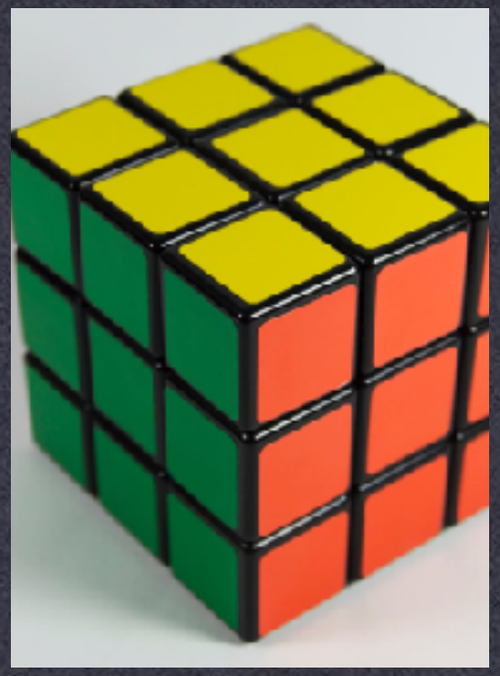
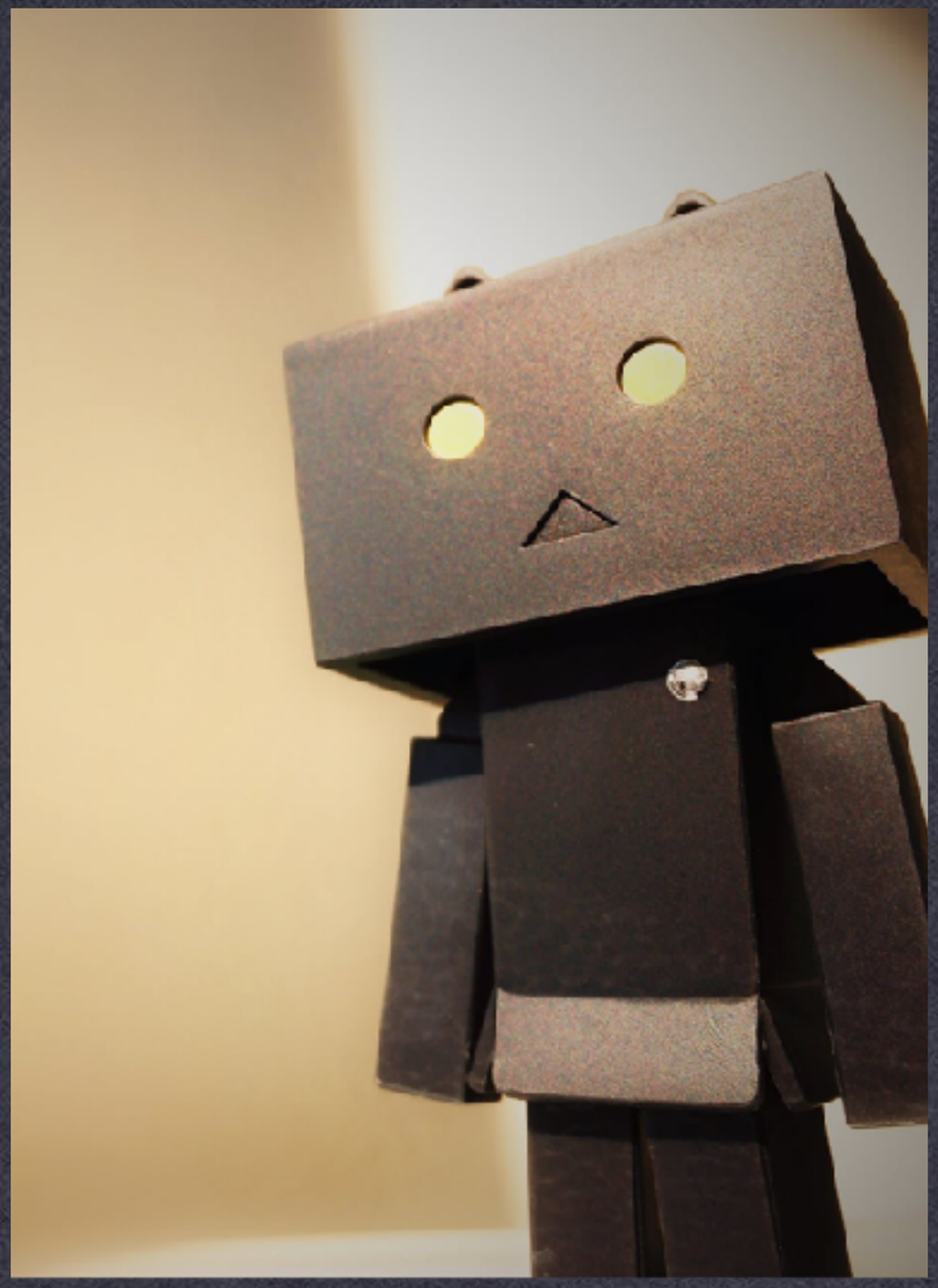




TITLE

30.099 PROBLEMS

AND DISPLAYING THEM AIN'T ONE

DATE **OCTOBER 18, 2017**

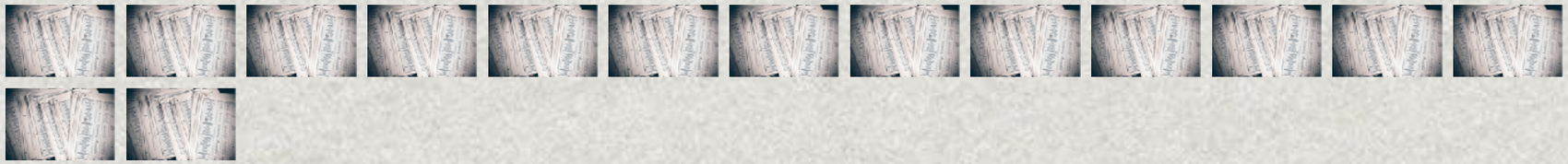
PLACE **MEETUP: EMBER LA**

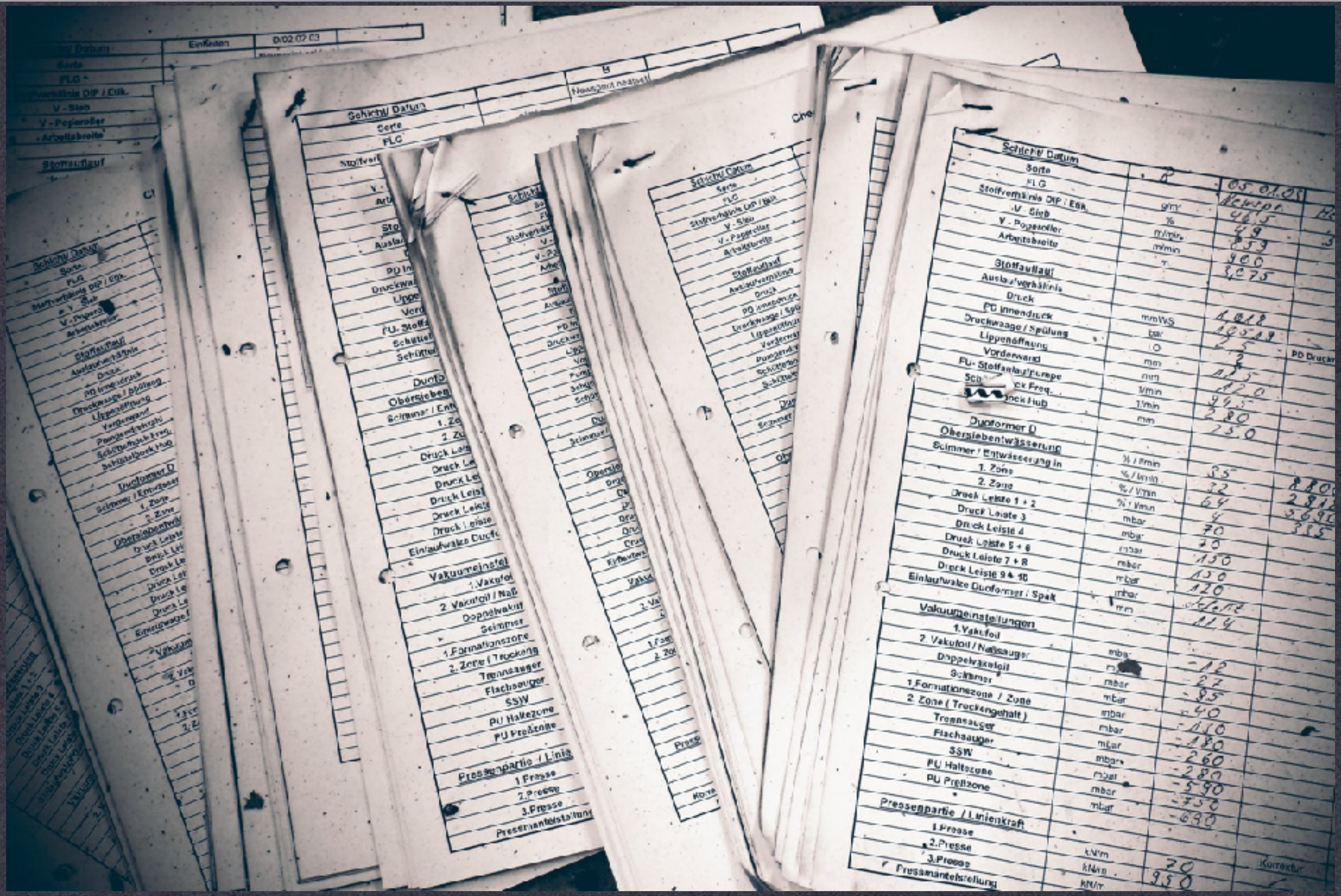
CUSTOMERS ARE AWESOME...

The best customers stick around awhile...

2012
2013
2014
2015
2016
2017

The longer they stick around, the more data they create...

2012	
2013	
2014	
2015	
2016	
2017	



OF COURSE, CUSTOMERS WOULD LIKE TO VIEW THEIR OWN DATA

The most loyal customers often have the most data.

This can create some tough UX challenges.

- How do I ship all that data from my API?
- How do I display hundreds or thousands of listings without crushing the browser with “DOM Node Overload”?

But first...

Who am I?

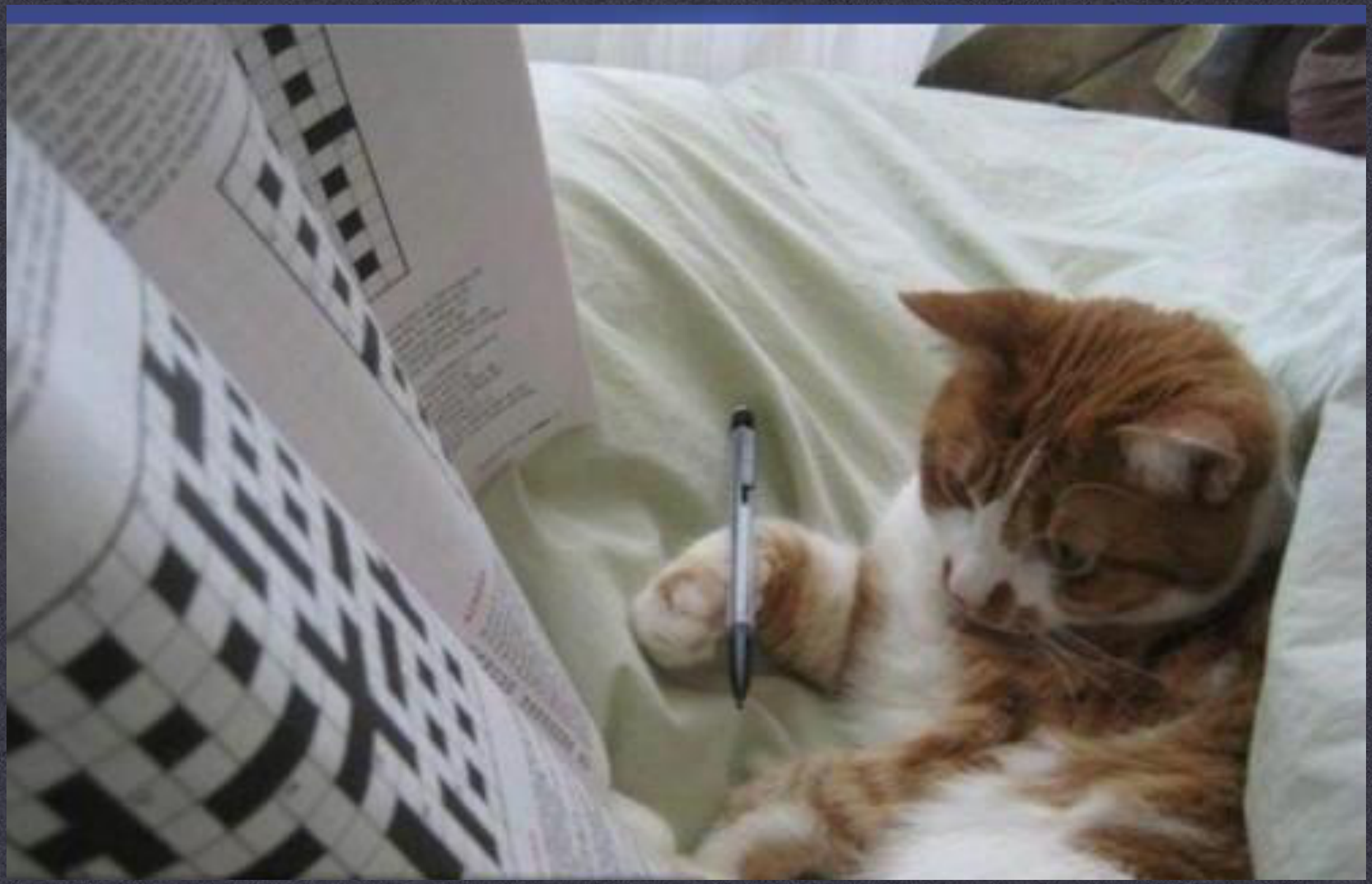
Name: Dana Franklin

Embereño since: 2012

Employer: TBWA \ Media Arts Lab (MAL)

Creator of the pile of code called "Emberella"

GitHub: realityendshere



BACK TO THE PROBLEM

HOW DO WE SHOW A CUSTOMER THEIR OWN DATA?

Some solutions

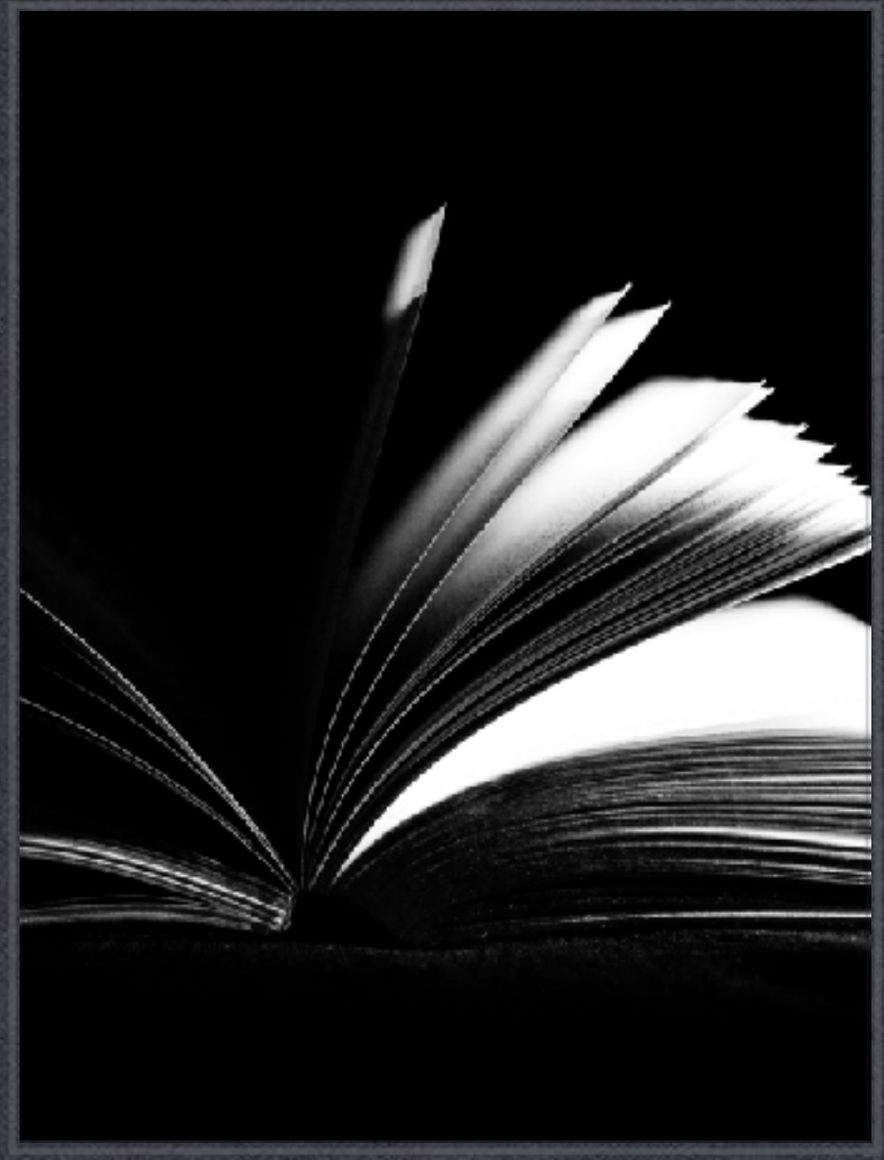
Pagination

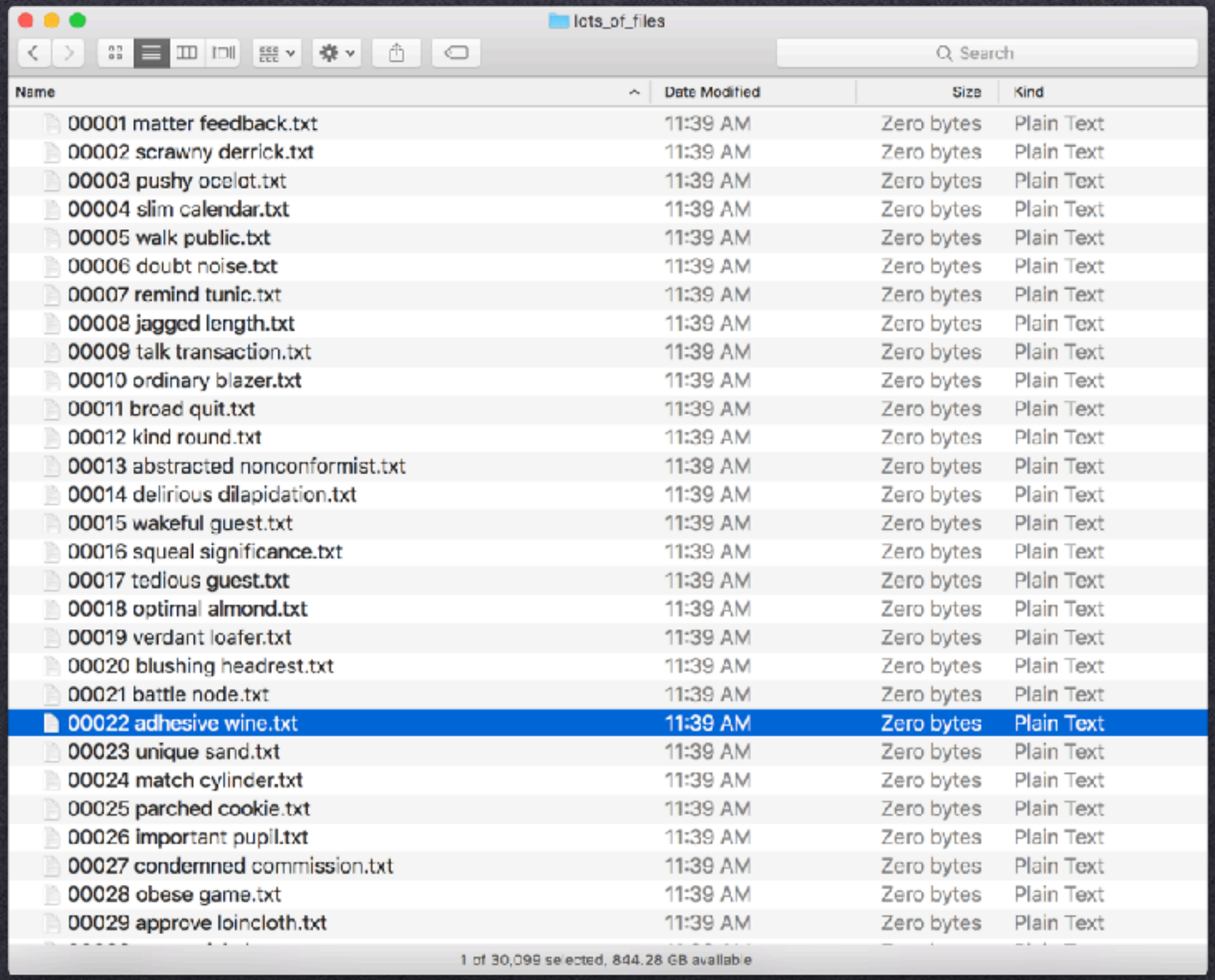
+ Good for records ranked by relevance

+ Loads data on demand

- UI/UX Variation

- Finding records in the middle of the pack





IMAGINE PAGINATION FOR YOUR OWN FILES

“Infinite” Scroll

+ Massive, aggregated data sets

+ Loads data on demand

- Can’t jump down the listing

- Difficult to share

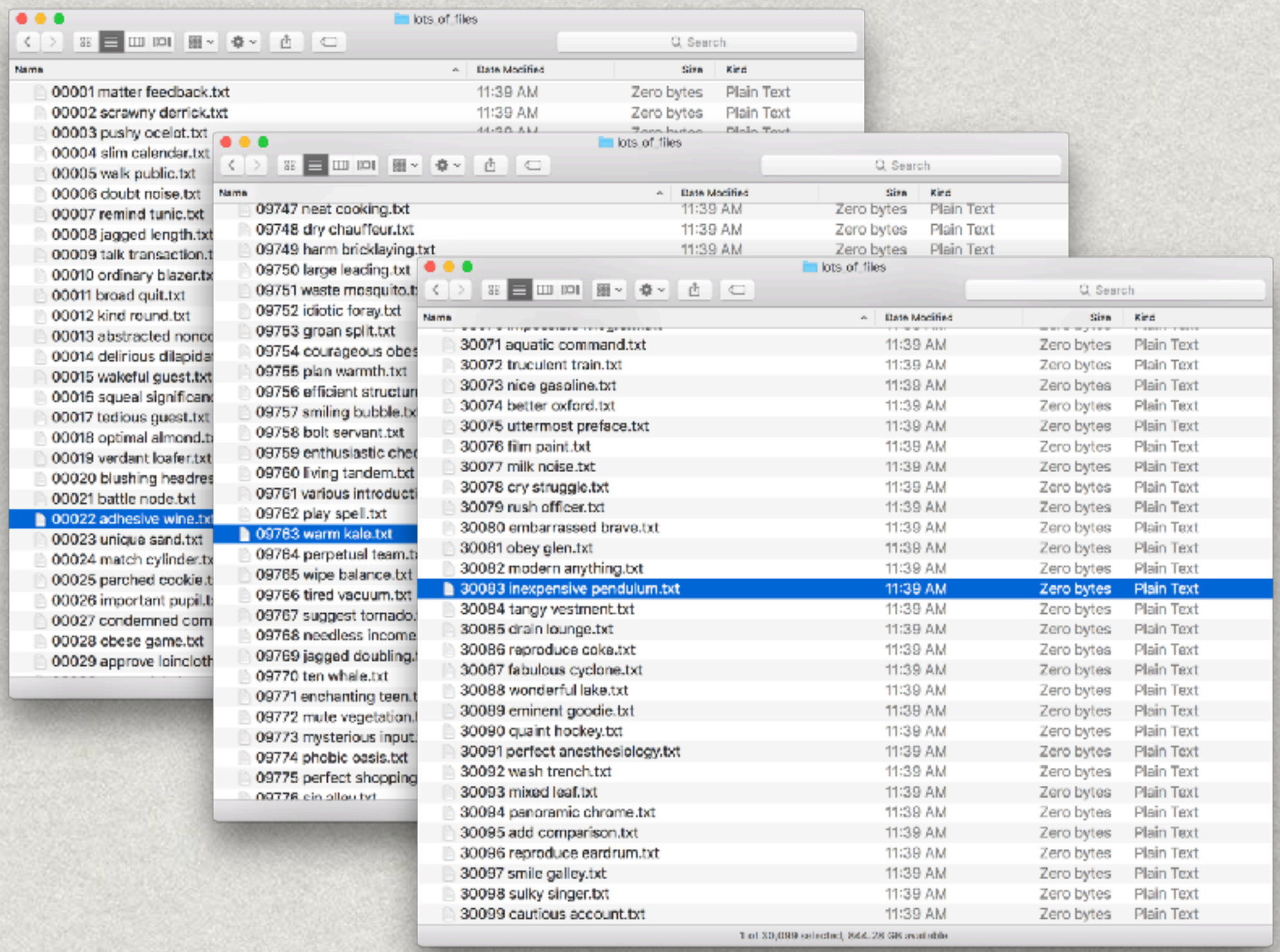
Giant List

+ Finite, customer-created records

+ Scroll to browse

- Stresses the API

- Browser performance



Browsing our own data in a giant list is familiar...

Your mission:

Show a list of 30,099 “problems”

```
model() {
  return this.store.findAll('problem');
},
```

Attempt #1

What typically happens with `this.store.findAll()`?

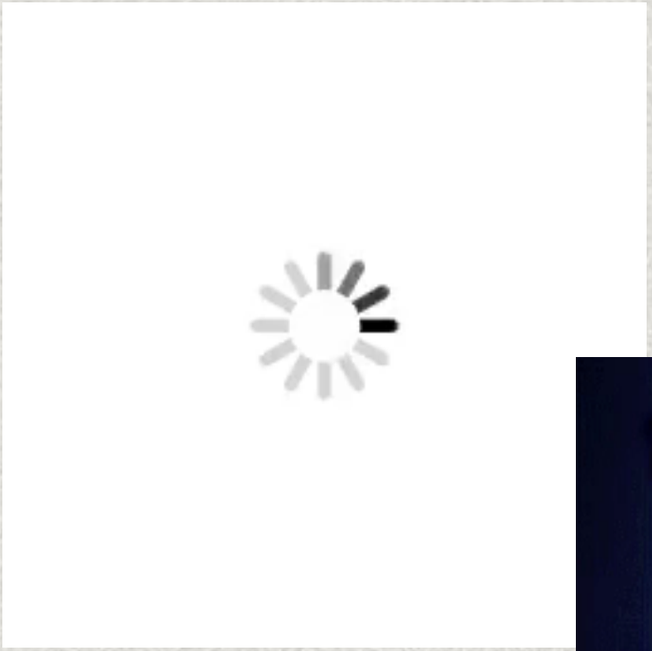
Let's find out!

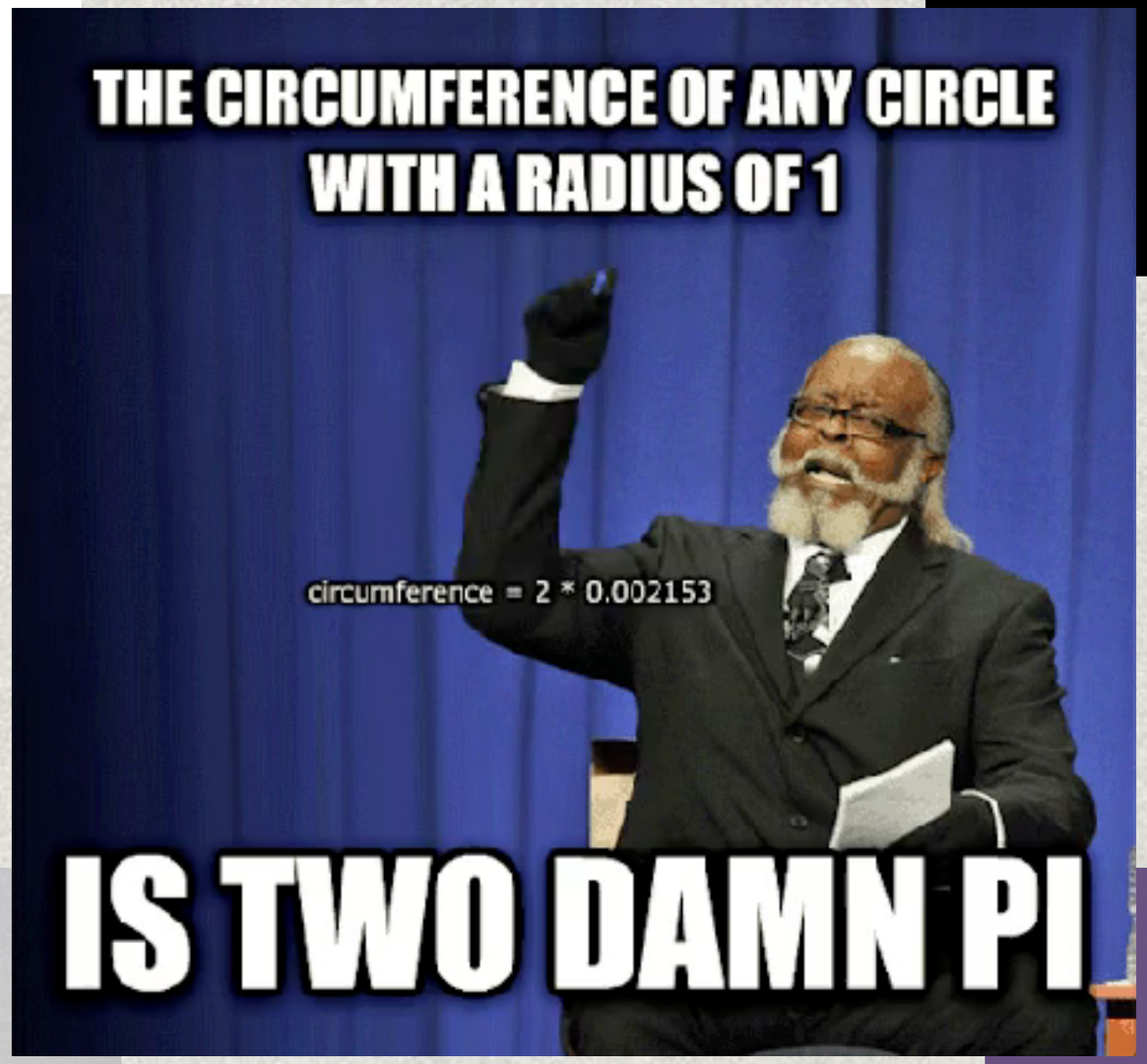
```
model() {
    return this.fetchAll();
},

fetchAll(offset = 0, limit = 200) {
    let query = { page: { offset: offset, limit: limit } };
    let loaded = this.store.peekAll('problem');

    return this.store.query('problem', query).then((results) => {
        if (get(loaded, 'length') < get(results, 'meta.total')) {
            return this.fetchAll(offset + limit, limit);
        } else {
            return this.store.peekAll('problem');
        }
    });
}
```

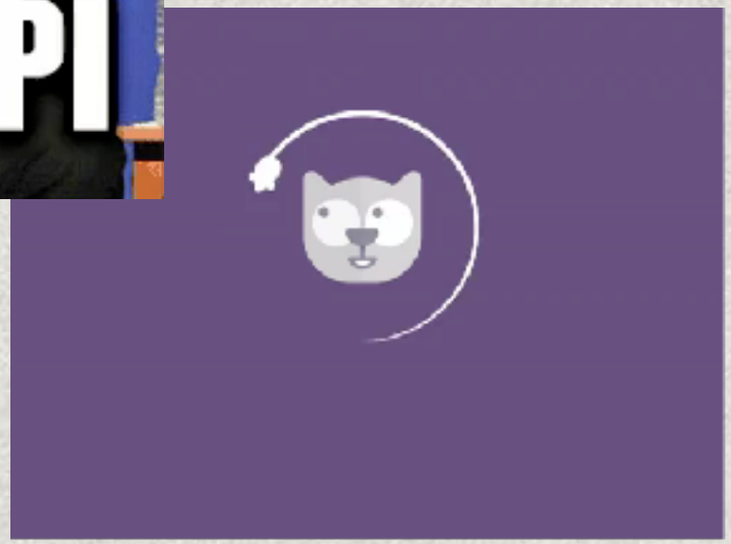
Attempt #2

Preload all the data... What could go wrong?

Let's find out!




THE CIRCUMFERENCE OF ANY CIRCLE
WITH A RADIUS OF 1
circumference = 2 * 0.002153
IS TWO DAMN PI



```
model() {
  return this.store.peekAll('problem');
},

afterModel() {
  this.fetchAll();
  return;
},

fetchAll(offset = 0, limit = 200) {
  let query = { page: { offset: offset, limit: limit } };
  let loaded = this.store.peekAll('problem');

  return this.store.query('problem', query).then((results) => {
    if (get(loaded, 'length') < get(results, 'meta.total')) {
      return this.fetchAll(offset + limit, limit);
    } else {
      return this.store.peekAll('problem');
    }
  });
}
```

Attempt #3

Load all the data after render

Let's find out!

Developer Tools - http://localhost:4200/load-all

Elements Console Sources Network Performance Memory Application Security Audits Ember

```
▶<div id="ember1328" class="problem-listing ember-view">…</div>
▶<div id="ember1330" class="problem-listing ember-view">…</div>
▶<div id="ember1332" class="problem-listing ember-view">…</div>
▶<div id="ember1334" class="problem-listing ember-view">…</div>
▶<div id="ember1336" class="problem-listing ember-view">…</div>
▶<div id="ember1338" class="problem-listing ember-view">…</div>
▶<div id="ember1340" class="problem-listing ember-view">…</div>
▶<div id="ember1342" class="problem-listing ember-view">…</div>
▶<div id="ember1344" class="problem-listing ember-view">…</div>
▶<div id="ember1346" class="problem-listing ember-view">…</div>
▶<div id="ember1348" class="problem-listing ember-view">…</div>
▶<div id="ember1350" class="problem-listing ember-view">…</div>
▶<div id="ember1352" class="problem-listing ember-view">…</div>
▶<div id="ember1354" class="problem-listing ember-view">…</div>
▶<div id="ember1356" class="problem-listing ember-view">…</div>
▶<div id="ember1358" class="problem-listing ember-view">…</div>
  Show All Nodes (29599 More)
  </main>
 </div>
</body>
</html>
```

html body.ember-application div#ember270.ember-view main

```
Styles »
:hov .cls +
element.style {
}
layout.sass:9
main {
  padding: ▶ 0;
  padding-bottom: 8rem;
}
layout.sass:6
main, header {
  margin: ▶ 0 2rem;
}
user agent s…
article,
aside,
footer
```

Console What's New

top | Filter | Default levels | 128 items hidden by filters

```
Mirage: [200] GET /problems?page%5Boffset%5D=29000&page%5Blimit%5D=200        server.js:30
▶{data: Array(200), meta: {…}}                                                 server.js:41
Mirage: [200] GET /problems?page%5Boffset%5D=29200&page%5Blimit%5D=200        server.js:30
▶{data: Array(200), meta: {…}}                                                 server.js:41
Mirage: [200] GET /problems?page%5Boffset%5D=29400&page%5Blimit%5D=200        server.js:30
▶{data: Array(200), meta: {…}}                                                 server.js:41
Mirage: [200] GET /problems?page%5Boffset%5D=29600&page%5Blimit%5D=200        server.js:30
▶{data: Array(200), meta: {…}}                                                 server.js:41
Mirage: [200] GET /problems?page%5Boffset%5D=29800&page%5Blimit%5D=200        server.js:30
▶{data: Array(200), meta: {…}}                                                 server.js:41
Mirage: [200] GET /problems?page%5Boffset%5D=30000&page%5Blimit%5D=200        server.js:30
▶{data: Array(99), meta: {…}}                                                  server.js:41
>
```

Node overload!

```
{{#ella-treadmill content=model row='92px' as |problem|}}
  {{problem-listing problem=problem}}
{{/ella-treadmill}}
```

Attempt #4

Introducing Emberella Treadmill

Demo

Better... but...

```
  ellaSparse: service('ella-sparse'),

  model() {
    let store = get(this, 'store');

    return get(this, 'ellaSparse').array((range = [], query = {}) => {
      let page = { limit: get(range, 'length'), offset: get(range, 'start') };
      let filter = { q: get(query, 'q') };

      return store.query('problem', { page: page, filter: filter }).then((result) => {
        return {
          data: result,
          total: get(result, 'meta.total')
        }
      });
    });
  }
```

Attempt #5

Introducing Emberella Sparse

Demo

Just a few more steps to Big Listing Nerdvana

```
model() {
  let store = get(this, 'store');

  return get(this, 'ellaSparse').array((range = {}, 
    // DATA FETCHING FUNCTION (SAME AS BEFORE)
  }, {
    ttl: 600000, //Time to live for fetched content
    limit: 25 //"Page size" for fetch queries
  });
},

actions: {
  handleScrollStart() {
    // Don't fetch data while scrolling
    set(this, 'controller.model.enabled', false);
  },

  handleScrollEnd() {
    set(this, 'controller.model.enabled', true);
  }
}
```

Attempt #6

Refine the Sparse Array config

Demo

Developer Tools - http://localhost:4200/sparse-better

Elements Console Sources Network Performance Memory Application Security Audits Ember

```
data-scroll-top="0" id="ember360" role="list" class="ella-treadmill not-resizing not-scrolling ember-
view">
  ▶<style type="text/css">…</style>
  ▶<ella-treadmill-item id="ember389" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-1 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember395" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-2 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember397" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-1 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember399" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-2 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember401" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-1 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember403" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-2 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember405" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-1 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember407" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-2 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember409" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-1 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  ▶<ella-treadmill-item id="ember411" role="listitem" class="ella-treadmill-item ella-treadmill-
  item-row-2 ella-treadmill-item-column-1 ember-view">…</ella-treadmill-item>
  </ella-treadmill>
```

html body #ember287 main #ember360 #ember389 div#ember394.problem-listing.ember-view p

Styles »

:hov .cls +

```
element.styl
e {
}
user agent s…
p {
  display:
    block;
  -webkit-
    margin-
    before:
    1em;
  -webkit-
    margin-
    after:
    1em;
  -webkit-
    margin-
    start:
    0px;
  -webkit-
    margin-
    end:
```

Console What's New

top ▼ Filter Default levels ▼ 5 items hidden by filters

```
Mirage: [200] GET /problems?page%5Blimit%5D=25&page%5Boffset%5D=0        vendor.js:127775
▶Object                                                                  vendor.js:127786
>
```

Just enough nodes. Just enough API requests.

```
{{#ella-treadmill content=problems row='92px' minColumnWidth='300px'
  on-scroll-start=(route-action 'handleScrollStart')
  on-scroll-end=(route-action 'handleScrollEnd')
  as |problem|
}}
  {{#if problem.is_loading}}
    {{loading-indicator class="blades"}}
  {{else}}
    {{problem-listing problem=problem}}
  {{/if}}
{{else}}
  {{#if model.loading}}
    <div class="loading data-loading">
      {{loading-indicator class="balls"}}
    </div>
  {{else}}
    <p>No problems today!</p>
  {{/if}}
{{/ella-treadmill}}
```

Attempt #7

Some final touches...

Demo

Giant List

+ Finite, customer-created records
+ Scroll to browse
+ Loads data on demand
+ Browser performance

Some known issues

- Uses inline styling, conflicts with Content Security Policies
- Images may momentarily appear again as they scroll off one edge of the viewport and in from the other*

* Try github.com/realityendshere/ella-image

Get Treadmill & Sparse

Emberella Treadmill	• github.com/realityendshere/ella-treadmill • ember install ember-ella-treadmill
Emberella Sparse	• github.com/realityendshere/ella-sparse • ember install ember-ella-sparse

Q & A

Thank you!

GitHub: realityendshere