

Workshop

Building Scalable Apps Featuring CockroachDB

MLH localhost



Welcome to MLH Localhost:

Building Scalable Apps featuring CockroachDB!



Wifi Network:
[Network]

Wifi Password:
[Password]



Event Hashtag:
#MLHLocalhost

Twitter Handle:
@MLHacks



[Put your Photo Here]

**Welcome! My name is
[INSERT YOUR NAME].**

1

I'm here to lead this session & help you learn something new today!

2

I'm a [LEVEL OF STUDY] at [SCHOOL].

3

My favorite programming language / tool is [LANGUAGE OR TOOL].

1

*Using your Web Browser,
Open this URL & Fill out the Form:*

<http://mlhlocal.host/checkin>

2

Afterwards, Check your Email to Find:

- Setup Instructions
- An Invite to the MLH Slack
- The Code Samples
- A Workshop FAQ
- These Workshop Slides
- More Learning Resources

What will you **learn** today?

- 1 What a database is & how the database you pick impacts your app's ability to scale.
- 2 How to build a scalable web application using CockroachDB.
- 3 How to test your application's ability to scale.

Why does this **matter**?

- 1 Successful apps need to grow without breaking/slowing. This is called **scaling**.
- 2 Scaling requires careful design of the stateful (data) part of your app.
- 3 Planning for scale helps you make better decisions earlier.
- 4 Employers want to hire engineers who can build for scale.



***Our Mission** is to Empower Hackers.*

65,000+
HACKERS

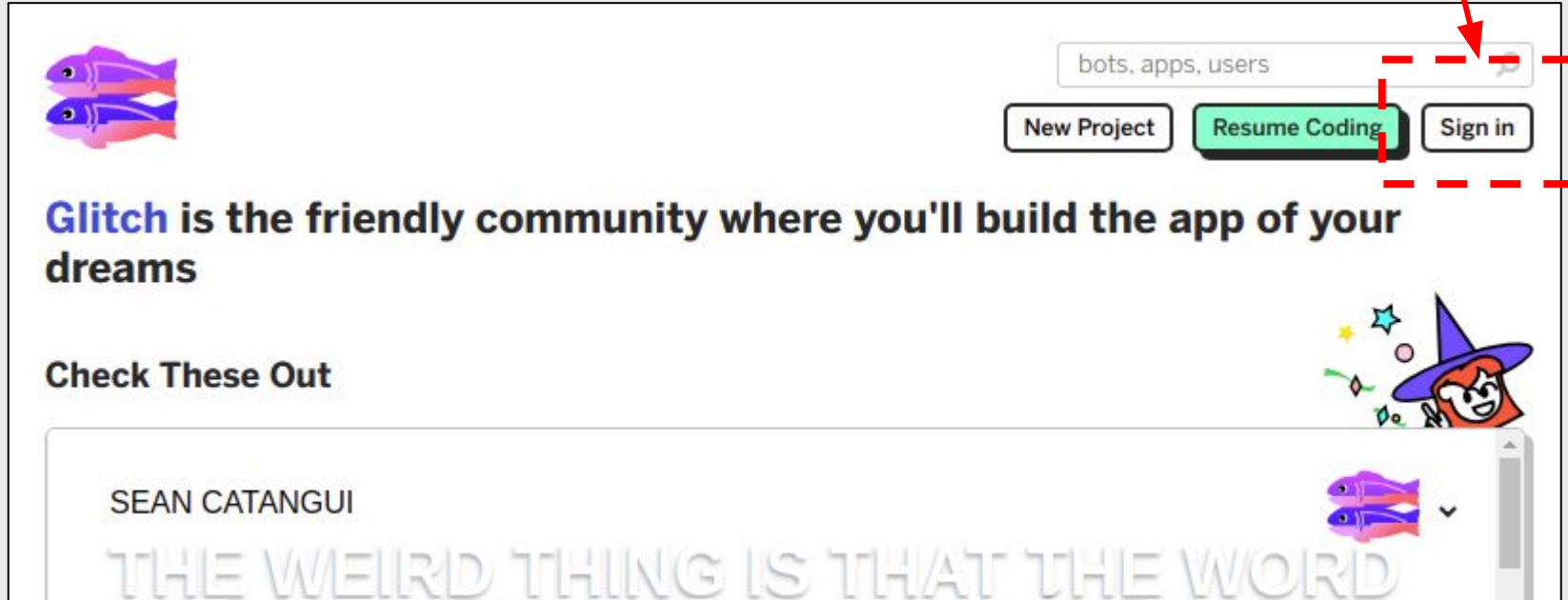
12,000+
PROJECTS CREATED

3,000+
SCHOOLS

We hope you learn something awesome today!
Find more resources: <http://mlh.io/>

Let's Set Up Glitch!

Sign in using GitHub or Facebook.

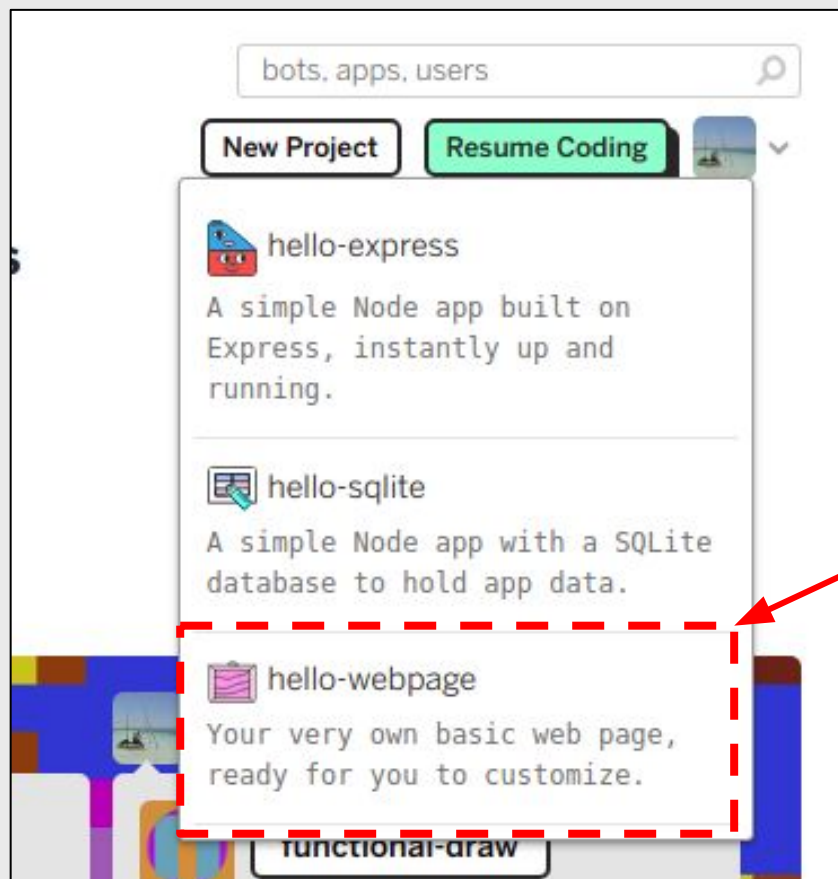


Note:

Making an account is not required, but your code will expire after 5 days if you don't.

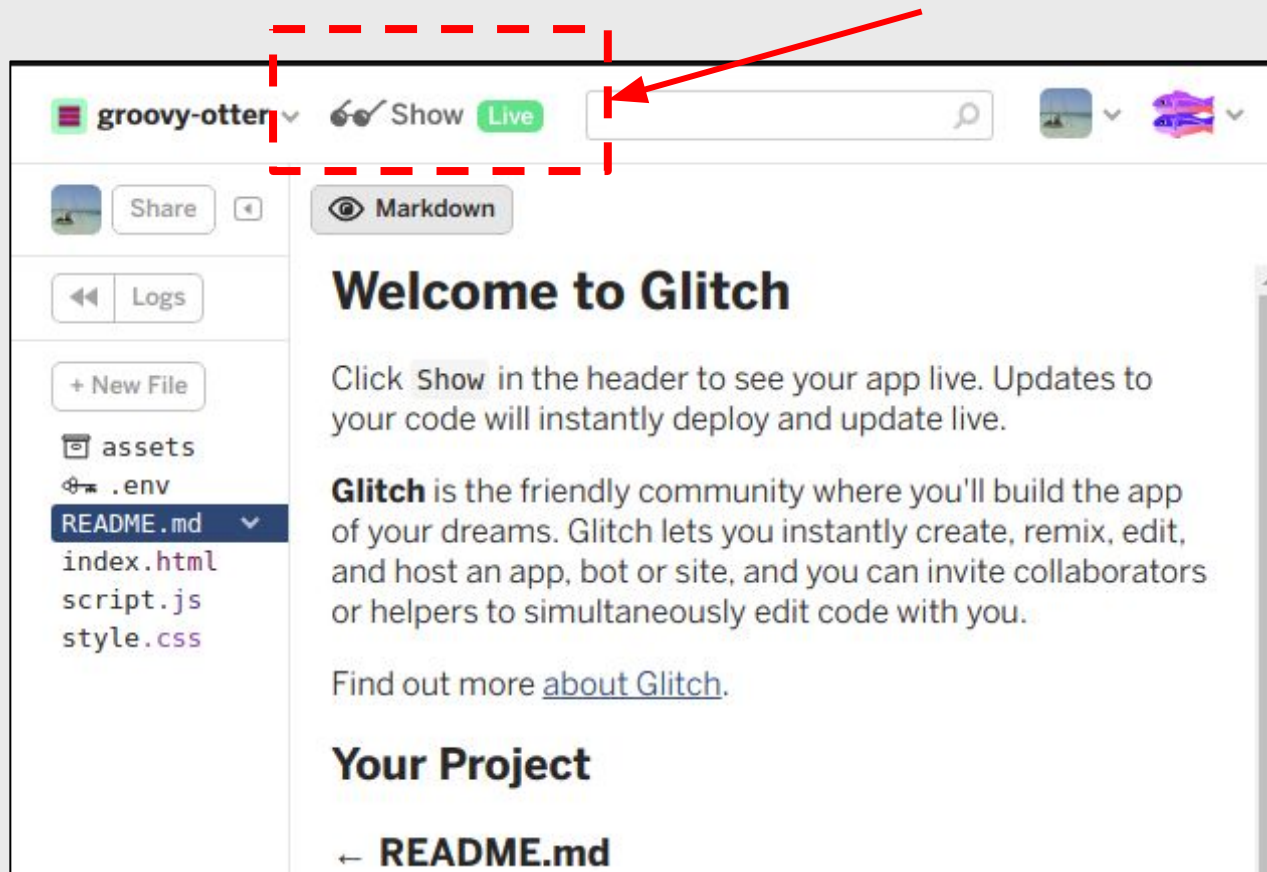
Let's Set Up Glitch!

Create a new web page project and run it.



Let's Set Up Glitch!

Show the output of the project



Let's Set Up Glitch!

You should see something like this, but with a different URL

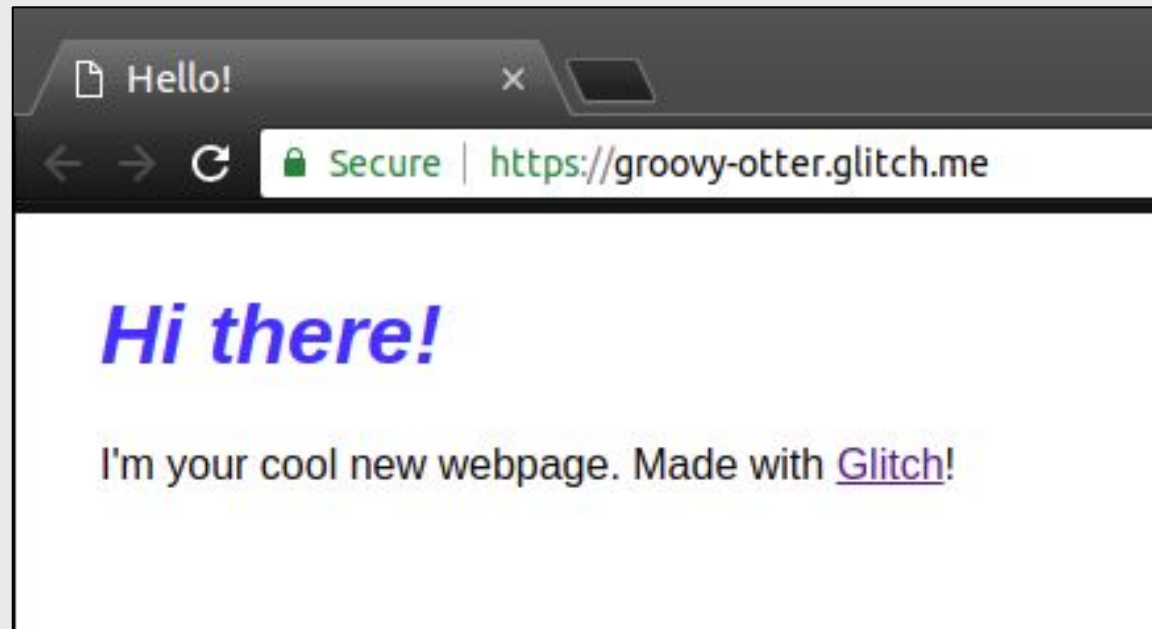



Table of Contents

-  **1.** Intro to Databases & CockroachDB
- 2.** Start your first CockroachDB Cluster
- 3.** Scale an App with CockroachDB
- 4.** Review & Quiz
- 5.** Next Steps

Quiz!

WHEN DO YOU NEED A DATABASE?

Case:

Local user settings on phone app

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Case:

Local user settings on phone app

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Case:

Music player phone app searchable collection browser

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Case:

Music player phone app searchable collection browser

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Case:

Online forum (Tumblr, Discourse)

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Case:

Online forum (Tumblr, Discourse)

- 1 Formatted text file (JSON, CSV)
- 2 Lightweight database (SQLite or NoSQL)
- 3 Database (MySQL, PostgreSQL, CockroachDB)

Database Basics

- 1 A database is an organized collection of data.
- 2 Data can be anything (numbers, text, files, etc.).
- 3 Databases allow web applications to have preserved state.



Database Management System (DBMS)

*A Database Management System (DBMS) is software that lets you **manage database structure and data**.*

Key Responsibilities





1. **Querying:**
Processing commands to interact with the data.
2. **Storage:**
Managing how the data is stored in the database.

Group Activity

Brainstorm:
List out as many Database Management Systems as we can in 2 minutes.

Hint:
CockroachDB is one example!

Four Types of Database Management Systems:

<u>Relational</u>	<u>NoSQL</u>	<u>Key/Value</u>	<u>NewSQL</u>
<p>→ Built for Consistency.</p> <p>→ Structured in tables.</p> <p>Example:</p> 	<p>→ Built for Scale (eventually).</p> <p>→ Unstructured.</p> <p>Example:</p> 	<p>→ Built for Speed.</p> <p>→ Unstructured</p> <p>Example:</p> 	<p>→ Scale & Consistency.</p> <p>→ Structured in Tables.</p> <p>Example:</p> 

Generally, you get ***either*** Consistency, Scale, or Speed.
NewSQL combines both Consistency & Scale.

What are Relational Databases?

- Data is structured into Tables (rows & columns).
- Every row in a Table has the same columns.
- Use “SQL” to query data.

ID	Hackathon	Date
1	hackNY	2017-04-08
2	Hack Princeton	2017-03-31
3	HackBCA	2017-03-25
4	PearlHacks	2017-02-11

```
/> SELECT * FROM my_database.hackathons WHERE Hackathon = "hackNY";
```

```

+----+-----+-----+
| id | Hackathon | Date       |
+----+-----+-----+
| 1  | hackNY    | 2017-04-08 |
+----+-----+-----+

```

What are NoSQL Databases?

- Data is unstructured and is stored in Collections.
- Every entry could have different columns.
- Use “JSON” to query data.

Sample Collection:

```
{
  "Hackathon": "hackNY",
  "Date": 2017-04-08
},
{
  "Hackathon": "Hack Princeton",
  "Date": 2017-04-08,
  "Custom-Data": "Some Custom Data."
}
```

```
/> db.hackathons.find({ Hackathon: "hackNY" })
```

```
{
  "Hackathon": "hackNY",
  "Date": 2017-04-08
}
```


Both Relational & NoSQL DBs have Trade Offs.

Relational Database Issues

1. Single-Server Databases are a point of failure.
2. Multi-Server Databases are complex (sharding).

NoSQL Database Issues

1. Eventual Consistency is hard to deal with.
2. Databases can be out of sync and lose data.

*Is there a way to get **both** scalability and consistency?
Yes, it's called NewSQL!*

Meet NewSQL & CockroachDB.

What is *NewSQL*?

- A new class of DBMS (started in 2011).
- Scales like NoSQL.
- Consistent like a Traditional Relational Database.



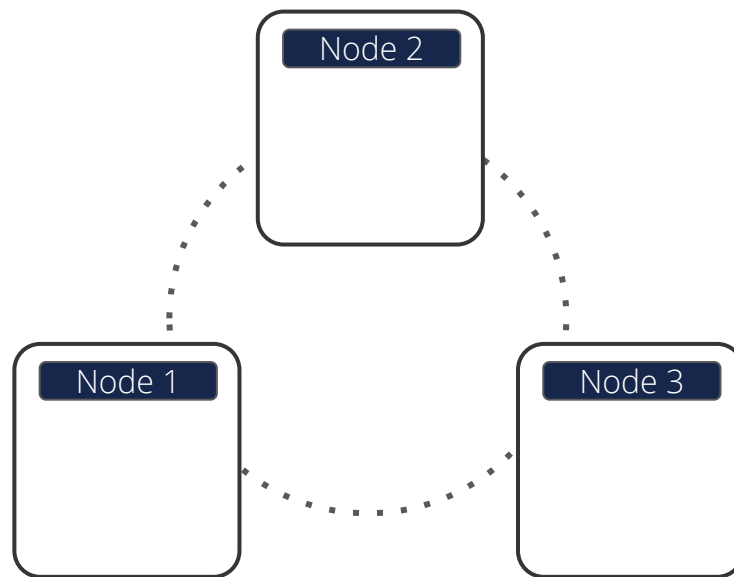
- Created by Cockroach Labs in 2014.
- Scalable SQL.
- Survivable.
- Consistent & Symmetric.

How does CockroachDB do this?

A CockroachDB deployment is

- Composed of many **Nodes** (single instances of the database software).
- Together they form a **cluster** (collection of nodes acting as a single database).

A CockroachDB Cluster:



How does CockroachDB do this?

Under the hood, CockroachDB...

- Splits your data into **ranges**, which are distributed across nodes in your cluster.
- **Replicates** data across many nodes in your cluster in case one goes down.
- Uses a peer-to-peer Gossip Protocol to organize Nodes and the RAFT Consensus Algorithm for consistency.

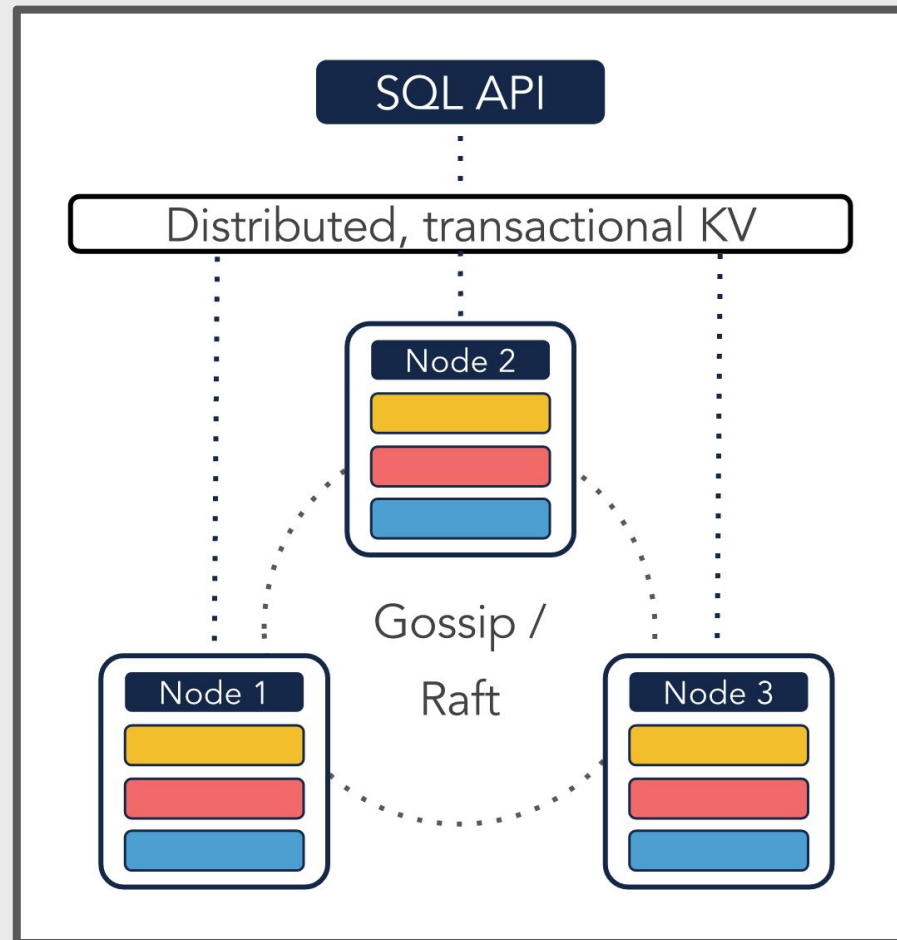



Table of Contents

1. Intro to Databases & CockroachDB
-  2. Start your first CockroachDB Cluster
3. Scale an App with CockroachDB
4. Review & Quiz
5. Next Steps

Try the Demo Application:

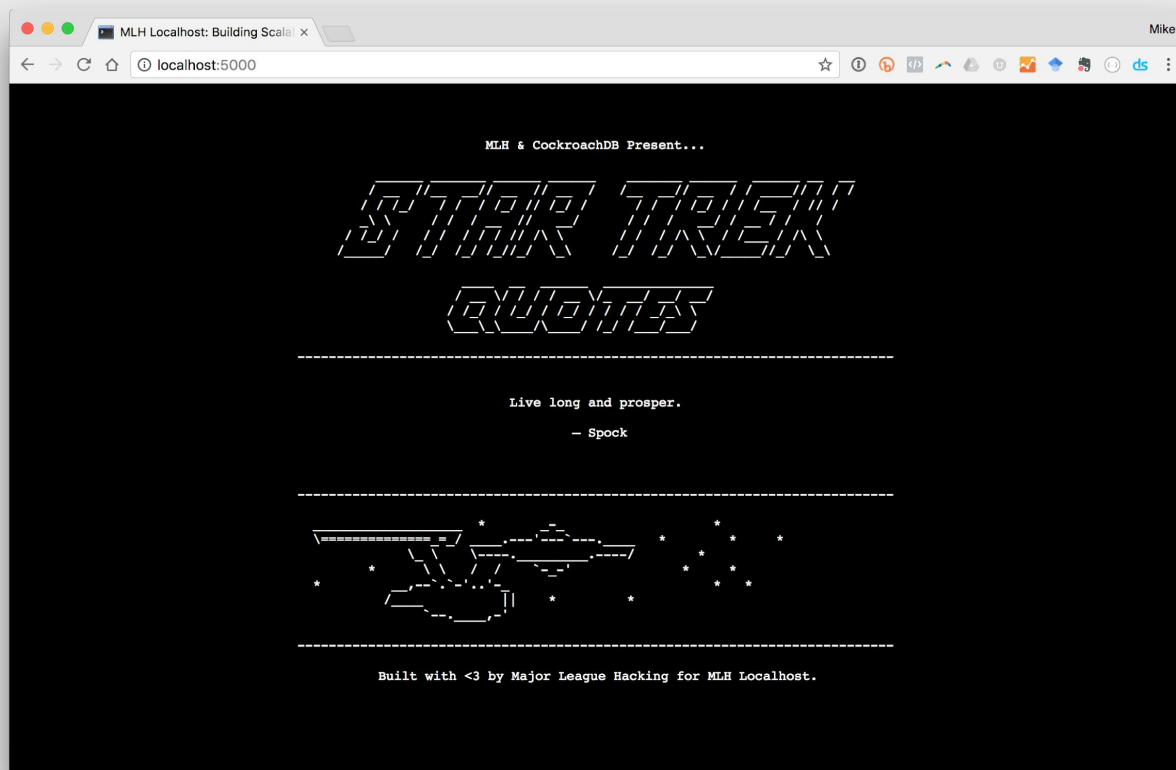
<http://mlhlocal.host/cockroachdb-demo>

Goal:

Display a random
Star Trek quote
every 5 seconds.

Technologies:

Node.js
JavaScript
HTML / CSS
jQuery

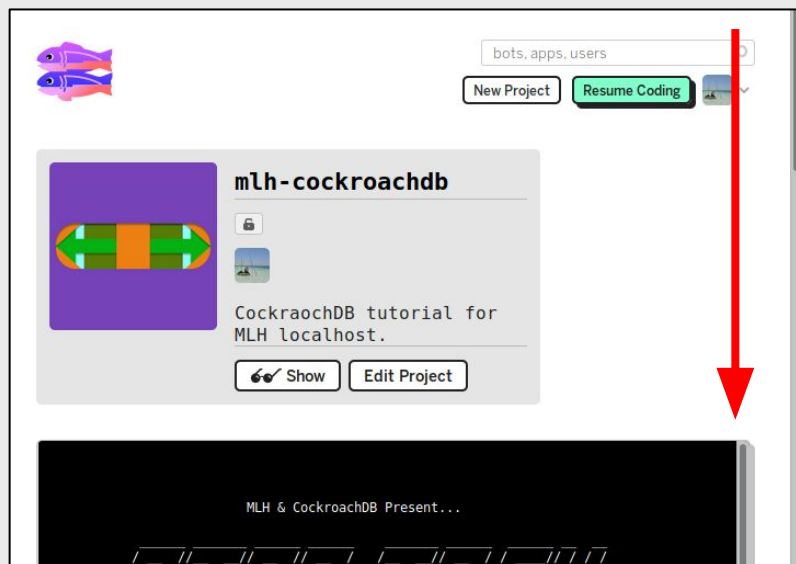


Make a copy of the demo

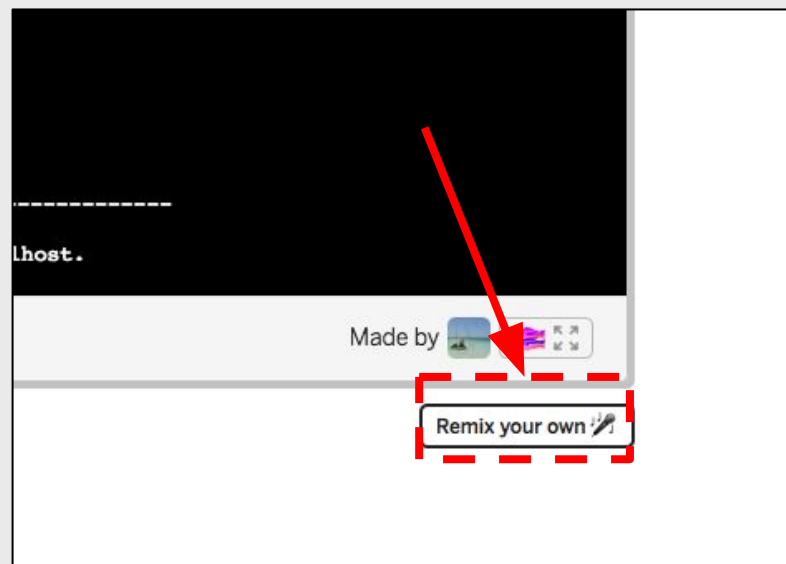
Remix the finished project at

<http://mlhlocal.host/cockroachdb-glitch>

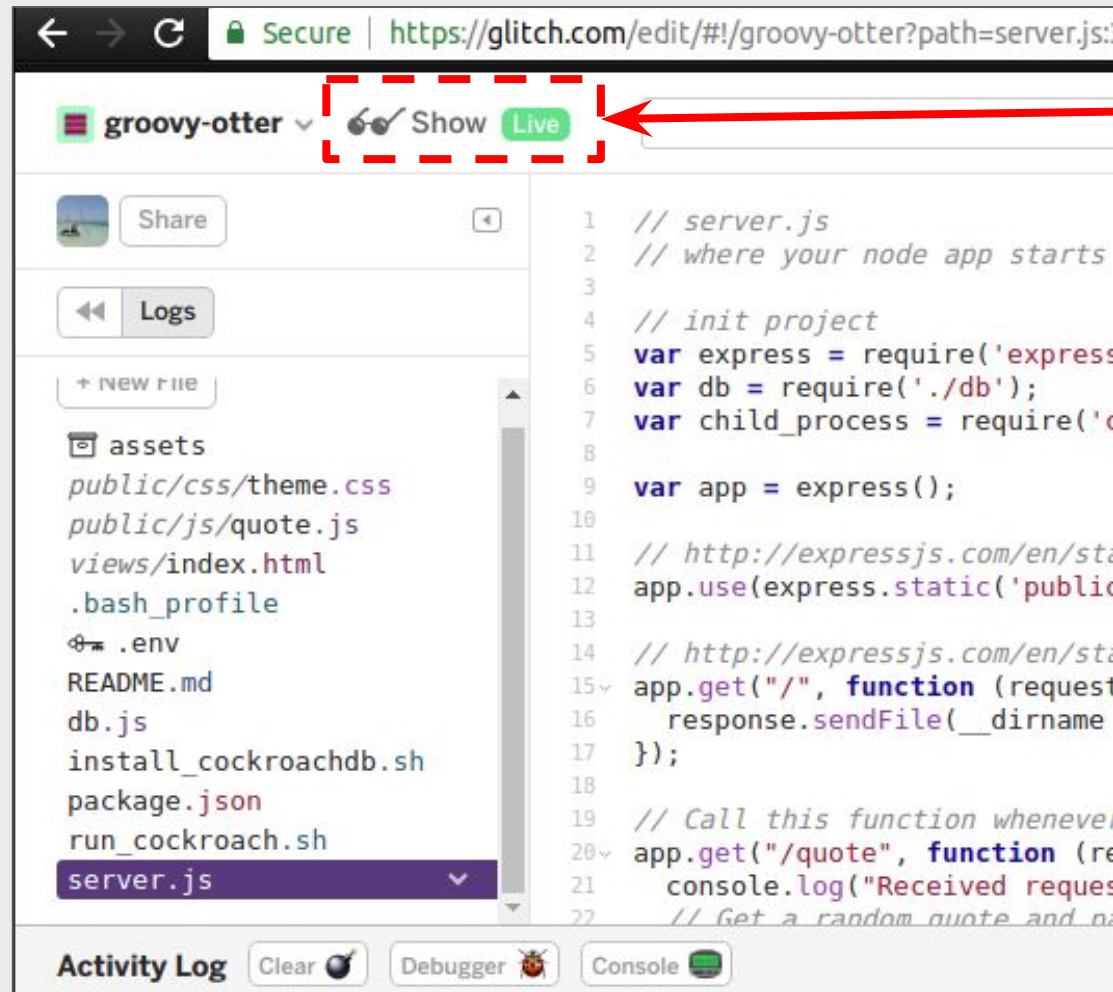
Scroll down past the preview.



Click **Remix your own**

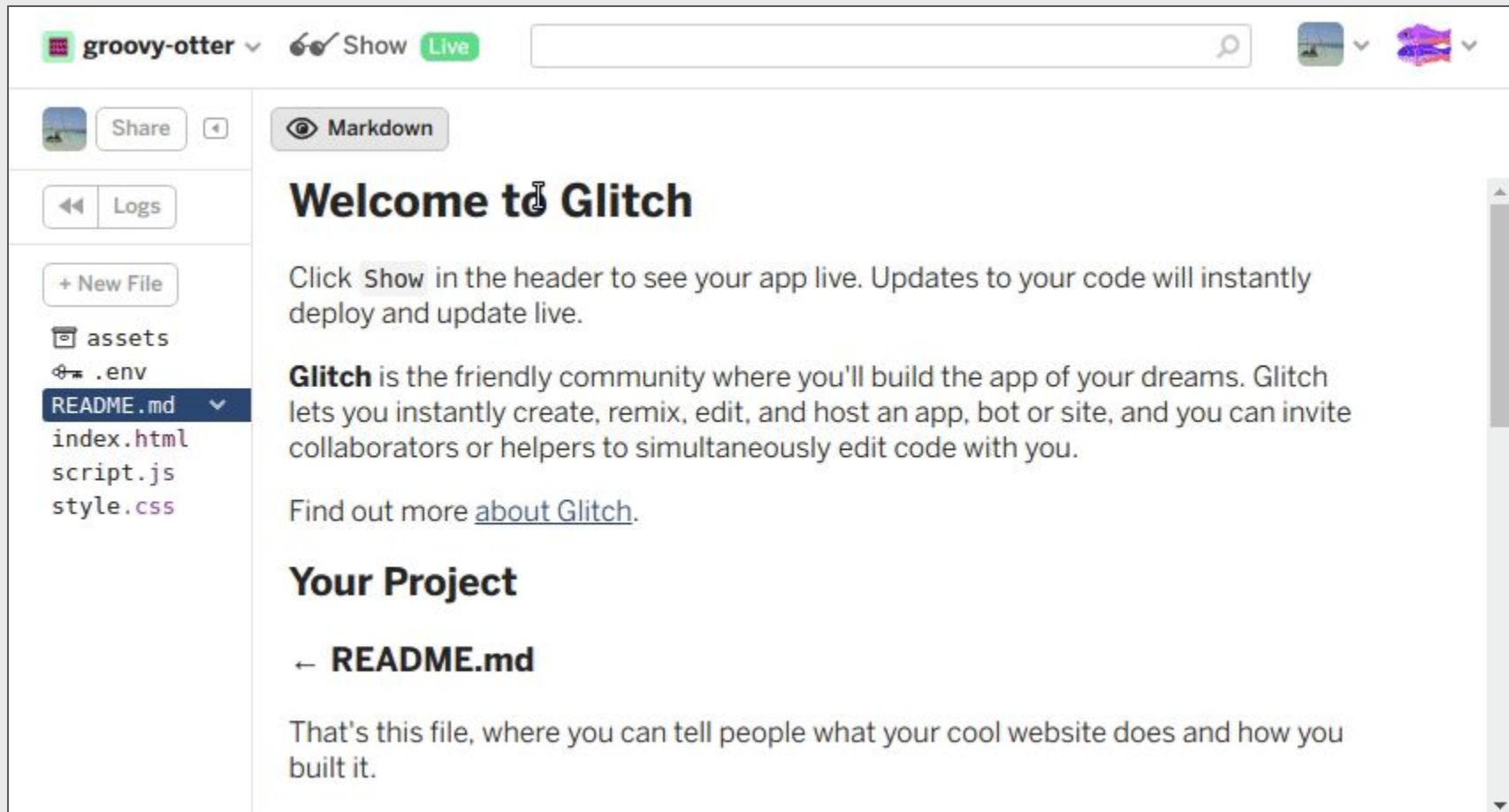


Run your copy of the demo



Open the Project Console

Click Logs -> Console



This will open the console in a new tab.

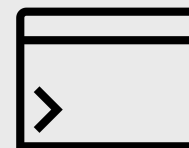
File Browser vs. Console

Different views of the same thing



File Browser

- Click things to select files and navigate
- Type things into the files



Console

- Type commands to perform actions and navigate
- Needs to be synced with File Browser with **refresh** command

Keep both these tabs open at all times.

Install CockroachDB

Run a series of commands for installation saved inside **install_cockroachdb.sh** using the **sh** command

```
$ source install_cockroachdb.sh
```

```
Installing CockroachDB
```

```
--2018-06-28 19:00:03--
```

```
https://binaries.cockroachdb.com/cockroach  
-latest.linux-amd64.tgz
```

```
...
```

Make sure CockroachDB is working.

To test that your installation was successful, type the command below in your Terminal.

```
$ cockroach version
```

```
Build Tag:      v2.0.3
Build Time:     2018/06/18 16:11:33
Distribution:   CCL
Platform:      linux amd64 (x86_64-unknown-linux-gnu)
Go Version:     go1.10
C Compiler:     gcc 6.3.0
Build SHA-1:    91715a9a95edbe716912173204fa4c0fc6724457
Build Type:     release
```

Start your first instance of CockroachDB!

To launch your first Cluster with one Node, run the following command in the terminal. **DON'T CLOSE THE TERMINAL FROM NOW ON!**

```
$ cd ~/.data
$ cockroach start --insecure --host=localhost --background

* WARNING: RUNNING IN INSECURE MODE!
*
* - Your cluster is open for any client that can access localhost.
* - Any user, even root, can log in without providing a password.
* - Any user, connecting as root, can read or write any data in your cluster.
* - There is no network encryption nor authentication, and thus no confidentiality.
*
* Check out how to secure your cluster:
https://www.cockroachlabs.com/docs/stable/secure-a-cluster.html
*

CockroachDB node starting at 2017-11-02 18:06:21.541283 +0000 UTC (took 0.4s)
build:      CCL v1.1.2 @ 2017/11/02 20:52:23 (go1.9.2)
admin:      http://localhost:8080
sql:
postgresql://root@localhost:26257?application_name=cockroach&sslmode=disable
logs:       /Users/majorleaguehacking/cockroach-data/logs
```

Let's create the sample data.

Run the following command to create a database named `startrek` which contains `quotes` & `episode` tables.

```
$ cockroach gen example-data startrek | cockroach sql --insecure
```

```
CREATE DATABASE
Time: 505.928737ms
SET
Time: 1.582383ms
DROP TABLE
Time: 2.200558ms
CREATE TABLE
Time: 538.467864ms
INSERT 1
Time: 968.790375ms
INSERT 1
Time: 220.132547ms
INSERT 1
Time: 792.940707ms
...
```

Use the Command Line to see your Data.

When you run the command bellow, CockroachDB will place you in an program that responds to SQL Queries.

```
$ cockroach sql --insecure
```

```
# Welcome to the cockroach SQL interface.  
# All statements must be terminated by a semicolon.  
# To exit: CTRL + D.  
root@:26257/>
```

List the available databases.

You can see a list of available databases using the `SHOW DATABASES;` Command.

```
root@:26257/> SHOW DATABASES;
```

Database
crdb_internal
information_schema
pg_catalog
startrek
system

```
(5 rows)
```

```
root@:26257/>
```


What's inside the startrek database?

The Command we ran earlier created the **startrek** database. See what's inside with `SHOW TABLES`.

```
root@:26257/> SHOW TABLES FROM startrek;
```

```
+-----+
```

```
| Table |
```

```
+-----+
```

```
| episodes |
```

```
| quotes |
```

```
+-----+
```

```
(2 rows)
```

```
root@:26257/>
```

What's inside the startrek database?

The Command we ran earlier created the **startrek** database. See what's inside with **SHOW TABLES**.

```
root@:26257/> SELECT * FROM startrek.episodes;
```

id	season	num	title	stardate
1	1	1	The Man Trap	1531.1
2	1	2	Charlie X	1533.6
3	1	3	Where No Man Has Gone Before	1312.4
4	1	4	The Naked Time	1704.2
5	1	5	The Enemy Within	1672.1
6	1	6	Mudd's Women	1329.8
7	1	7	What Are Little Girls Made Of?	2712.4
8	1	8	Miri	2713.5
9	1	9	Dagger of the Mind	2715.1
10	1	10	The Corbomite Maneuver	1512.2

...


(79 rows)

```
root@:26257/>
```


How would we get a random startrek quote?

This is what you would write in SQL:


```
SELECT * FROM startrek.quotes ORDER BY RANDOM() LIMIT 1;
```




Read all
Columns



From the **quotes** Table
in the **startrek** Database



Order the results
randomly



Limit the
results to one

Get a random startrek quote.

Let's get a random quote from the startrek `quotes` table using the query we just wrote.

```
root@:26257/> SELECT * FROM startrek.quotes ORDER BY RANDOM() LIMIT 1;
```

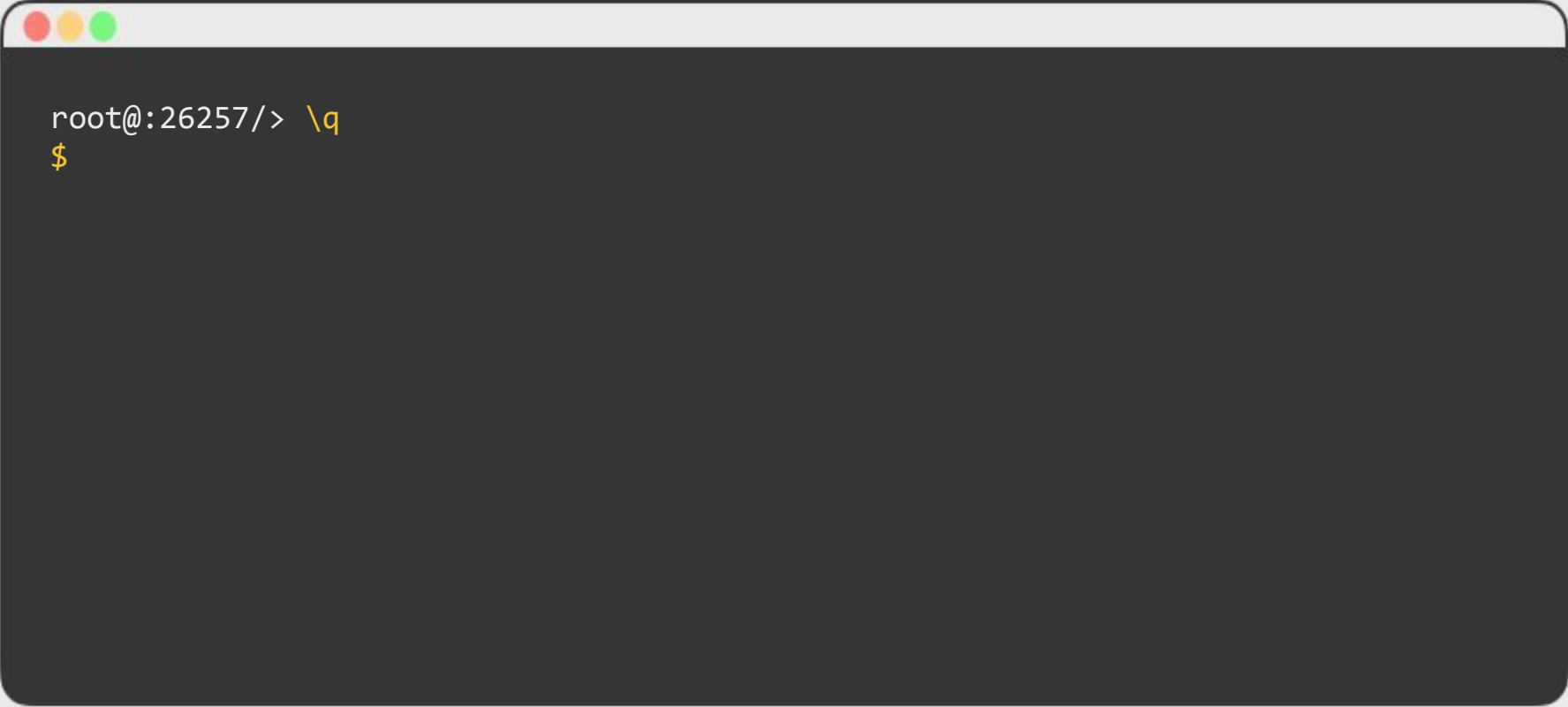
quote	characters	stardate	episode
You're dead, Jim.	McCoy	3372.7	30

```
(1 rows)
```

```
root@:26257/>
```

How do you exit the SQL shell?

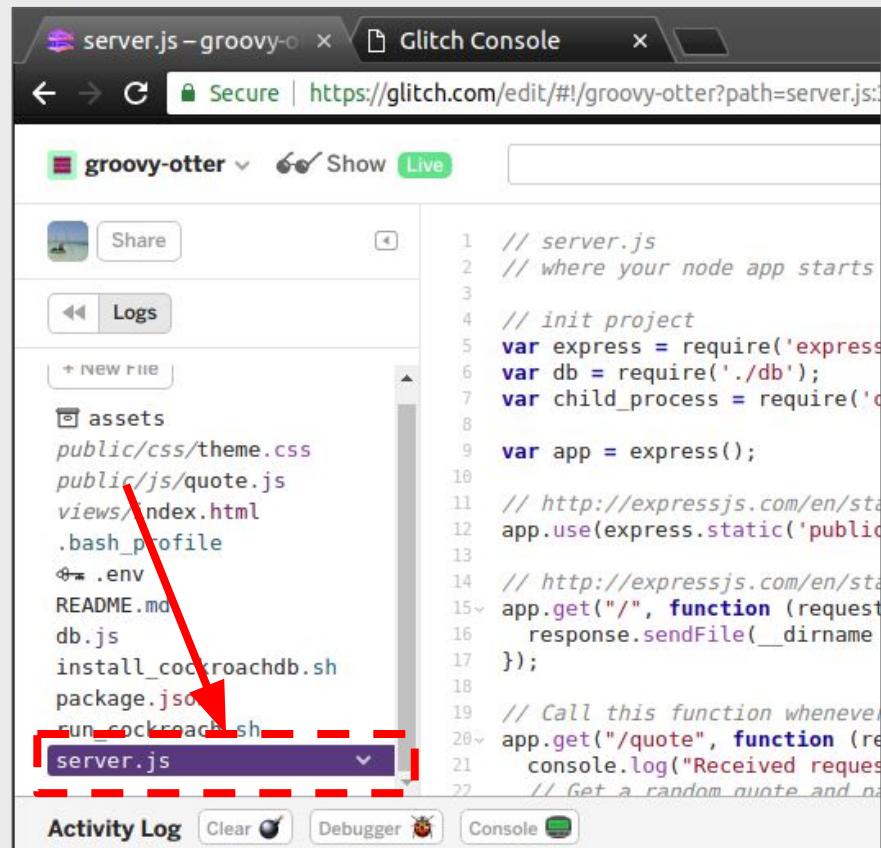
We're done with the SQL shell. Quit and return to your normal terminal by using `\q`.



```
root@:26257/> \q
$
```

Let's explore the application code!

In the file tree on the left side of the file browser tab, click **server.js**.



What does this application do?

Lines 5 - 17 and 31 - 37 cover the server boilerplate code.

All this app does is serve static files (like HTML).

server.js

```
// load libraries
var express = require('express');
var db = require('./db');
var child_process = require('child_process')

// Use Express to serve files from the /public directory
// http://expressjs.com/en/starter/static-files.html
var app = express();
app.use(express.static('public'));

// http://expressjs.com/en/starter/basic-routing.html
app.get("/", function (request, response) {
  response.sendFile(__dirname + '/views/index.html');
});
```

How do we return quotes?

Whenever someone visits the demo website, we run the following code to return a random quote.

server.js

```
11 // Call this function whenever someone requests the /quote path
12 app.get('/quote', function(req, res) {
13   // Get a random quote and pass it to the browser.
14   client.query('SELECT * FROM quotes ORDER BY RANDOM() LIMIT 1')
15     .then(data => res.send(data.rows[0]))
16     .catch(err => res.send({ error: "BRB! Connecting to CockroachDB" }))
17 })
```


Next, let's see how we connect to CockroachDB in db.js

Open `db.js`

`db.js`

```
01 // Connect to the startrek Database on a node in our CockroachDB cluster
02 var pg = require('pg');
03 var config = { user: 'root', database: 'startrek', port: 26257 };
04 var db = new pg.Pool(config);
05
06 // Log when we connect to the CockroachDB node
07 db.on('connect', function(client) { console.log("Connected to CockroachDB.") })
08
09 // Log any errors we encounter
10 db.on('error', function(e) { console.error("Err connecting to CockroachDB.") })
11
12 // Export the database connection so anyone can use it
13 module.exports = db;
```

Run the application.

Click the **Show** button in your Glitch file browser tab as before.

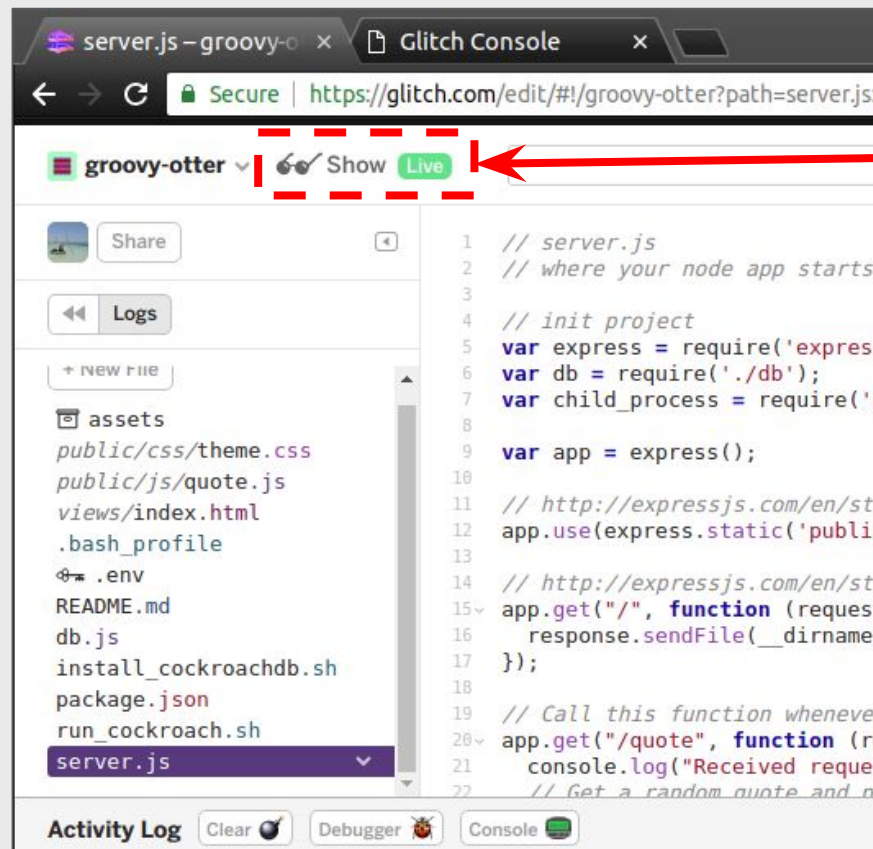



Table of Contents

1. Intro to Databases & CockroachDB
2. Start your first CockroachDB Cluster
-  3. Scale an App with CockroachDB
4. Review & Quiz
5. Next Steps

Let's add a 2nd Node to our Cluster.

To add another node, we need to tell it which cluster to join and specify an open set of ports.

```
$ cockroach start --insecure --host=localhost --background --join=localhost:26257
--store=mlh-node2 --port=26258 --http-port=8081

*
* WARNING: RUNNING IN INSECURE MODE!
// More Warnings
*
CockroachDB node starting at 2017-11-02 18:26:28.771325 +0000 UTC (took 1.2s)
build:      CCL v1.1.0 @ 2017/10/18 08:14:16 (go1.9.1)
admin:      http://localhost:8081
sql:
postgres://root@localhost:26258?application_name=cockroach&sslmode=disable
logs:
/Users/majorleaguehacking/Downloads/mlh-localhost-cockroachdb-master/mlh-node2/logs
...
```

And a 3rd Node.

Just change “node2” to “node3” and increment the ports by one. Make sure the cluster stays the same though.

```
$ cockroach start --insecure --host=localhost --background --join=localhost:26257
--store=mlh-node3 --port=26259 --http-port=8082

*
* WARNING: RUNNING IN INSECURE MODE!
*
// More Warnings

CockroachDB node starting at 2017-11-02 18:28:15.597333 +0000 UTC (took 1.2s)
build:      CCL v1.1.0 @ 2017/10/18 08:14:16 (go1.9.1)
admin:      http://localhost:8082
sql:
postgresql://root@localhost:26259?application_name=cockroach&sslmode=disable
logs:
/Users/majorleaguehacking/Downloads/mlh-localhost-cockroachdb-master/mlh-node3/logs
...
```

You should see 3 Nodes listed.

```
$ cockroach node ls --insecure
```

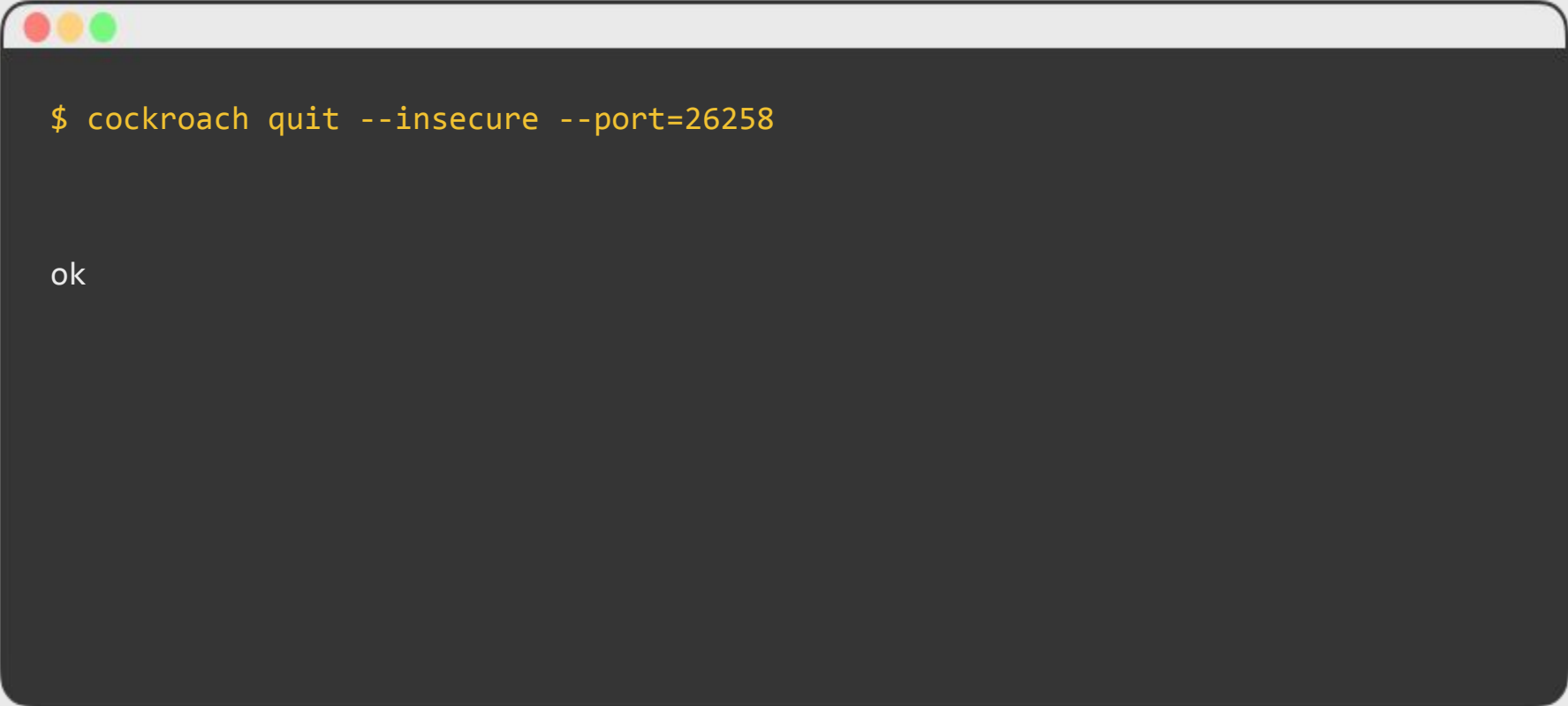
```
+-----+
| id |
+-----+
| 1 |
| 2 |
| 3 |
+-----+
(3 rows)
```

```
$ cockroach node status --insecure
```

```
+-----+-----+-----+-----+-----+-----+
| id | address | build | updated_at | started_at | is_live |
+-----+-----+-----+-----+-----+-----+
| 1 | localhost:26257 | v2.0.3 | 2018-06-19 16:39:33 | 2018-06-19 16:31:13 | true |
| 2 | localhost:26258 | v2.0.3 | 2018-06-19 16:39:33 | 2018-06-19 16:35:43 | true |
| 3 | localhost:26259 | v2.0.3 | 2018-06-19 16:39:31 | 2018-06-19 16:37:01 | true |
+-----+-----+-----+-----+-----+-----+
```

What about Fault Tolerance?

Let's take one of our nodes offline, change some data while it's down, and see what happens!



```
$ cockroach quit --insecure --port=26258
```

```
ok
```

What do you see when the Node is down?

Notice anything?

```
$ cockroach node status --insecure
```

id	address	build	updated_at	started_at	is_live
1	localhost:26257	v2.0.3	2018-06-19 16:52:23	2018-06-19 16:31:13	true
2	localhost:26258	v2.0.3	2018-06-19 16:49:33	2018-06-19 16:35:43	false
3	localhost:26259	v2.0.3	2018-06-19 16:52:21	2018-06-19 16:37:01	true

What do you see when the Node is down?

The previous command took down Node 2.

```
$ cockroach node status --insecure
```

id	address	build	updated_at	started_at	is_live
1	localhost:26257	v2.0.3	2018-06-19 16:52:23	2018-06-19 16:31:13	true
2	localhost:26258	v2.0.3	2018-06-19 16:49:33	2018-06-19 16:35:43	false
3	localhost:26259	v2.0.3	2018-06-19 16:52:21	2018-06-19 16:37:01	true

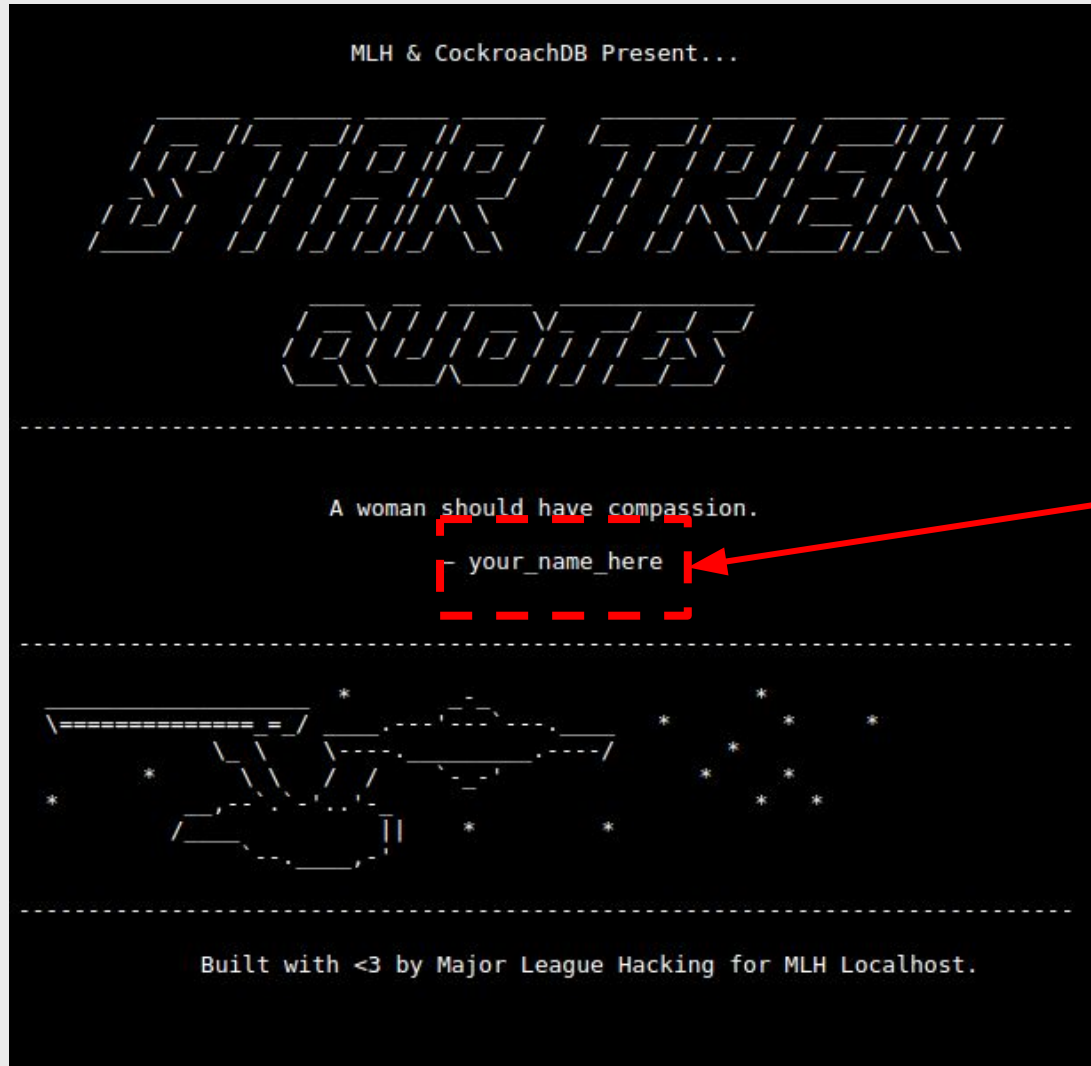
Let's write some data to your Cluster.

This time, let's write to Node #3 by specifying a port.
In the example, we're going to change some data.

```
$ cockroach sql --insecure --port=26259
```

```
# Welcome to the cockroach SQL interface.
# All statements must be terminated by a semicolon.
# To exit: CTRL + D.
#
# Server version: CockroachDB CCL v1.1.0 (darwin amd64, built 2017/10/18 08:14:16,
go1.9.1) (same version as client)
# Cluster ID: 41b2e624-ce49-40fa-8ef3-e573f4dda3bb
#
# Enter \? for a brief introduction.
#
root@:26259/> SET sql_safe_updates = false;
root@:26259/> UPDATE startrek.quotes SET characters = 'your_name_here';
UPDATE 200
root@:26259/> \q
```

To see if it worked, refresh your App.



Let's bring our Node back online.

We can just reuse the same command we used to start the node in the first place.

```
$ cockroach start --insecure --host=localhost --background --join=localhost:26257  
--store=mlh-node2 --port=26258 --http-port=8081
```

```
*
```

```
* WARNING: RUNNING IN INSECURE MODE!
```

```
*
```

```
// More Warnings
```

```
CockroachDB node starting at 2017-11-02 18:42:09.9441 +0000 UTC (took 2.2s)
```

```
build:      CCL v1.1.0 @ 2017/10/18 08:14:16 (go1.9.1)
```

```
admin:      http://localhost:8081
```

```
sql:
```

```
postgres://root@localhost:26258?application_name=cockroach&sslmode=disable
```

```
...
```

Let's connect & see what data it has...

Connect to the SQL console on the Node you just brought back online. What data does it have?

```
$ cockroach sql --insecure --port=26258
```

```
# Welcome to the cockroach SQL interface.
# All statements must be terminated by a semicolon.
# To exit: CTRL + D.
```

```
root@:26258/> SELECT * FROM startrek.quotes ORDER BY RANDOM() LIMIT 1;
```

quote	characters	stardate	episode
We have phasers, I vote we blast 'em!	your_name_here	1514.2	10

```
root@:26258/> \q
```

Let's connect & see what data it has...

Pay close attention below - we have changed the port from 26258 to 26257.

```
$ cockroach sql --insecure --port=26257
```


```
# Welcome to the cockroach SQL interface.
# All statements must be terminated by a semicolon.
# To exit: CTRL + D.
```

```
root@:26258/> SELECT * FROM startrek.quotes ORDER BY RANDOM() LIMIT 1;
```

quote	characters	stardate	episode
We have phasers, I vote we blast 'em!	your_name_here	1514.2	10

```
root@:26258/> \q
```

Table of Contents

1. Intro to Databases & CockroachDB
2. Start your first CockroachDB Cluster
3. Scale an App with CockroachDB
-  4. Review & Quiz
5. Next Steps



Let's recap quickly...


- 1 It was simple to connect CockroachDB to our application using an existing library.
- 2 CockroachDB **split & replicated** our data across multiple Nodes in our Cluster.
- 3 When a Node came back online, it worked with the other Nodes to **catch up** on what it missed.

What did you learn today?

We created a fun quiz to test your knowledge and see what you learned from this workshop.

<http://mlhlocal.host/quiz>

Table of Contents

1. Intro to Databases & CockroachDB
2. Start your first CockroachDB Cluster
3. Scale an App with CockroachDB
4. Review & Quiz
-  5. Next Steps



Join the CockroachDB community today:

- 1** Read the Documentation:
<http://mlhlocal.host/cockroach-docs>
- 2** Post Questions on StackOverflow:
<http://mlhlocal.host/cockroach-stackoverflow>
- 3** Use with CockroachDB in your favorite language:
<http://mlhlocal.host/cockroach-build>

Keep Learning: Practice Problems for later.

#1: Advanced SQL

Challenge: Display the episode name next to the quote in your application.

Instructions: Perform a JOIN between the quotes and episodes tables to get the episode title.

#2: Deploy on AWS

Challenge: Deploy your application on AWS and use HAProxy to load balance.

Instructions: Make your application production ready by deploying on 4 AWS micro instances with HAProxy.

Learning shouldn't stop when the workshop ends...

Check your email for access to:



- These workshop slides
- Practice problems to keep learning
- Deeper dives into key topics
- Instructions to join the community
- More opportunities from MLH!



Sign up for the MLH Career Lab!

<http://mlhlocal.host/career-lab>

- Browse a curated list of hacker jobs.
- Apply for jobs and internships from companies that want to recruit directly from the MLH community.
- Receive updates and career advice from MLH!



Workshop

Building Scalable Apps Featuring CockroachDB

MLH localhost

