11
12 wascally
13     .configure(rabbitConfig)
14     .then(receiveRequests)
15     .then(undefined, reportError);
16
17 // run the job
18 // -----
19
20 function receiveRequests(){
21     var jobStatusSender = new JobStatusSender();
22     var jobRequestReceiver = new JobRequestReceiver();
23
24     jobRequestReceiver.receive(function(request, done){
25         var progress = {
26             id: request.id,
27             percent: 0
28         };
29
30         console.log("starting processing...");
31         setTimeout(function(){
32
33             console.log("33% done");
34             progress.percent = 33;
35             jobStatusSender.send(progress);
36         }, 1000);
37     });
38 }
39
40 // -----
41
42 function reportError(error) {
43     console.error(error);
44 }
```

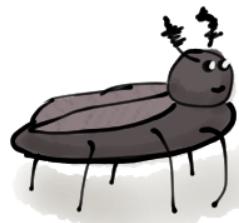
<screencasts/rabbitmq/78-errors [1] longProcess/index.js [+] [javascript] [Git(master)] 11,0-1 14%

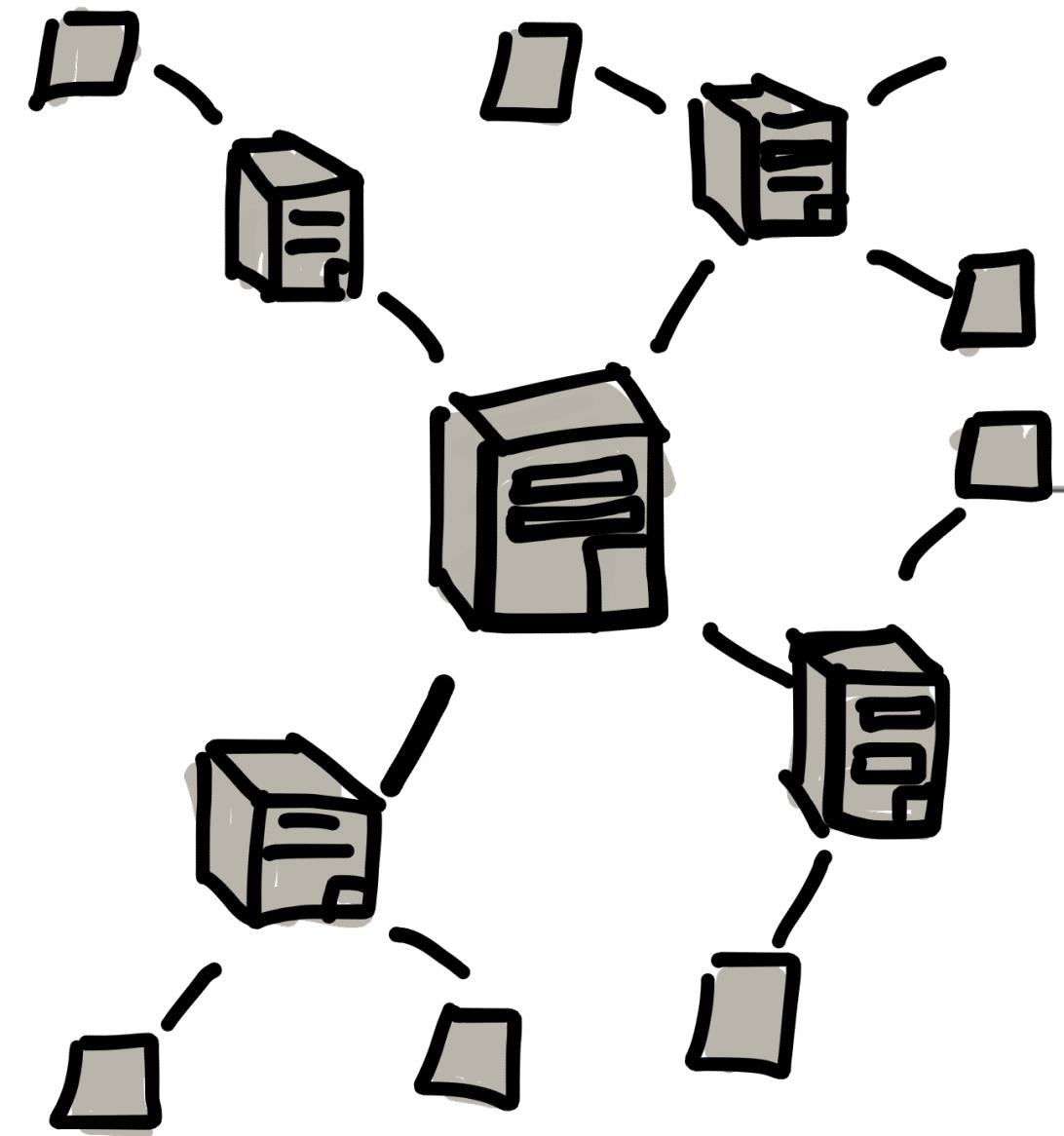
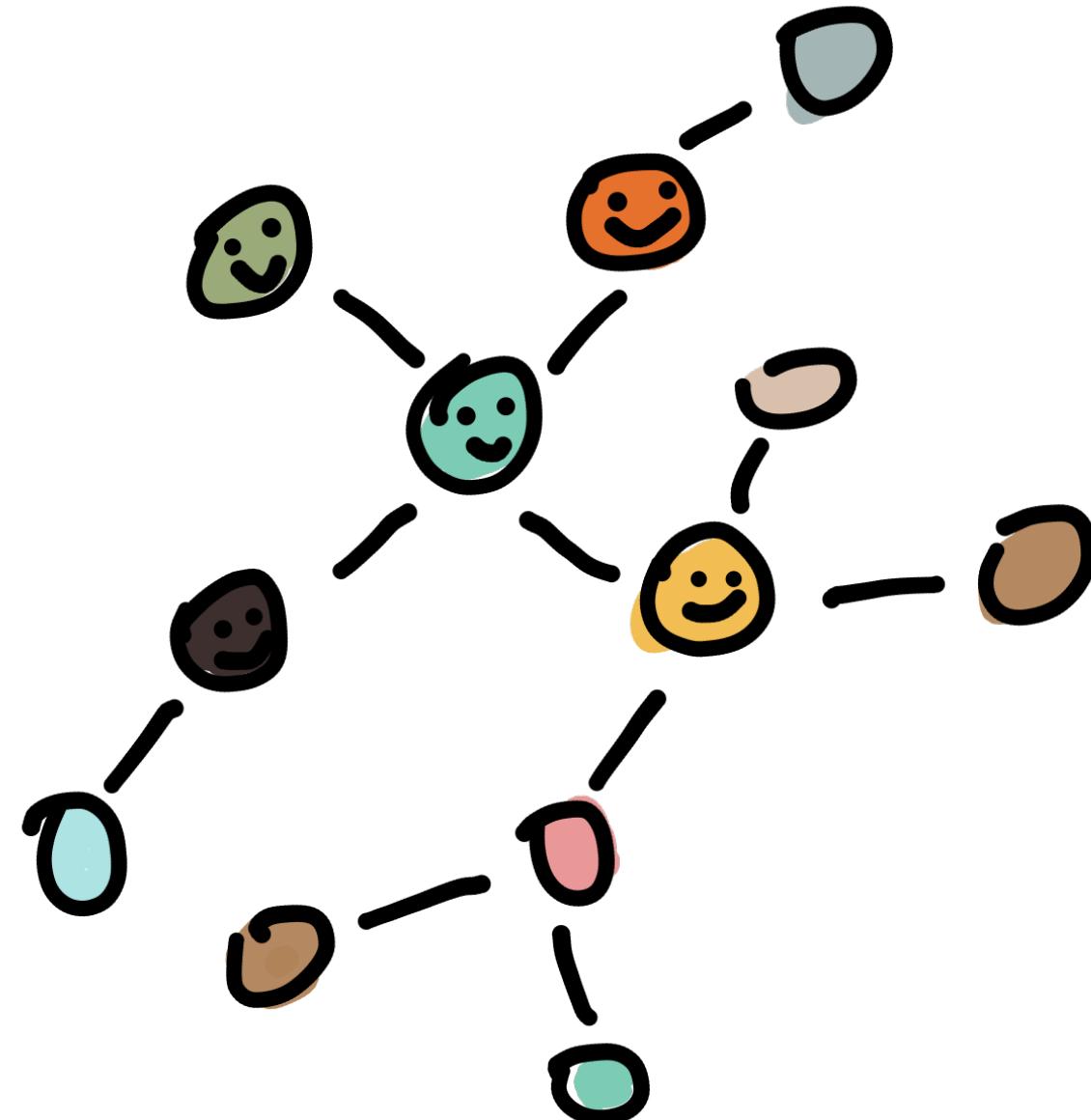
```
</dev/signalleaf/app/
└── admin/
    ├── bin/
    ├── public/
    ├── routes/
    ├── views/
    ├── app.js
    ├── everyauth-config.js
    └── mustbe-config.js
    ├── database/
    ├── deploy/
    ├── env/
    ├── import/
    ├── lib/
    ├── media/
    ├── node_modules/
    ├── qhandlers/
    ├── queries/
    ├── reports/
    ├── samples-and-demos/
    └── www/
        ├── bin/
        ├── public/
        ├── routes/
        ├── views/
        ├── app.js
        ├── iamConfig.js
        └── mustBeConfig.js
```

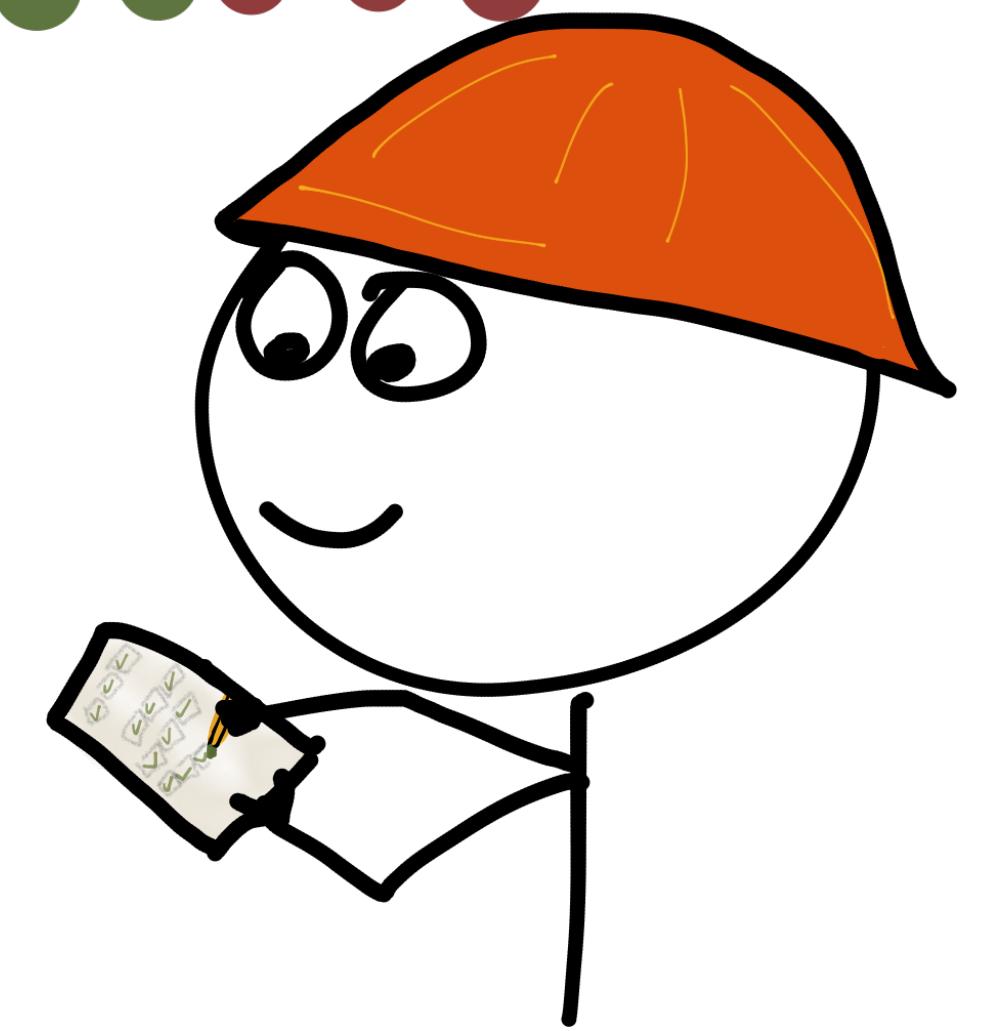
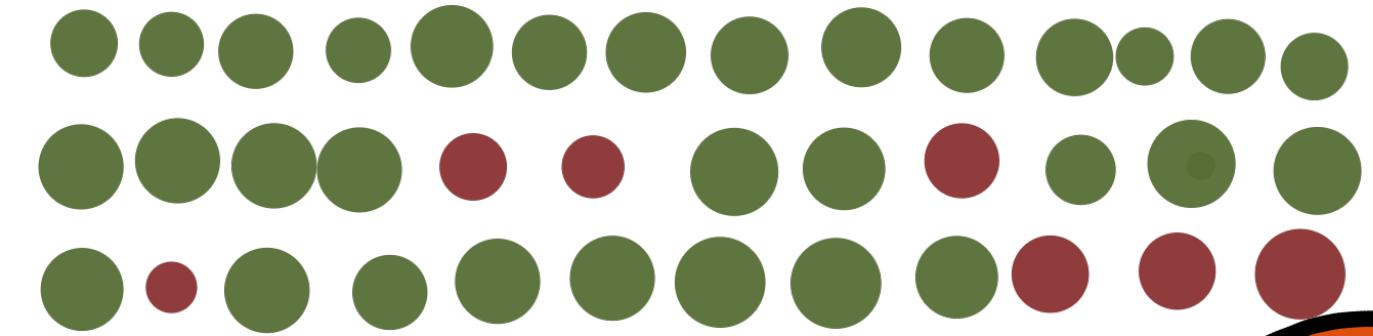


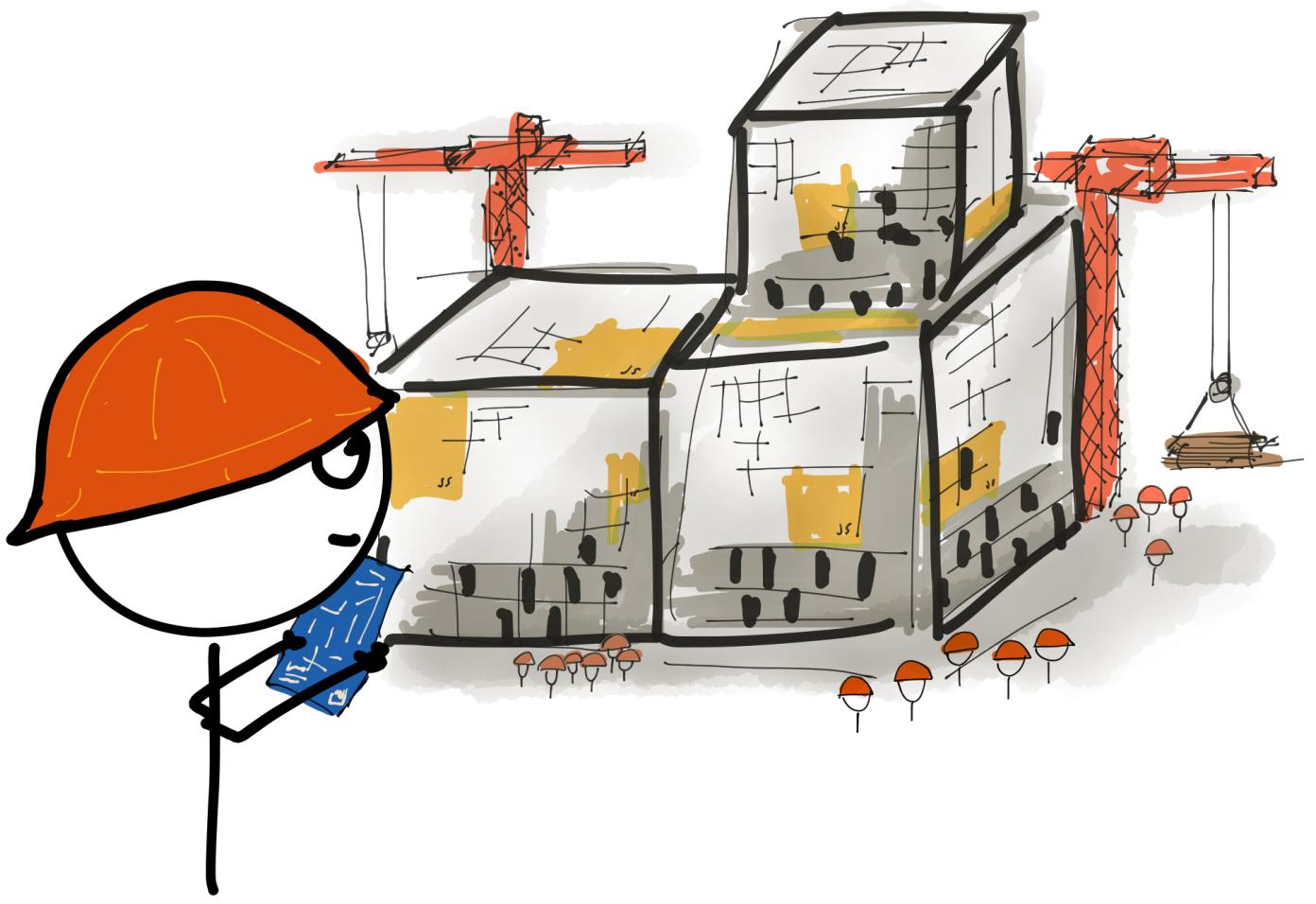






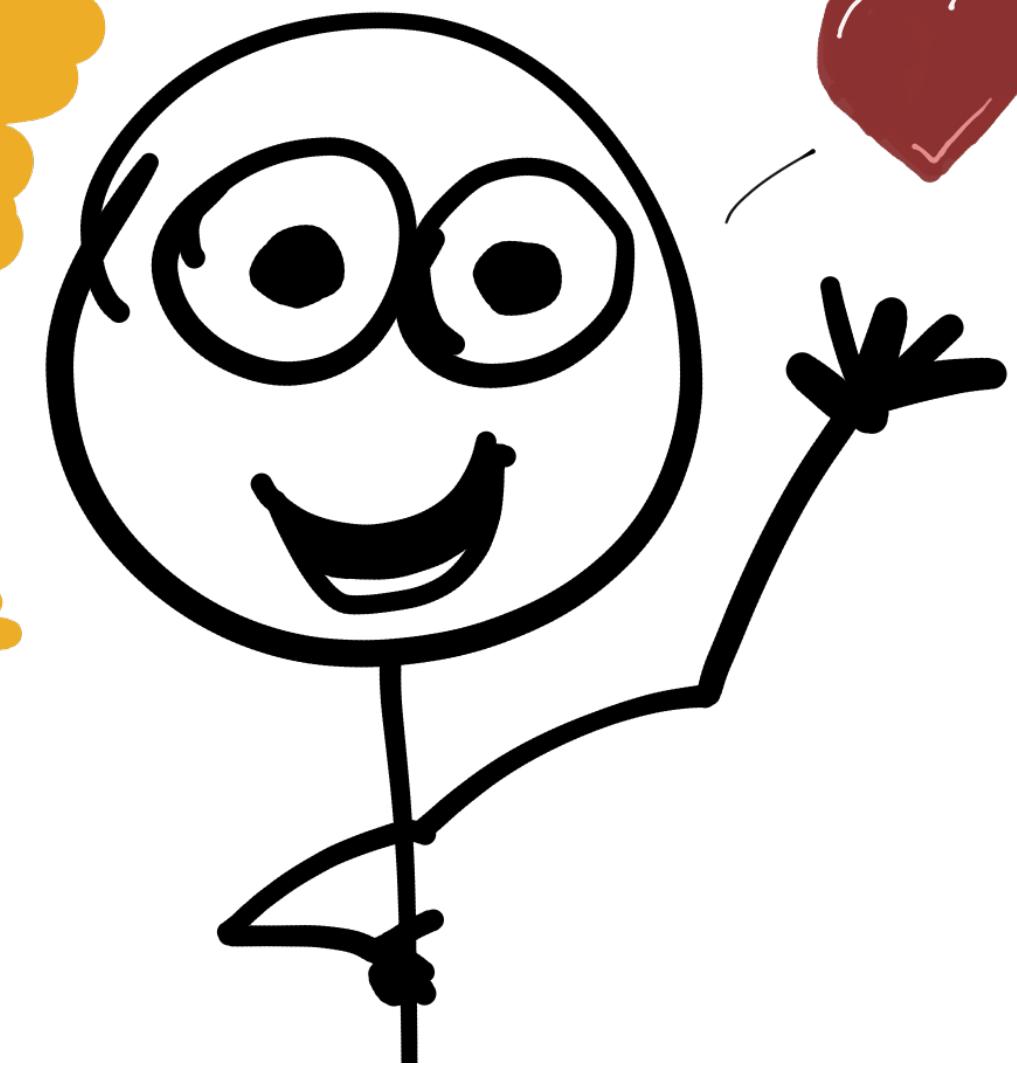






THANK

YOU



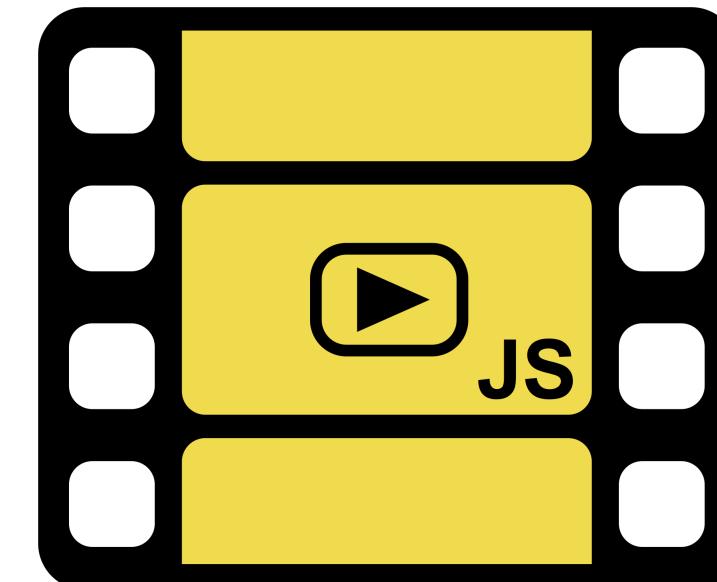
DerickBailey.com

Entrepreneur, Screencaster, Author, etc.



WatchMeCode.net

35% Off Discount Code: **jsremoteconf**



Grow Your Express Architecture

With These How-To Screencasts

- **Pro ExpressJS** (9 episodes)
- **Architecting Express** (11 episodes)
- **Deploying Express** (7 episodes)
- **RabbitMQ For Node.js** (12 episodes)
- **Mongoose.js Fundamentals** (11 episodes)
- ... and so much more!