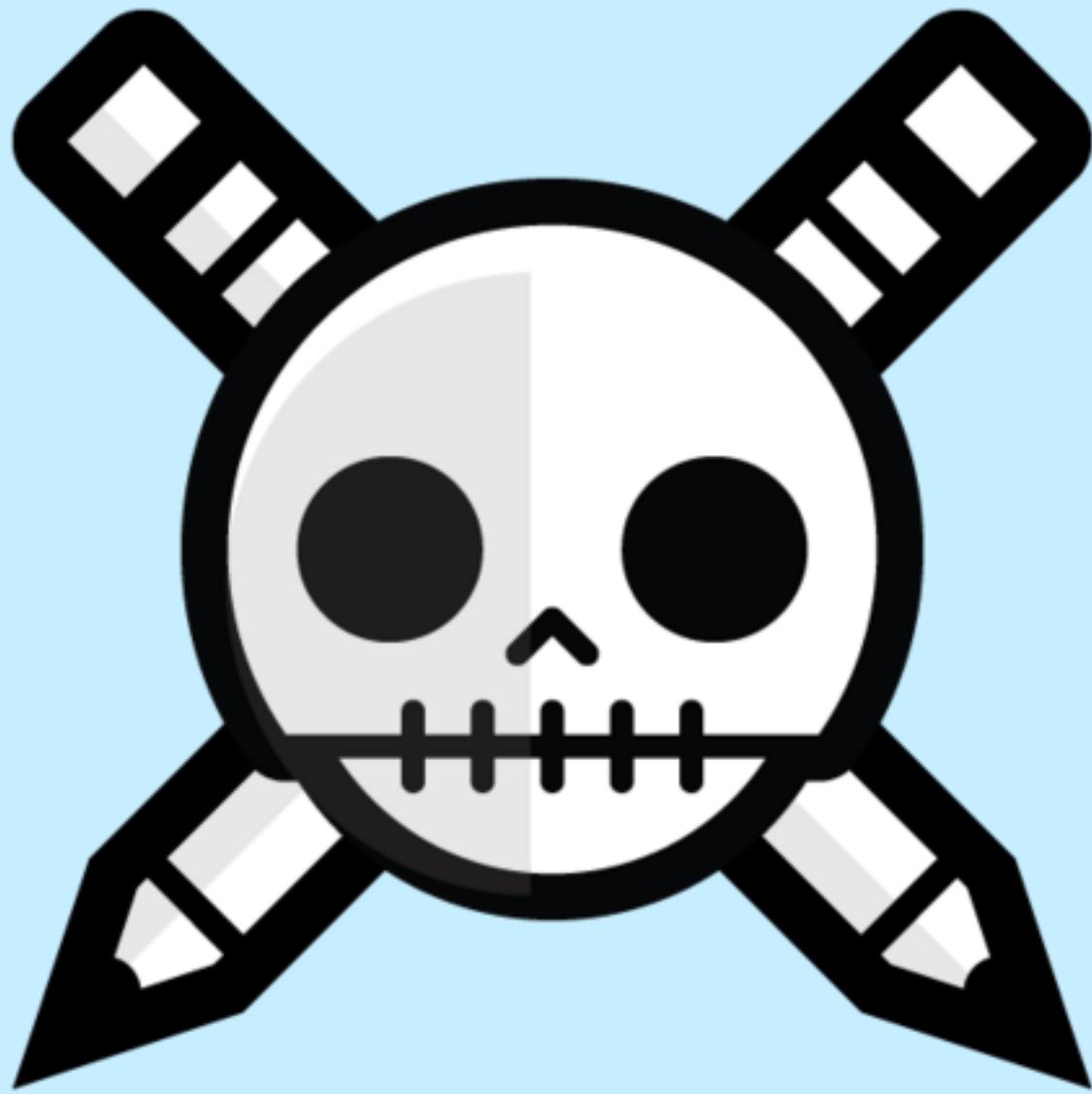


doodle tech

or, how to build a playground of surreal
and grotesque and lovely

by josh essex
github.com/jessex
josh@doodleordie.com



doodle or die



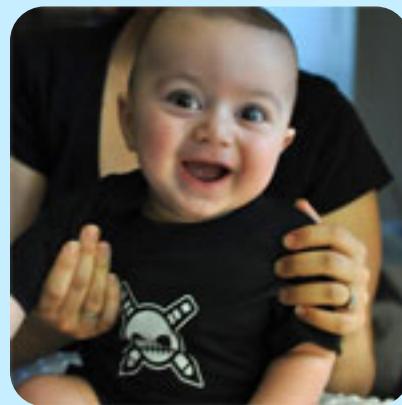
dylan greene



aaron silverman



garrett miller



asher greene

doodle or die



dylan greene

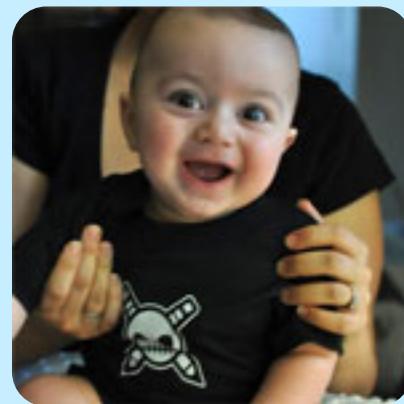


aaron silverman



garrett miller

+ wyn
bennett



asher greene

doodle or die



dylan greene

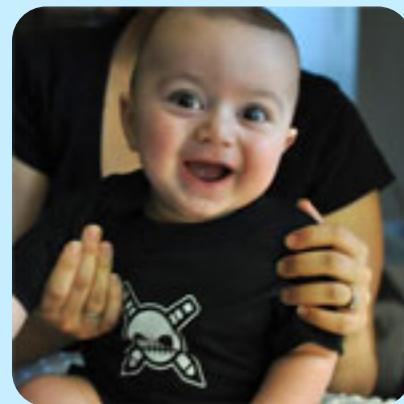


aaron silverman



garrett miller

+ wyn
bennett



asher greene

+ josh
essex

doodle or die

the game of telephone, only with
drawings

doodle or die

the game of telephone, only with
drawings

describe, doodle, describe, doodle...

doodle or die

Doodle or Die

doodleordie.com/play

Play Admin About josh.essex

You get to start a new Doodle Chain! What should the next player draw? Skip this

140

Submit description

NEW CHAIN NEW CHAIN

9 steps 0 likes Chain page: <http://doodleordie.com/c/EhHlctAUhE>

1 Jaspers Avarei 0

2 larissa.hallford.5 0

▼ YOURS

The screenshot shows a web browser window for the 'Doodle or Die' website. At the top, there's a navigation bar with links for 'Play', 'Admin', 'About', and a user account 'josh.essex'. Below this, a message asks the user to start a new Doodle Chain, with a 'Skip this' button. A large text input field is provided for drawing a doodle, with a character limit of 140. A red 'Submit description' button is at the bottom right of the input field. Below the input field is a row of nine small thumbnail doodles, with the second one labeled 'NEW CHAIN'. Underneath this row, it says '9 steps 0 likes' and provides a 'Chain page' URL. The main content area displays two steps of the current doodle chain. Step 1 shows a doodle of a person with the name 'Jaspers' written next to it, and the name 'Avarei' with a '0' likes count. Step 2 shows a doodle of a cat-like creature, and the name 'larissa.hallford.5' with a '0' likes count. At the bottom right, there's a pink button with a downward arrow and the text '▼ YOURS'.

3

Reptar fell down because he wanted cheese.

FiftySixer

♡ 6

4



Niko86M

♡ 29

5

a cute godzilla craves for some cheddar cheese.

jimbobsonofgod

♡ 6

6



Svartmes

♡ 35

7

cute godzilla thinking about cheese

olafgranciglio

♥ 6

8



jambandit

♥ 13

9

a cute dinosaur wants some classic cheese.

Luce

5

10



falco

15

11

A T-Rex wishes there were mammals around so that he could have some cheese.

stroopid

10

12

I WANT CREAMY GORGONZOLA CHEESE!!
I WISH THERE WERE MAMMALS AROUND...

iIVice

136



13

Really awesome looking T-rex, saddened because there's no gorgonzola cheese around.

Intemoob

♡ 8

14



underwearhero

♡ 217

15

T-rex wants some gorgonzola

Gustavo

3

16



Preview Player

17

the tech stack

the tech stack

node.js

the tech stack

node.js

express.js

the tech stack

node.js

express.js

mongodb

the tech stack

node.js

express.js

mongodb

amazon s3

node.js

node.js

web platform built on v8

node.js

web platform built on v8

server-side javascript

node.js

web platform built on v8

server-side javascript

event-driven

node.js

web platform built on v8

server-side javascript

event-driven

asynchronous

event-driven

event-driven

actions triggered by events

event-driven

actions triggered by events

cause and effect:
event detected, so handler activated

event-driven

actions triggered by events

cause and effect:
event detected, so handler activated

callback functions

asynchronous

asynchronous

no blocking, ever

asynchronous

no blocking, ever

independent of main thread;
application can continue processing

asynchronous

no blocking, ever

independent of main thread;
application can continue processing

callback functions

how node does it

how node does it

I/O is event-driven and asynchronous

how node does it

I/O is event-driven and asynchronous

no locks, no blocks, one process

how node does it

I/O is event-driven and asynchronous

no locks, no blocks, one process

acts on **callbacks**

callbacks

callbacks

functions passed into other functions as
parameters

callbacks

functions passed into other functions as parameters

called when the calling function finishes

callbacks

functions passed into other functions as parameters

called when the calling function finishes

finish = event, callback = handler

callbacks

functions passed into other functions as parameters

called when the calling function finishes

finish = event, callback = handler

called indeterminate time in the future

example

10 GOTO TERMINAL

20 IMPRESS AUDIENCE

drilling it home

node server is asynchronous call

**server passed callback defining
response to requests**

request handling spawns callbacks

callbacks all the way down

node pros

node pros

highly concurrent, efficient

node pros

highly concurrent, efficient

server-client consistency

node pros

highly concurrent, efficient

server-client consistency

encourages proper error handling

node pros

highly concurrent, efficient

server-client consistency

encourages proper error handling

robust module system: npm

node cons

node cons

paradigm is complex

node cons

paradigm is complex

potentially fragile

node cons

paradigm is complex

potentially fragile

single-threaded, event-driven model:
great for I/O, not for CPU

interlude

interlude



interlude

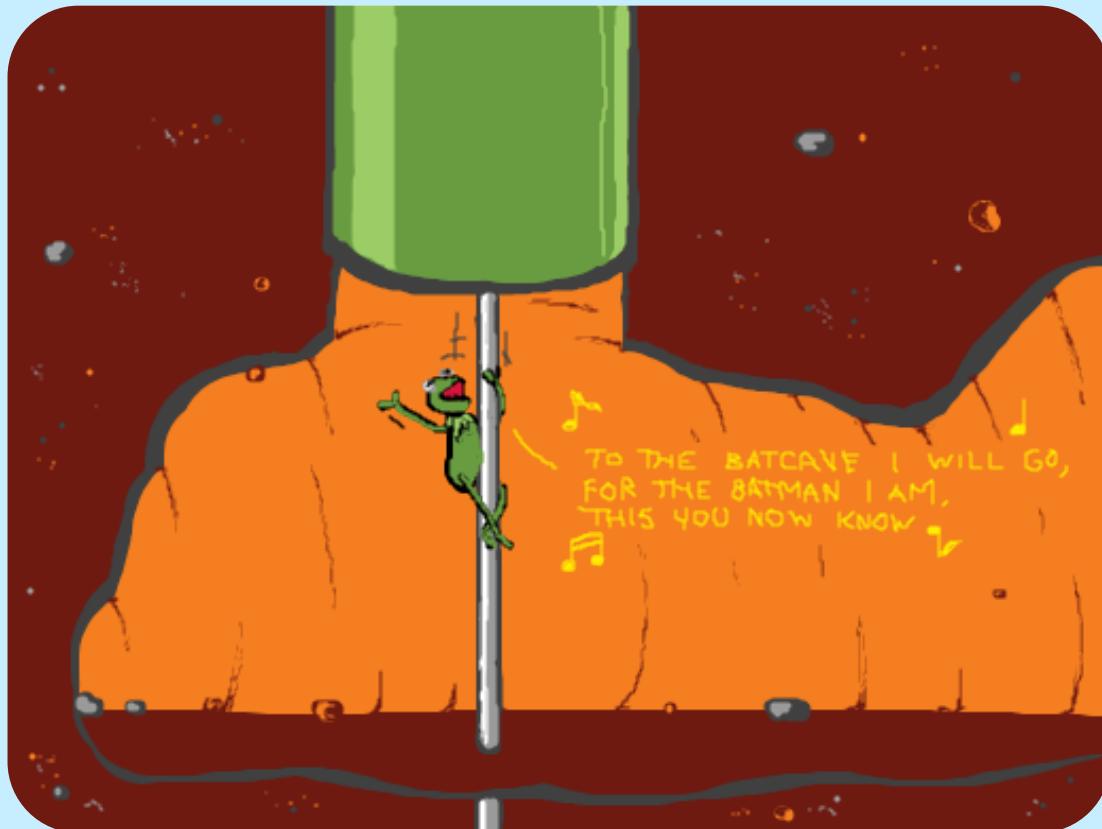


interlude



M, I NAMTAB M, I NAMTAB M, I NAMTAB
YAMTAB M, I NAMTAB M, I NAMTAB M, I NAMTAB
-TAB M, I NAMTAB M, I NAMTAB M, I NAMTAB M,
(IINAMTAB M, I NAMTAB M, I NAMTAB M, I NAMTAB

interlude

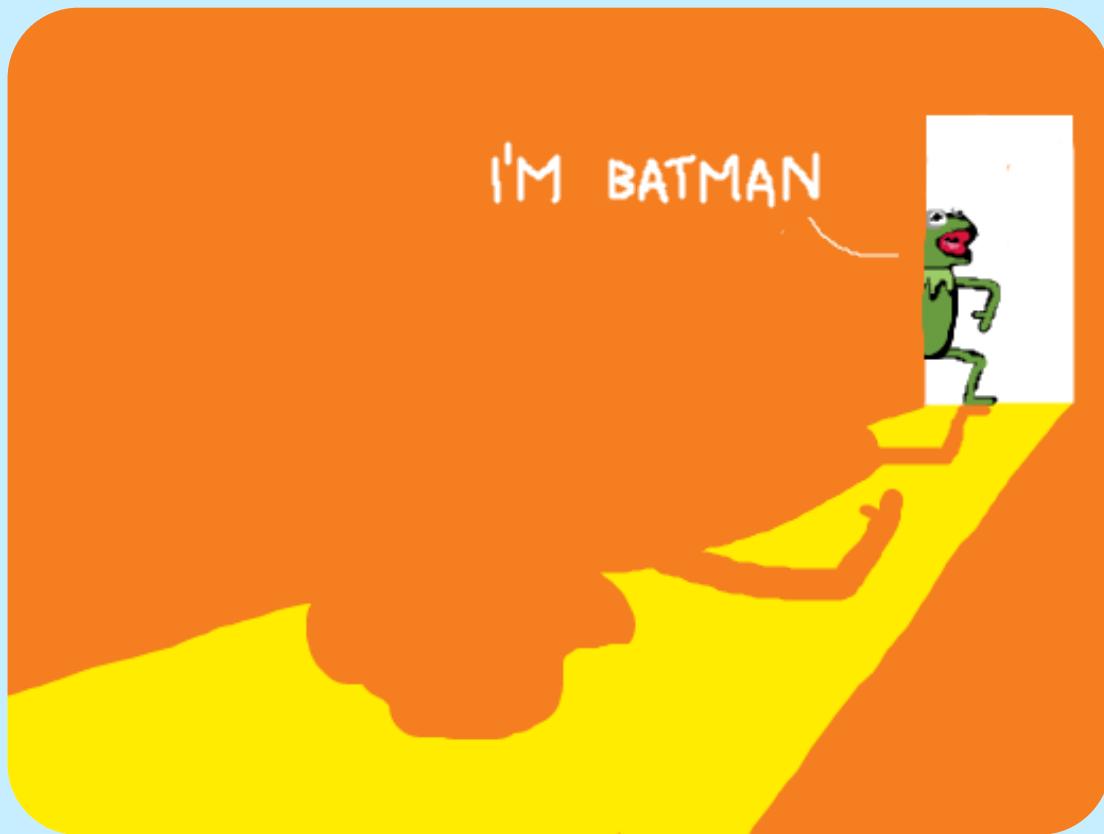


interlude

THIS JUST IN:
I'M BATMAN!



interlude



interlude

BAT CAVE, BATMAN SPEAKING...



interlude

I'M BATMAN



interlude



interlude



express.js

express.js

web framework node module

express.js

web framework node module

asynchronous by design

express.js

web framework node module

asynchronous by design

fully-featured

what it provides

what it provides

comprehensive routing

what it provides

comprehensive routing

environment-specific configuration

what it provides

comprehensive routing

environment-specific configuration

middleware

what it provides

comprehensive routing

environment-specific configuration

middleware

easy templating

routing

routing

set response by relative path and verb

routing

set response by relative path and verb

callbacks define response

routing

```
app.get('/', function(req, res) {
  // response for home page
});

app.get('/about', function(req, res) {
  // response for about page
});

app.post('/user/create', function(req, res) {
  // logic for creating user and responding
});

app.put('/user/:user_id/edit', function(req, res) {
  // logic for editing user and responding
});

app.delete('/user/:user_id/delete', function(req, res) {
  // logic for deleting user and responding
});
```

example

10 GOTO TERMINAL

20 DO BETTER THIS TIME

middleware

middleware

bits of code that extend functionality

middleware

bits of code that extend functionality

expressed as callbacks

middleware

bits of code that extend functionality

expressed as callbacks

defined in order;

applied to each request sequentially

middleware

```
app.use(express.logger());
app.use(express.static(__dirname + '/public'));
app.use(function(req, res) {
  res.set({
    'Cache-Control': 'no-cache'
  })
});
app.use(function(err, req, res, next) {
  console.error(err.stack);
  res.send(500, 'Oops!');
});
```

so this happens...

```
// Test for bad urls such as /bad%url
// the % doesn't decode correctly, throwing an exception
App.use(function(req, res, next) {
  try {
    var u = decodeURIComponent(req.url);
    if (u.length > 1000 || u.indexOf('data:') > -1) {
      return res.send(404);
      //return ErrorHelper.err('Data url: ' + req.url.substr(0, 40) + '...', req, res);
    }
  } catch(e) {
    return res.send(404);
    //return ErrorHelper.err('Bad url: ' + req.url, req, res);
  }
  next();
});

App.use(function(req, res, next) {

  // For logging in - check for cookie, must have it
  if (req.url === '/welcome' && !req.session.info) {
    res.redirect('/err/cookies');
  } else if (req.url === '/welcome' ) {
    res.redirect('/play');
  } else {
    next();
  }
});

App.use(UserHelper.hostname);
App.use(UserHelper.updateInfo);
App.use(EveryAuth.middleware());
App.use(UserHelper.recallUser);
App.use(App.router);

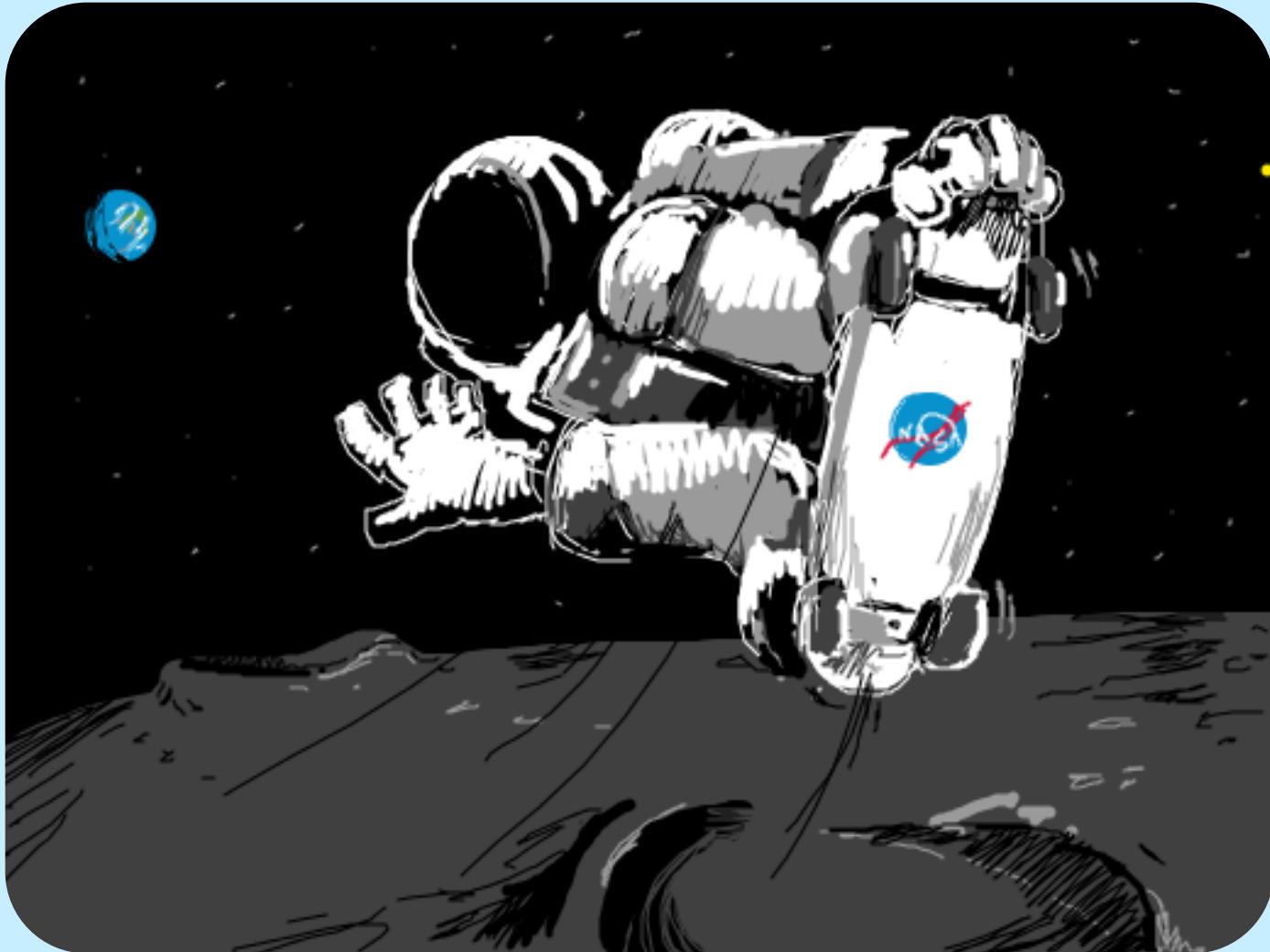
// handle 404's
App.use(function(req, res, next) {
  // Old browsers don't get to log 404 errors
  if (req.session && req.session.info && req.session.info.oldBrowser) {
    return res.send(404);
  }

  ErrorHelper.err('404 - Page not found', req, res);
});
```

... when this happens



... and this



... and especially this



templates

templates

html with minimal logic

templates

html with minimal logic

variables, conditions

templates

html with minimal logic

variables, conditions

used to dynamically render a page

templates

partials are *part* of a page

templates

partials are *part* of a page

encourage re-use and consistency

templates

partials are *part of a page*

encourage re-use and consistency

can be rendered on server or client

how doodle does it

how doodle does it

ejs

how doodle does it

ejs

client-side javascript with jquery

how doodle does it

ejs

client-side javascript with jquery

asynchronously call api to get json

how doodle does it

ejs

client-side javascript with jquery

asynchronously call api to get json

populate partials with data

so this...

```
<% if (!everyauth.loggedIn && !user) { %>
<div id="what-is-this">
  <div class="wrapper container">
    <h2>
      It's like the game of <em>Telephone</em>. Except with doodles.
      <a href="/about">(Learn more)</a>
    </h2>
  </div>
</div>
<% } %>

<div id="chain-header">
  <div class="wrapper container">

    <% if (chain.groupInfo && (chain.playerGroup || (user && user.isAdmin))) { %>
      <div class="group-name span-13 last chain-group<%= user && user.isAdmin && !chain.playerGroup ? ' admin-peek' : '' %>">
        <% /* Show chain group. Note: currently only support 0 or 1 groups for chain */ %>
        <span>Group</span>
        <a href="/g/<%= chain.groupInfo._id %>"><strong><%= chain.groupInfo.displayName %></strong></a>
      </div>
    <% } %>

    <div class="span-3">
      <ul class="stats">
        <li class="steps-played">
          <strong><%= chain.steps.length || 0 %></strong>
          <span>steps</span>
        </li>
        <li class="likes">
          <strong class="js-global-like-listener"><%= chain.stats.numLikes || 0 %></strong>
          <span>likes</span>
        </li>
      </ul>
    </div>

    <div class="share span-10 last">
      <% if (pageName !== 'chain') { %>
        Chain page: <a href="<%= Env.HOST %>/c/<%= chain._id %>"><%= Env.HOST %>/c/<%= chain._id %></a>
      <% } %>
    </div>
  </div>
</div>
```

... becomes this

DOODLE OR DIE

About Create account Sign In

It's like the game of *Telephone*. Except with doodles. [\(Learn more\)](#)

18 steps 47 likes

1 A badger with a monocle and a moustache stealing KFC chicken.

bigtits27
1

2 

underwearhero
42

3 

Birdiyo

express pros

express pros

abstracts away much dev pain

express pros

abstracts away much dev pain

covers some node deficiencies

express pros

abstracts away much dev pain

covers some node deficiencies

lightweight, flexible

express cons

express cons

almost too configurable

interlude

interlude

doodle time!

mongodb

mongodb

nosql data store

mongodb

nosql data store

non-relational

mongodb

nosql data store

non-relational

minimal, json-like structure

mongo structure

*sql	mongo
database	database
table	collection
row	document
column	field
join	embed and/or link

collections

collections

groups of documents without constraints

collections

groups of documents without constraints

**no declaration of fields;
client defines fields for read/write**

collections

groups of documents without constraints

**no declaration of fields;
client defines fields for read/write**

tend toward homogeneity

documents

documents

json objects and arrays

documents

json objects and arrays

contain fields and subdocuments

embedding

embedding

nesting subdocuments within documents

embedding

nesting subdocuments within documents

**can retrieve like any other field;
can also specify subdocument fields**

embedding

nesting subdocuments within documents

**can retrieve like any other field;
can also specify subdocument fields**

easily handled by server

linking

linking

reference one document from another

linking

reference one document from another

must be processed by client;
no foreign key-like enforcement

linking

reference one document from another

must be processed by client;
no foreign key-like enforcement

follow-up queries

example

```
> db.comments.save({"_id": "l4k5J78L0", "chain_id": "EhwLLIYk_",
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   "player_id": "EiUGQKG3_", "text": "This chain is swagtastic!"});
```

example

```
> db.players.findOne({_id: 'EiUGQKG3_'}, {chainHistory: 1});  
  
{  
  "_id" : "EiUGQKG3_",  
  "chainHistory" : {  
    "EG510fG3Z" : {  
      "datePlayed" : ISODate("2012-10-08T23:37:54.054Z"),  
      "dateViewed" : ISODate("2012-10-09T01:04:18.120Z"),  
      "numLikes" : 1,  
      "statePlayed" : "draw",  
      "stepIndexPlayed" : 2  
    },  
    "EhwLLIYk_" : {  
      "datePlayed" : ISODate("2012-10-12T02:56:12.379Z"),  
      "dateViewed" : ISODate("2012-10-21T04:45:29.747Z"),  
      "statePlayed" : "draw",  
      "stepIndexPlayed" : 0  
    }  
  }  
}
```

example

```
> db.players.find({"game.activeState": "phrase"}, {"login.password.uid": 1});  
{ "_id" : "EiUGQKG3_", "login" : { "password" : { "uid" : "josh.essex" } } }
```

how doodle does it

how doodle does it

question-oriented document structure

how doodle does it

question-oriented document structure

**able to answer a question about a
player or chain with one query**

how doodle does it

question-oriented document structure

able to answer a question about a
player or chain with one query

heavy embedding of subdocuments

example

what is this player working on right now?

example

what is this player working on right now?

```
db.players.findOne({_id: '58mwYlTKV'}, {game: 1});
```

example

how active is this player?

example

how active is this player?

```
db.players.findOne({_id: '58mwYlTKV'}, {stats: 1});
```

example

**what recently modified chains need a
new doodle?**

example

what recently modified chains need a new doodle?

```
db.chains.ensureIndex({inUse: -1, activeState: 1, lastModified : -1});
```

saving doodles

saving doodles

collection of chains

saving doodles

collection of chains

**chain embeds steps of descriptions and
doodles**

saving doodles

collection of chains

**chain embeds steps of descriptions and
doodles**

steps embed content

amazon s3

amazon s3

metadata stored in step in mongo

amazon s3

metadata stored in step in mongo

doodle is series of brushstroke objects

amazon s3

metadata stored in step in mongo

doodle is series of brushstroke objects

brushstrokes saved in amazon s3

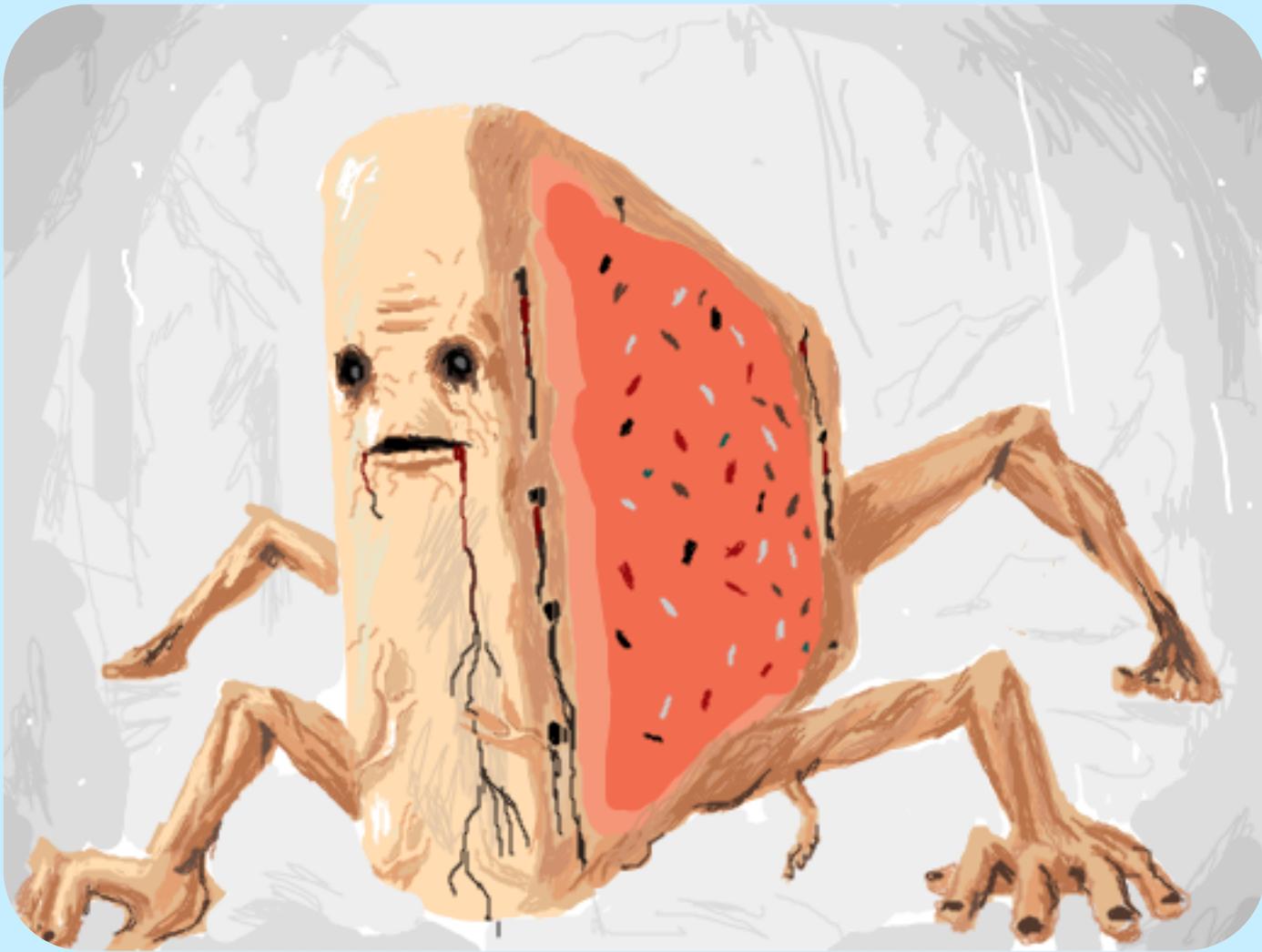
a series of these...

```
{"color": "#000000", "size": 15, "path": [307, 66, 308, 66, 308, 68, 308, 69, 308, 70, 308, 71, 308, 72, 308, 73, 306, 76, 305, 79, 305, 81, 302, 83, 302, 84, 302, 85, 302, 86, 301, 87, 300, 89, 300, 90, 300, 91, 300, 92, 300, 95, 300, 97, 300, 102, 300, 103, 300, 104, 300, 108, 300, 109, 300, 110, 301, 114, 303, 116, 304, 116, 305, 119, 305, 121, 307, 124, 309, 127, 310, 130, 310, 131, 313, 132, 314, 133, 316, 137, 317, 138, 320, 140, 321, 140, 323, 143, 326, 143, 328, 143, 333, 144, 337, 144, 341, 144, 343, 144, 347, 143, 352, 143, 354, 141, 357, 140, 358, 140, 359, 139, 362, 138, 363, 138, 365, 137, 368, 135, 369, 132, 370, 131, 371, 130, 374, 128, 375, 128, 376, 125, 378, 125, 379, 124, 380, 123, 381, 123, 382, 120, 385, 119, 386, 118, 388, 115, 391, 112, 394, 109, 394, 107, 394, 106, 397, 104, 397, 103, 397, 102, 397, 101, 397, 100, 397, 98, 397, 96, 397, 94, 397, 93, 397, 91, 397, 90, 397, 88, 397, 87, 397, 86, 397, 83, 395, 80, 395, 79, 395, 77, 394, 74, 393, 72, 393, 70, 393, 67, 392, 65, 391, 63, 389, 62, 388, 59, 386, 57, 383, 54, 383, 52, 381, 49, 379, 48, 378, 47, 376, 46, 375, 44, 374, 43, 372, 43, 370, 42, 369, 42, 368, 42, 364, 41, 362, 41, 359, 41, 355, 41, 351, 42, 349, 42, 347, 42, 346, 44, 343, 44, 342, 45, 339, 46, 337, 47, 336, 48, 333, 49, 332, 51, 330, 51, 328, 51, 327, 52, 326, 53, 323, 53, 322, 53, 321, 54, 320, 55, 317, 55, 316, 56, 314, 58, 309, 59, 306, 61, 306, 62, 305, 62, 304, 63]}}
```

... equals this



... and also this



other tips

other tips

store. everything.

other tips

store. everything.

always specify fields

other tips

store. everything.

always specify fields

use indices

mongo pros

mongo pros

optimized for most common use case

mongo pros

optimized for most common use case

fits well with javascript development

mongo pros

optimized for most common use case

fits well with javascript development

great for rapid prototyping

mongo pros

optimized for most common use case

fits well with javascript development

great for rapid prototyping

horizontally scalable

mongo cons

mongo cons

non-obvious data structure

mongo cons

non-obvious data structure

client must maintain data integrity

mongo cons

non-obvious data structure

client must maintain data integrity

non-atomic for multiple collections

mongo cons

non-obvious data structure

client must maintain data integrity

non-atomic for multiple collections

reduced query flexibility

other fascinations

other fascinations

hosting on joyent

other fascinations

hosting on joyent

version control with git

other fascinations

hosting on joyent

version control with git

hot deploys via git push

other fascinations

async

other fascinations

async

nodemon

other fascinations

async

nodemon

passport

log everything

```
Terminal — ssh — 175x54
bash ssh

27 Oct 11:48:57 - [4.playerManager] - Player_id 8hBRmWEyZ has resumed playing
27 Oct 11:48:58 - [5.playerManager] - player_id 157nsD4J8 active chain_id 5iPC74oxiE is no longer assigned to them, treating them as stale
27 Oct 11:49:00 - [5.game-api] - player_id 8ikwlodg (damashek) started working on private_id 1q1rfpLIsiB
27 Oct 11:49:04 - [1.game-api] - player_id iiZyGC5nZ (TempAccount154660) started working on private_id 5i10IIPJxq1
27 Oct 11:49:06 - [5.gameHelper] - Stairs wrote: A fish jumping into water. http://doodleordie.com/c/5iST0yJsG1
27 Oct 11:49:06 - [5.game-api] - player_id 50upC1zsP (iwillpwn2) started working on private_id 8h89Csj6DhE

A highly pixelated, low-resolution image of a group of people, possibly a protest or crowd, viewed through a terminal window. The image is composed of large, colorful pixels in shades of red, orange, yellow, and black.

27 Oct 11:49:10 - [5.gameHelper] - Paxman doodled 943 strokes http://doodleordie.com/c/5Ge_w-Drh5
27 Oct 11:49:11 - [4.game-api] - player_id 8hBRmWEyZ (jnnrss) started working on private_id 1cEgr794n
27 Oct 11:49:12 - [5.playerManager] - player_88FP7P34n active chain_id 5icg3Yoxq5 is no longer assigned to them, treating them as stale
27 Oct 11:49:13 - [5.game-api] - player_id EiPjls2Yd (sparx18) started working on private_id EhEgp6EIrG1
27 Oct 11:49:14 - [5.gameHelper] - iwillpwn2 wrote: OP is gay http://doodleordie.com/c/Eiemxy6rh5
27 Oct 11:49:17 - [5.gameHelper] - damashék wrote: two men shopping for milk http://doodleordie.com/c/EqxfpwIxhE
27 Oct 11:49:20 - [5.gameHelper] - TempAccount155273 wrote: pollywog looks to the future http://doodleordie.com/c/8GrkJKVXh1
27 Oct 11:49:25 - [5.game-api] - player_id 8ikwlodg (damashek) started working on private_id 131jYvLYv7
27 Oct 11:49:25 - [5.userHelper] - Referrer: http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CB8QFjAA&url=http%3A%2F%2Fdoodleordie.com%2F&ei=QauMUKe7DY0Q9gTf9YDADQ&usg=AFQjCNE2tfknZ6hs_rvqnAOxamAQRlV01g&sig2=sM22pjFrR_UfhYoPldnmQ&cad=rja /
27 Oct 11:49:29 - [5.game-api] - player_id 50upC1zsP (iwillpwn2) started working on private_id 1h5Sz-m-Wsi8
```

