

bork bork!

Final Project ISFWS



Birth Rate Down, “Happy Dog” Rate Up

Professor: Alexander Yohan, Teacher Assistant: Jeremy Hartono

Mi5117701 Ford Iammongkol
M10815809 Eric Dao / 姚昭宇

1

Intro

Dogs in Taiwan











Meet the Team



bork! bork!





About bork bork

"bork bork" was made to both make enjoyment and realization of the many instances of dogs being treated like little children in Taiwan. It is both incredibly funny and entertaining. However, at the same time, a possible warning of a low birth-rate population choosing to raise dogs rather than children.

2

Our Setup



Our Setup

Operating system		Windows 10
Front-end		html, css, js Bootstrap 4
Back-end		Nodejs v14.15.0 express, Passport, jsonwebtoken
Database		Mysql v8.0
Hosting		Hosting on an NTUST lab ip. A proxy heroku url is used to redirect to the lab ip.





Our Services

1. View the current posts



2. Create your own post



3. View Instagram posts



System Architecture

Web Service

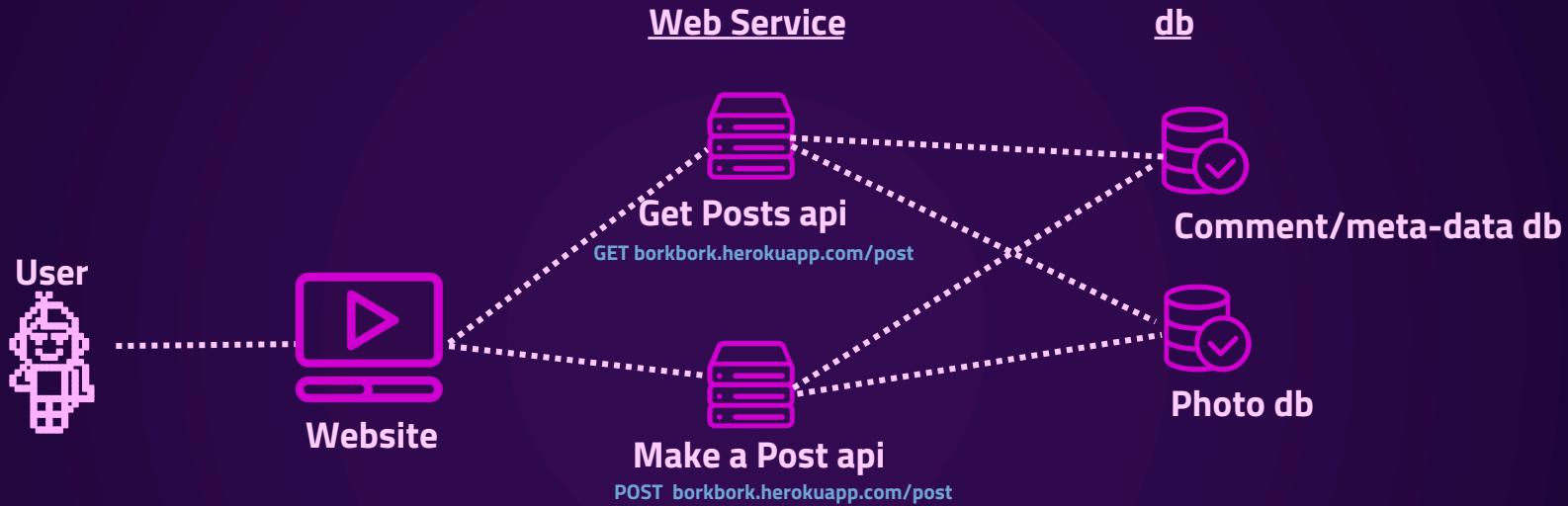
db



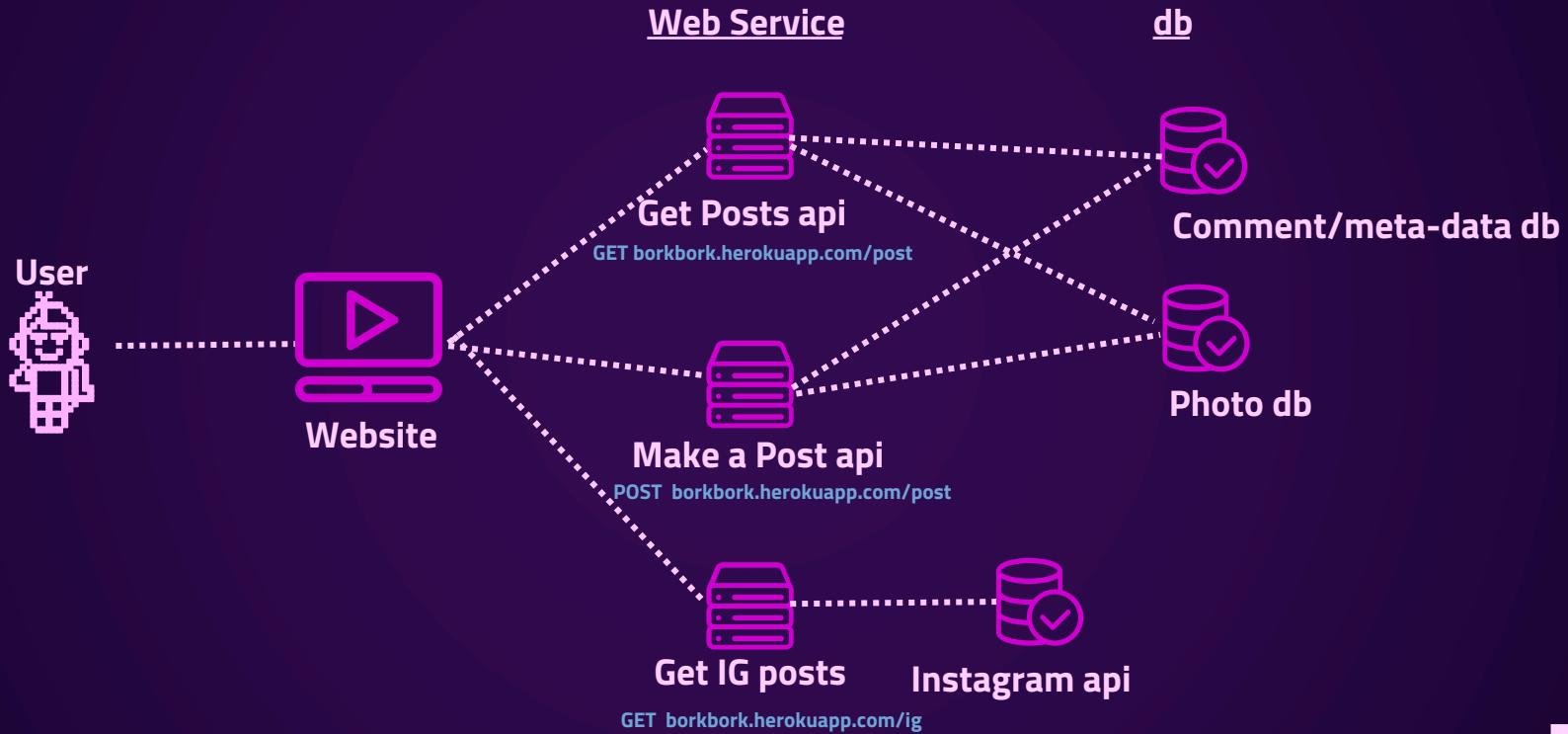
System Architecture



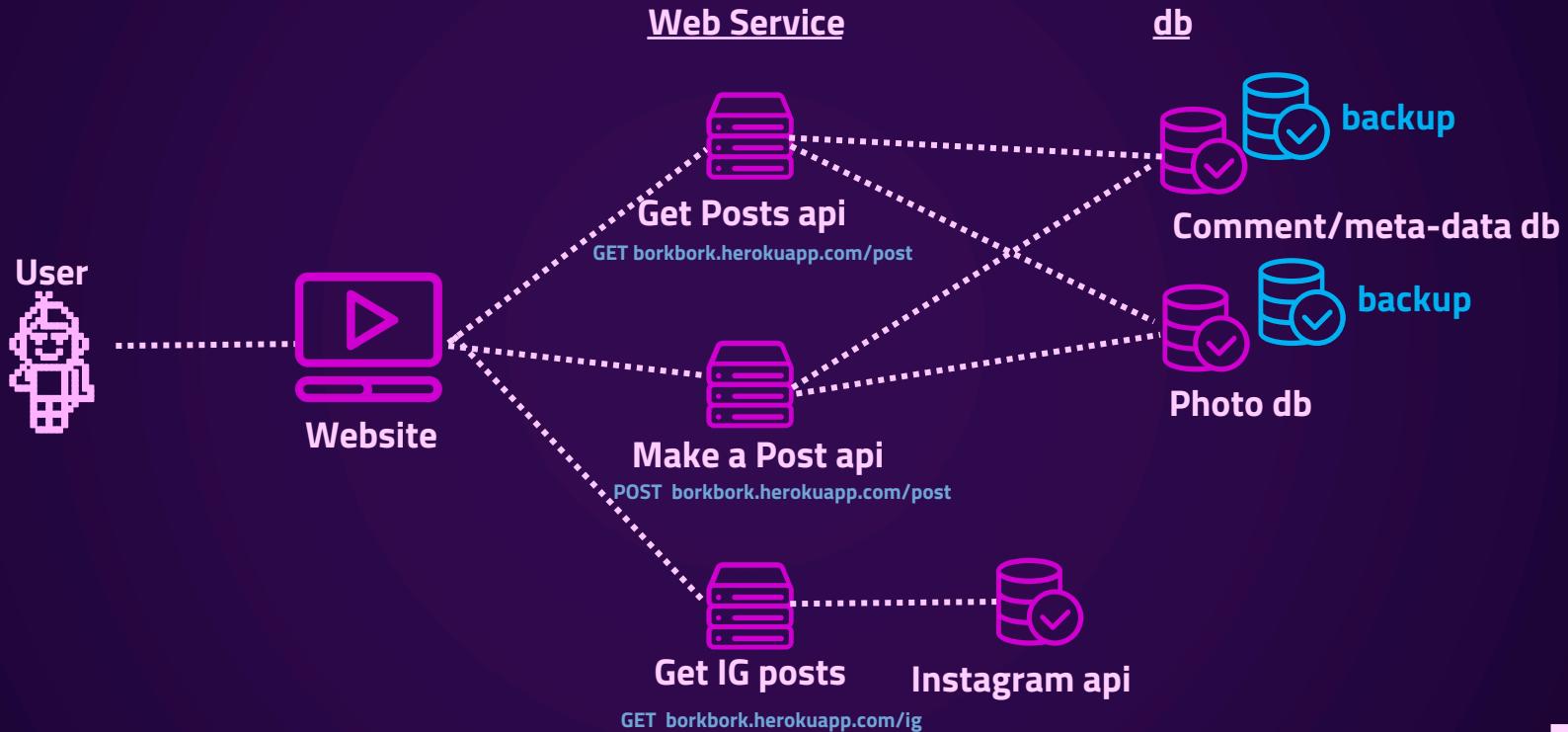
System Architecture



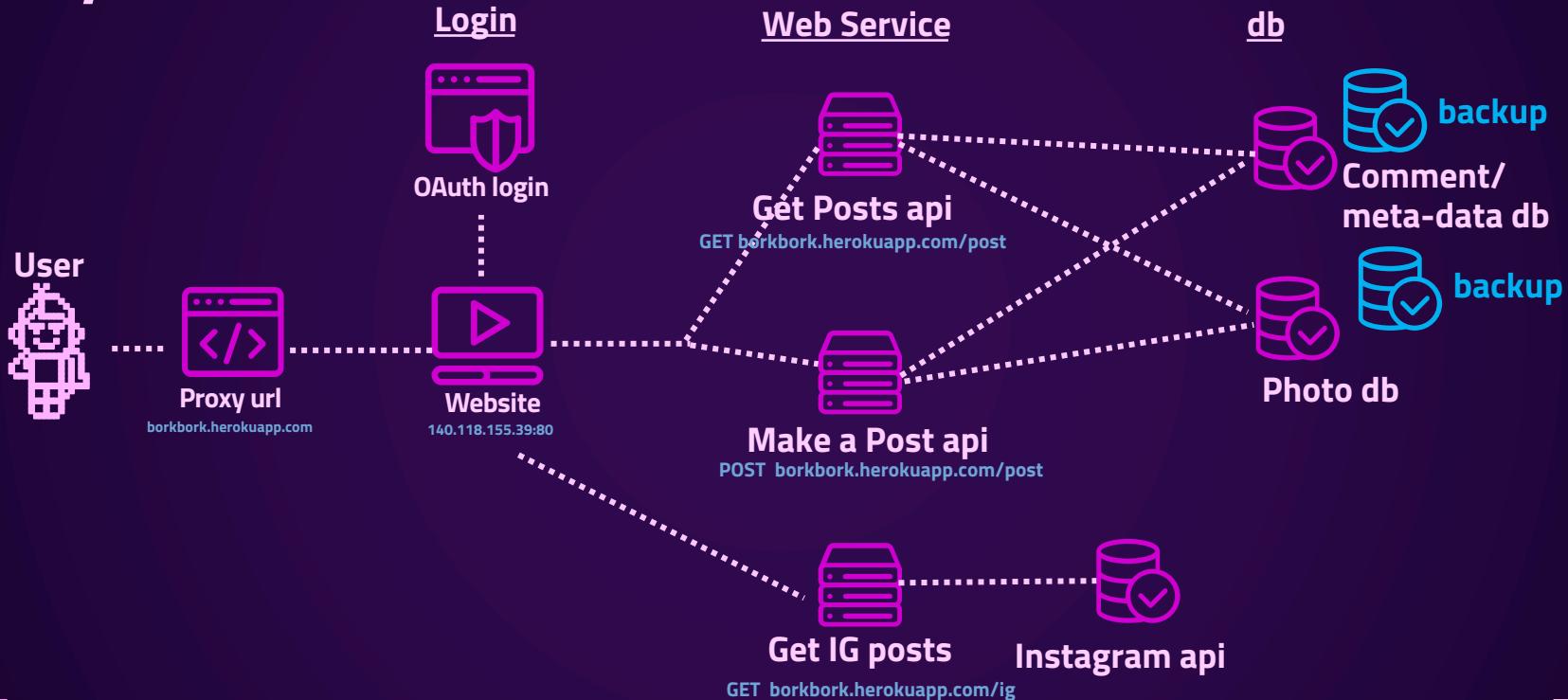
System Architecture



System Architecture



System Architecture



Databases

Comment/meta-data db

```
mysql> desc post;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| post_id | bigint | NO  |  | NULL    |          |
| post_text | varchar(500) | YES |  | NULL    |          |
| poster_id | bigint | YES |  | NULL    |          |
| post_time | datetime | YES |  | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Photo db

s PC > Local Disk (D:) > node_js > scripts > public > images			
Name	Date modified	Type	
img_id_1608621364306.jpeg	12/22/2020 3:16 PM	JPEG File	
img_id_1608632072664.jpeg	12/22/2020 6:14 PM	JPEG File	
img_id_1608632279086.jpeg	12/22/2020 6:17 PM	JPEG File	
img_id_1608632330735.jpeg	12/22/2020 6:18 PM	JPEG File	
img_id_1608632330735.jpg	12/22/2020 6:18 PM	JPEG File	

Accepted formats

.jpg
.jpeg
.png
.gif

3

Demo Website

bork bork!



4

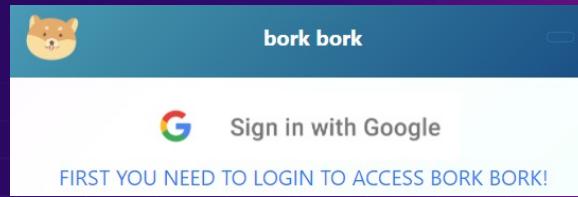
Web Service Security



Restricting access using OAuth



Restricting access using OAuth



Restricting access using OAuth

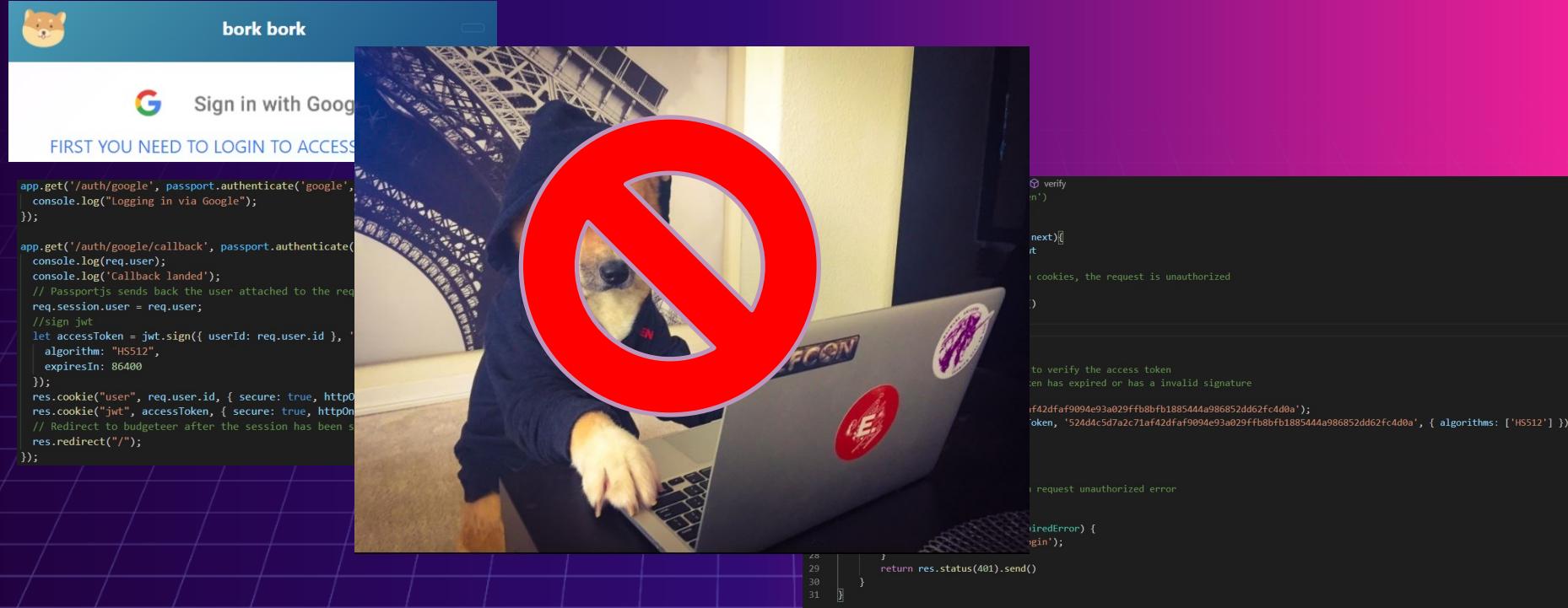
```
app.get('/auth/google', passport.authenticate('google', { scope: ['email', 'profile'] }), (req, res) => {
  console.log("Logging in via Google");
});

app.get('/auth/google/callback', passport.authenticate('google', { scope: ['email', 'profile'] }), (req, res) => {
  console.log(req.user);
  console.log('Callback landed');
  // Passportjs sends back the user attached to the request object, I set it as part of the session
  req.session.user = req.user;
  //sign jwt
  let accessToken = jwt.sign({ userId: req.user.id }, '524d4c5d7a2c71af42dfaf9094e93a029ffb8bfb1885444a986852dd62fc4d0a', {
    algorithm: "HS512",
    expiresIn: 86400
  });
  res.cookie("user", req.user.id, { secure: true, httpOnly: true, path: '/' })
  res.cookie("jwt", accessToken, { secure: true, httpOnly: true, path: '/' })
  // Redirect to budgeteer after the session has been set
  res.redirect("/");
});
```

Restricting access using OAuth

```
D: > node.js > scripts > js middleware.js > verify > verify
1  //const jwt = require('jsonwebtoken')
2  const jwt = ...require('jsonwebtoken');
3
4  exports.verify = function(req, res, next){
5      let accessToken = req.cookies.jwt
6
7      //if there is no token stored in cookies, the request is redirected to login page
8      if (!accessToken){
9          //return res.status(403).send()
10         return res.redirect('/login');
11     }
12
13     let payload
14     try{
15         //use the jwt.verify method to verify the access token
16         //throws an error if the token has expired or has a invalid signature
17         console.log(accessToken);
18         console.log('524d4c5d7a2c71af42dfaf9094e93a029ffb8bfb1885444a986852dd62fc4d0a');
19         payload = jwt.verify(accessToken, '524d4c5d7a2c71af42dfaf9094e93a029ffb8bfb1885444a98
20         console.log(payload);
21         next()
22     }
23     catch(e){
24         //if an error occurred return request unauthorized error
25         console.log(e)
26         res.clearCookie("jwt");
27         if(e instanceof jwt.TokenExpiredError) {
28             return res.redirect('/login');
29         }
30         return res.status(401).send()
31     }
32 }
```

Restricting access using OAuth



A dog is sitting at a laptop, looking at the screen. A large red 'no' symbol is overlaid on the image, indicating that the process shown is incorrect or prohibited.

The image is split into three sections:

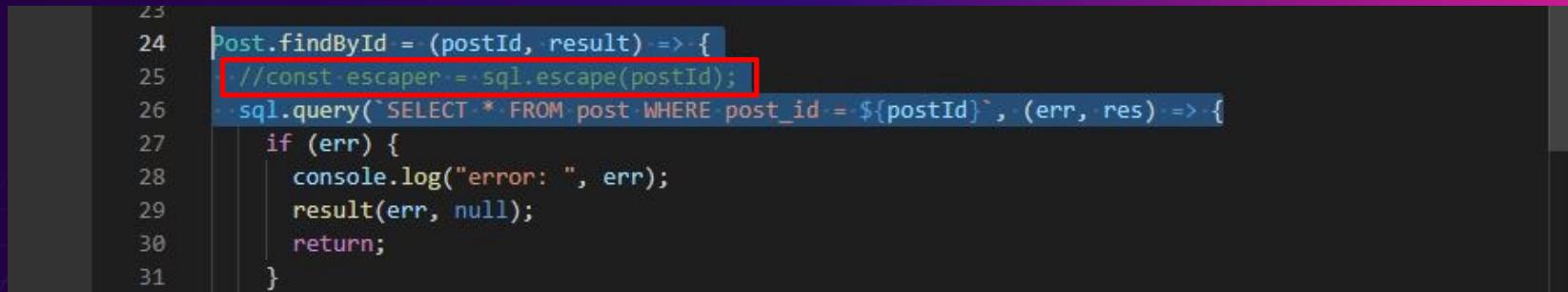
- Left Panel:** A screenshot of a web browser showing a Google sign-in page. The URL is `bork.bork/auth/google`. The page displays the message "FIRST YOU NEED TO LOGIN TO ACCESS".
- Middle Panel:** A screenshot of a laptop screen showing a command-line interface with Node.js code. The code handles OAuth authentication for Google. It includes imports for express, passport, and jwt, and defines routes for the main auth endpoint and a callback endpoint. It uses passport.authenticate to handle the login flow and signs a JWT with the user's ID.
- Right Panel:** A screenshot of a terminal window showing a Node.js application running. It logs the message "verify in" and then checks if the token is valid. If it fails, it logs "to verify the access token token has expired or has a invalid signature". If successful, it logs "request unauthorized error". Finally, it handles an unauthorized error by returning a 401 status and the message "Login".

Sql injection risk

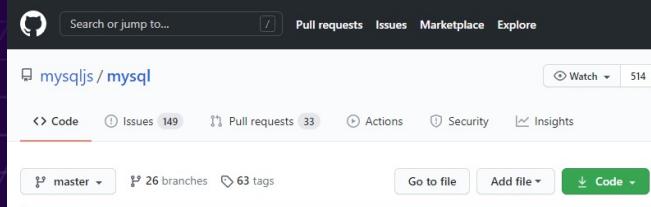
Hacking into the database to find out
who's a good boy



Sql injection risk



```
23
24 Post.findById = (postId, result) => {
25   //const escaper = sql.escape(postId);
26   sql.query(`SELECT * FROM post WHERE post_id = ${postId}`, (err, res) => {
27     if (err) {
28       console.log("error: ", err);
29       result(err, null);
30     }
31   })
}
```



mysqljs / mysql

Code Issues 149 Pull requests 33 Actions Security Insights

master 26 branches 63 tags Go to file Add file Code

Different value types are escaped differently, here is how:

- Numbers are left untouched
- Booleans are converted to `true` / `false`
- Date objects are converted to `'YYYY-mm-dd HH:ii:ss'` strings
- Buffers are converted to hex strings, e.g. `X'0fa5'`
- Strings are safely escaped
- Arrays are turned into list, e.g. `['a', 'b']` turns into `'a', 'b'`
- Nested arrays are turned into grouped lists (for bulk inserts), e.g. `[['a', 'b'], ['c', 'd']]` turns into `('a', 'b'), ('c', 'd')`
- Objects that have a `toSqlString` method will have `.toSqlString()` called and the returned value is used as the raw SQL.
- Objects are turned into `key = 'val'` pairs for each enumerable property on the object. If the property's value is a function, it is skipped; if the property's value is an object, `toString()` is called on it and the returned value is used.
- `undefined` / `null` are converted to `NULL`
- `Nan` / `Infinity` are left as-is. MySQL does not support these, and trying to insert them as values will trigger MySQL errors until they implement support.

My Workspace

New Import

GET http://localhost:3000

a / <http://localhost:3000/post>

GET http://localhost:3000/post/1608621364306 OR 1=1:

Params	Authorization	Headers (9)	Body	Pre-request Script	Tests	Settings
--------	---------------	-------------	------	--------------------	-------	----------

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ▾ UI

```
1   [
2     {
3       "post_id": 1608460935042,
4       "post_text": "This is a sample post to test everything out.",
5       "poster_id": 0,
6       "liked_count": 0,
7       "post_time": "2020-12-20T10:42:15.000Z"
8     },
9     {
10       "post_id": 1608464611544,
11       "post_text": "Another sample post to test things out",
12       "poster_id": 0,
13       "liked_count": 0,
14       "post_time": "2020-12-20T11:43:31.000Z"
15     },
16     {
17       "post_id": 1608555474457,
18       "post_text": "Testing everything once again after having made",
19       "poster_id": 0,
20       "liked_count": 0,
21       "post_time": "2020-12-21T12:57:54.000Z"
22     },
23     {
24       "post_id": 1608616576754,
25       "post_text": "Trying once again another post",
26       "poster_id": 0,
27       "liked_count": 0
28     }
29   ]
```

```
D:\node_js\scripts>node server.js
Server is running on port 3000.
Successfully connected to the database.
found post: 1 OR 1=1; [
  RowDataPacket {
    post_id: 1608460935042,
    post_text: 'This is a sample post to test everything out.',
    poster_id: 0,
    liked_count: 0,
    post_time: 2020-12-20T10:42:15.000Z
  },
  RowDataPacket {
    post_id: 1608464611544,
    post_text: 'Another sample post to test things out',
    poster_id: 0,
    liked_count: 0,
    post_time: 2020-12-20T11:43:31.000Z
  },
  RowDataPacket {
    post_id: 1608555474457,
    post_text: 'Testing everything once again after having ma
    poster_id: 0,
    liked_count: 0,
    post_time: 2020-12-21T12:57:54.000Z
  }
]
```

Sql injection risk

```
Post.findById = (postId, result) => {
  const escaper = sql.escape(postId);
  sql.query(`SELECT * FROM post WHERE post_id = ${escaper}` ,
```

```
D:\node_js\scripts>node server.js
Server is running on port 3000.
Successfully connected to the database.
1324; OR 1=1;
'1324; OR 1=1;'
```

a / http://localhost:3000/post

GET http://localhost:3000/post/1 OR 1=1;

Params Authorization Headers (9) Body Pre-request Script

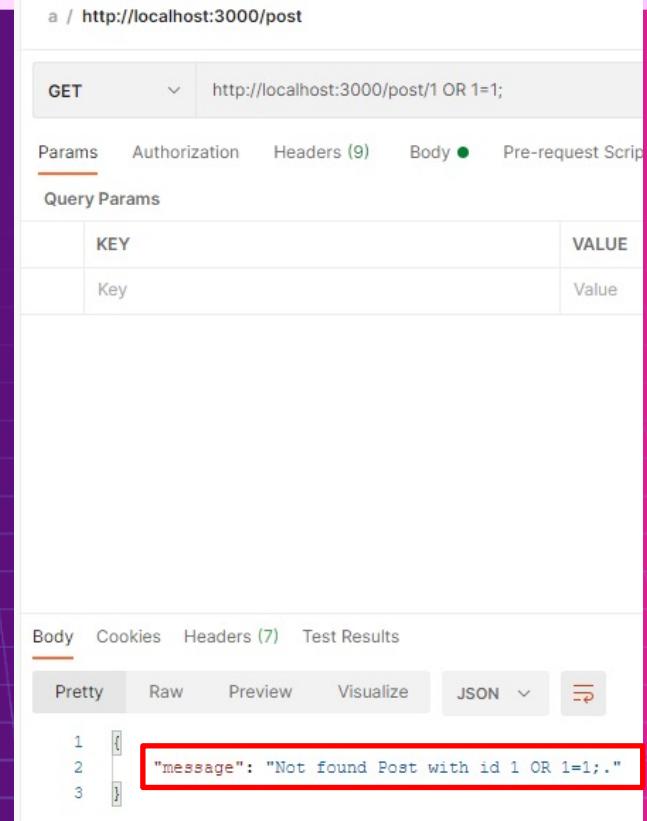
Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3
"message": "Not found Post with id 1 OR 1=1;."
```



Sql injection risk

Using “?” characters as placeholders for values to be escaped

```
3 // constructor
4 const Post = function(post) {
5   this.post_id = post.post_id;
6   this.post_text = post.post_text;
7   this.poster_id = post.poster_id;
8 };
9
10
11 Post.create = (newPost, result) => {
12   sql.query(`INSERT INTO post SET ?`, newPost, (err, res) => {
13     if (err) {
14       console.log("error: ", err);
15       result(err, null);
16       return;
17     }
18   });
19 }
```

Sql injection risk

Using “?” characters as placeholders for values to be escaped

```
3 // constructor
4 const Post = function(...args) {
5   this.post_id = post_id;
6   this.post_text = post_text;
7   this.poster_id = poster_id;
8 }
9
10 Post.create = (newPost) => {
11   sql.query(`INSERT INTO posts ${JSON.stringify(newPost)} `, (err, result) => {
12     if (err) {
13       console.log(`Error creating post: ${err}`);
14       result(err, false);
15       return;
16     }
17   });
18 }
```

Hacking into the database to find out
who's a good boy



```
DELETE FROM post WHERE post_id = 1;`
```

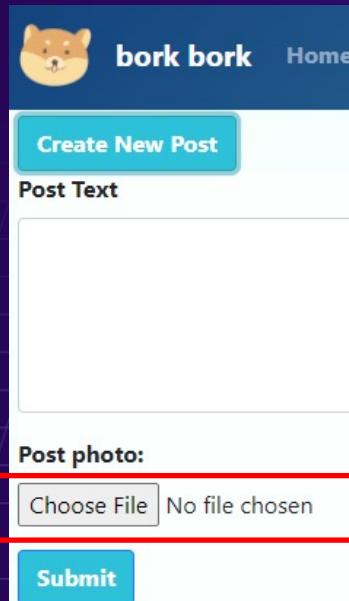
Malicious file upload risk

When u sign on amazon and rate all
the vacuums as 1 star

@cabbagecatmemes



Malicious file upload risk



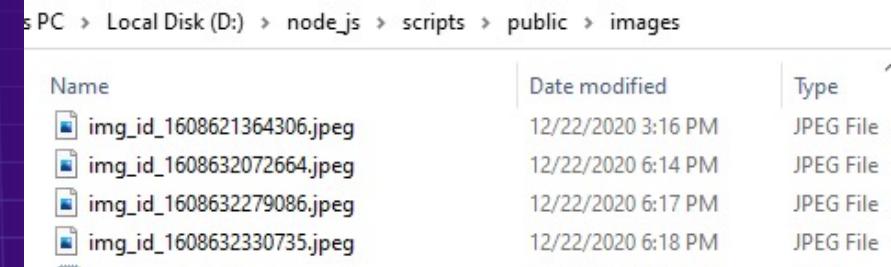
bork bork Home

Create New Post

Post Text

Post photo:

No file chosen



S PC > Local Disk (D:) > node_js > scripts > public > images

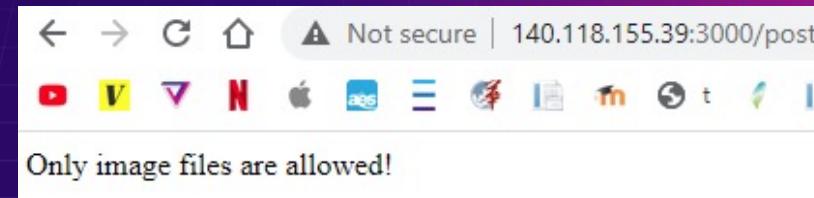
Name	Date modified	Type
img_id_1608621364306.jpeg	12/22/2020 3:16 PM	JPEG File
img_id_1608632072664.jpeg	12/22/2020 6:14 PM	JPEG File
img_id_1608632279086.jpeg	12/22/2020 6:17 PM	JPEG File
img_id_1608632330735.jpeg	12/22/2020 6:18 PM	JPEG File

Malicious file upload risk

```
D: > node.js > scripts > app > controllers > JS helpers.js > ...
1  const imageFilter = function(req, file, cb) {
2      // Accept images only
3      if (!file.originalname.match(/\.(jpg|JPG|jpeg|JPEG|png|PNG|gif|GIF)$/)) {
4          req.fileValidationError = 'Only image files are allowed!';
5          return cb(new Error('Only image files are allowed!'), false);
6      }
}
```

Malicious file upload risk

The screenshot shows a web application interface with a teal header bar. On the left is a small yellow dog icon. To its right, the text "bork bork" is displayed in a bold, sans-serif font. To the right of that are two links: "Home" and "About". Below the header is a light blue section containing a button labeled "Create New Post". Underneath this is a form area with a white background. The first part of the form is labeled "Post Text" and contains a text input field with the placeholder text "Testing out a malicious upload". Below this is another section labeled "Post photo:" which includes a "Choose File" button and a red-bordered input field. The input field contains the text "SetupEnIPExplorer_1.1.exe". At the bottom of the form is a blue "Submit" button.



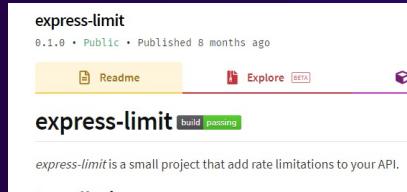
Malicious file upload risk

The image shows a screenshot of a web-based application with a teal header bar. On the left, there's a small dog icon and the text "bork bork". To its right are "Home" and "About" links. Below the header, a teal button says "Create New Post". Underneath it, a text input field contains the placeholder "Post Text" and the text "Testing out a malicious upload". A large red "no" symbol (prohibited sign) is overlaid on a photo of a dog wearing sunglasses. The photo is part of a post, which also includes a URL "55.39.3000/post" and some small icons. At the bottom left, there's a "Post photo:" section with a "Choose File" button and a dropdown menu showing "SetupEnIPExplorer". A blue "Submit" button is at the bottom right.

ddos risk /api abuse risk



ddos risk /api abuse risk



Accessing pages

```
app.set('/', limit({
  max:    20,          // 20 requests
  period: 60 * 1000 // per minute (60 seconds)
}), (req, res) => {
  res.sendFile(path.join(__dirname, 'app/views/index.html'));
});
app.get('/about', limit({
  max:    20,
  period: 60 * 1000
}), (req, res) => {
  res.sendFile(path.join(__dirname, 'app/views/about.html'));
});
app.get('/contact', limit({
  max:    20,
  period: 60 * 1000
}), (req, res) => {
  res.sendFile(path.join(__dirname, 'app/views/contact.html'));
});
```

Accessing api

```
app.post('/post', limit({
  max:    5,           // 5 requests
  period: 60 * 1000 // per minute (60 seconds)
}), post.create);
//app.post("/post", post.create);
app.get("/post", limit({
  max:    20,
  period: 60 * 1000
}), post.findAll);
app.get("/post/:postId", limit({
  max:    20,
  period: 60 * 1000
}), post.findOne);
```

ddos risk /api abuse risk

Trying to exceed limit:
post over 5 times within 60 sec

The screenshot shows a web browser window with a red status bar at the top stating "Trying to exceed limit: post over 5 times within 60 sec". The main content area displays a web page titled "bork bork" with a teal header. On the left, there's a "Create New Post" button and two input fields: "Post Text" containing "testing api abuse" and "Post photo:" with a "Choose File" button and "IMG_3223.jpeg" selected. At the bottom is a "Submit" button. The right side of the screen shows the browser's address bar with "Not secure | 140.118.155.39:3000/post" and a detailed error stack trace in the main content area.

```
Error: Too many requests
at D:\node_js\scripts\node_modules\express-limit\src\rate-limiter.js:74:33
at InMemoryStore.increment (D:\node_js\scripts\node_modules\express-limit\src\in-memory-store.js:26:9)
at D:\node_js\scripts\node_modules\express-limit\src\rate-limiter.js:67:25
at Layer.handle [as handle_request] (D:\node_js\scripts\node_modules\express\lib\router\layer.js:95:5)
at next (D:\node_js\scripts\node_modules\express\lib\router\route.js:137:13)
at Route.dispatch (D:\node_js\scripts\node_modules\express\lib\router\route.js:112:3)
at Layer.handle [as handle_request] (D:\node_js\scripts\node_modules\express\lib\router\layer.js:95:5)
at D:\node_js\scripts\node_modules\express\lib\router\index.js:281:22
at Function.process_params (D:\node_js\scripts\node_modules\express\lib\router\index.js:335:12)
at next (D:\node_js\scripts\node_modules\express\lib\router\index.js:275:10)
```

ddos risk /api abuse risk

Trying to post over 5 times within 60 sec

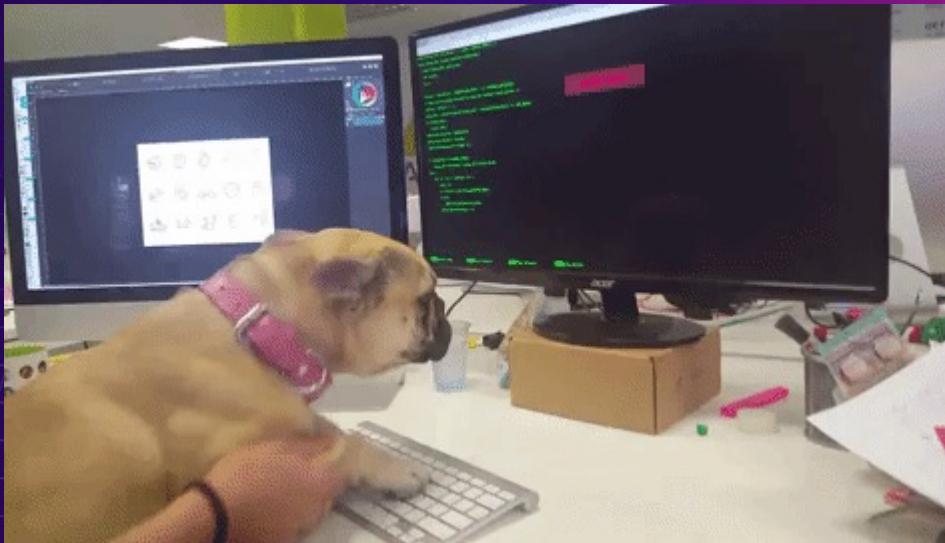
I can't hack into this with you looking over my shoulder

testing ddos

Post photo:
Choose File IMG_0833.jpeg
Submit

74:33
imit\src\in-memory-store.js:26:9)
67:25
express\lib\router\layer.js:95:5)
137:13)
\route.js:112:3)
express\lib\router\layer.js:95:5)
ib\router\index.js:335:12)
275:10)

Databases takedown risk



Sql backup database

Sql procedure

```
mysql> show procedure status like 'backup'
-> ;
+-----+-----+-----+-----+-----+-----+-----+
| Db   | Name  | Type   | Definer | Modified    | Created    | Security_type |
+-----+-----+-----+-----+-----+-----+-----+
| doggo | backup | PROCEDURE | root@localhost | 2020-12-29 21:46:49 | 2020-12-29 21:46:49 | DEFINER      |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.31 sec)
```

Sql procedure code

```
DELIMITER //
CREATE PROCEDURE backup()
BEGIN
DROP TABLE IF EXISTS `post_backup`;
CREATE TABLE post_backup SELECT * FROM post;
END //
DELIMITER ;

DELIMITER //
CREATE EVENT mybackup
ON SCHEDULE EVERY 1 DAY
DO
    CALL backup(); //
DELIMITER ;
```

Database

```
mysql> select* from post;
+-----+-----+
| post_id | post_text
+-----+-----+
| 1608460935042 | This is a sample post to test everything out.
| 1608464611544 | Another sample post to test things out
| 1608555474457 | Testing everything once again after having made some changes.
+-----+-----+
```

Database backup

```
mysql> select* from post_backup;
+-----+-----+
| post_id | post_text
+-----+-----+
| 1608460935042 | This is a sample post to test everything out.
| 1608464611544 | Another sample post to test things out
| 1608555474457 | Testing everything once again after having made some changes.
+-----+-----+
```

image backup database

node_js > scripts > public > images		
	Name	Date created
	img_id_1608460935042.jpeg	12/20/2020 6:42 PM
	img_id_1608464611544.jpeg	12/20/2020 7:43 PM
	img_id_1608555474457.jpeg	12/21/2020 8:57 PM
	img_id_1608616576754.jpeg	12/22/2020 1:56 PM
	img_id_1608621364306.jpeg	12/22/2020 3:16 PM
	img_id_1608632072664.jpeg	12/22/2020 6:14 PM
	img_id_1608632279086.jpeg	12/22/2020 6:18 PM
	img_id_1608632330735.jpeg	12/22/2020 6:18 PM
	img_id_1608988345906.jpeg	12/26/2020 9:12 PM
	img_id_1608989259523.jpeg	12/26/2020 9:27 PM
	img_id_1608989586785.jpeg	12/26/2020 9:33 PM

node_js > scripts > public > backup_images		
	Name	Date created
	img_id_1608460935...jpeg	12/31/2020 7:18 PM
	img_id_1608464611...jpeg	12/31/2020 7:18 PM
	img_id_1608555474...jpeg	12/31/2020 7:18 PM
	img_id_1608616576...jpeg	12/31/2020 7:18 PM
	img_id_1608621364...jpeg	12/31/2020 7:18 PM
	img_id_1608632072...jpeg	12/31/2020 7:18 PM
	img_id_1608632279...jpeg	12/31/2020 7:18 PM
	img_id_1608632330...jpeg	12/31/2020 7:18 PM
	img_id_1608988345...jpeg	12/31/2020 7:18 PM
	img_id_1608989259...jpeg	12/31/2020 7:18 PM
	img_id_1608989586...jpeg	12/31/2020 7:18 PM

image backup database

img_backup Properties (Local Computer)

General Triggers Actions Conditions Settings History

When you create a task, you can specify the conditions that

Trigger Details
One time At 10:04 PM on 12/29/2020

Edit Trigger

Begin the task: On a schedule

Settings
 One time Daily Weekly Monthly
Start: 12/29/2020 10:04:33 PM Synchronize across time zones
Recur every: 1 days

Advanced settings
 Delay task for up to (random delay): 1 hour
 Repeat task every: 1 hour for a duration of: 1 day
 Stop all running tasks at end of repetition duration
 Stop task if it runs longer than: 3 days
 Expire: 12/29/2021 10:22:41 PM Synchronize across time zones
 Enabled

OK Cancel

Task Scheduler Library

Next Run Time	Last Run Time	Last Run Result
12/30/2020 4:00:00 PM	12/29/2020 9:40:19 PM	The operation completed successfully. (0x0)
12/30/2020 3:27:59 PM	12/29/2020 4:31:47 PM	(0x1)
12/30/2020 9:40:18 PM	12/29/2020 9:40:19 PM	The operation completed successfully. (0x0)
12/29/2020 10:40:18 PM	12/29/2020 9:40:19 PM	The operation completed successfully. (0x0)
12/29/2020 10:07:00 PM	12/29/2020 9:30:28 PM	The operator or administrator has refused the request. (0x800710ED)
	12/29/2020 10:14:16 PM	The system cannot find the file specified. (0x80070002)
	11/30/1999 12:00:00 AM	The task has not yet run. (0x41302)
12/30/2020 12:07:05 PM	12/29/2020 9:25:29 PM	The operation completed successfully. (0x0)

up.bat

Edit Action

You must specify what action this task will perform.

Action: Start a program

Program/script: C:\Users\CNSLab1\CloudDrive\Desktop\school\security\up.bat

Add arguments (optional):

Start in (optional):

Selected Item

- Run
- End
- Disable
- Export...
- Properties
- Delete
- Help

backup.bat - Notepad

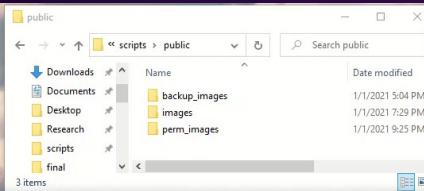
```
file | cut -f1 -d' '| grep -v '^$' | xargs -I{} cp -r {} D:\node_js\scripts\public\images\* D:\node_js\scripts\public\backup_images
exit 0
```

Demo backup databases

bork bork!



Image Database and Web Service takedown risk



```
Command Prompt - node server.js
eyIhbGcJ0IJU1U2U1l1sln5c161pxWC19 eyI1c2VysWQ10lxTH50D3h3ay
MDQ3MjcwNzU1Nz11LC1pXK010xE2MDk4MzcxMzs1w74C16MTy0TkHmZuOHO
.v146Lnts8ksu0GEH1y7KmDQy34El340-9P5y9ak3w01MNIG71WpWfnt3Ht-XnQ
tdySTHLxLMED1aXKC2w
5244e5d7a2c71af42difa9094e93a029ff8bf1885444a98652dd62fc4d0a
{ userId: '113984770204727075572', iat: 1609837138, exp: 16099235
38 }
^C
D:\node_js\scripts>node server.js
Server is running on port 8080
Success! Your computer is connected to the database.
eyIhbGcJ0IJU1U2U1l1sln5c161pxWC19 eyI1c2VysWQ10lxTH50D3h3ay
MDQ3MjcwNzU1Nz11LC1pXK010xE2MDk4MzcxMzs1w74C16MTy0TkHmZuOHO
.v146Lnts8ksu0GEH1y7KmDQy34El340-9P5y9ak3w01MNIG71WpWfnt3Ht-XnQ
tdySTHLxLMED1aXKC2w
5244e5d7a2c71af42difa9094e93a029ff8bf1885444a98652dd62fc4d0a
{ userId: '113984770204727075572', iat: 1609837138, exp: 16099235
38 }
eyIhbGcJ0IJU1U2U1l1sln5c161pxWC19 eyI1c2VysWQ10lxTH50D3h3ay
MDQ3MjcwNzU1Nz11LC1pXK010xE2MDk4MzcxMzs1w74C16MTy0TkHmZuOHO
.v146Lnts8ksu0GEH1y7KmDQy34El340-9P5y9ak3w01MNIG71WpWfnt3Ht-XnQ
tdySTHLxLMED1aXKC2w
5244e5d7a2c71af42difa9094e93a029ff8bf1885444a98652dd62fc4d0a
{ userId: '113984770204727075572', iat: 1609837138, exp: 16099235
38 }
```

```
Command Prompt - mysql -u"root" -p
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

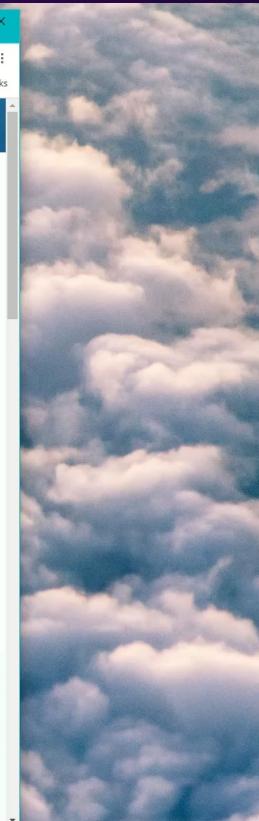
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



A screenshot of a web browser window titled 'bork bork!'. The URL is 'borkborkherokuapp...'. The page displays a post from user '73819773124' with the caption 'Happiest pup ever.' and a photo of a pug. Below it is another post from the same user with the caption 'Waiting for borkbork.herokuapp...' and a photo of two dogs in a stroller.



SQL Database and Web Service takedown risk

File Explorer:

- public
- Downloads
- Documents
- Desktop
- Research
- scripts
- final

scripts > public

- Name
- Date modified

Name	Date modified
backup_images	1/1/2021 5:04 PM
images	1/5/2021 5:02 PM
perm_images	1/1/2021 9:25 PM

Command Prompt - node server.js

```
AC
D:\node_js\scripts>node server.js
Server is running on port 80.
Success! Connected to MySQL database.
eyJhbGciOiJIUzIwMjlsIn5C161kpXVCJ9.eylc2vysWQj0i1xMTM50DQ3Nzay
MDQ3N1cwMjU1Nz1lLC1jYX10JE20k4mccMzgs1w4c16NTw0Tkwz2eH0.-w146Lwntsksks0UGEH1yk7kmQy34kL340-9P5yak3w0IMN1G71VpPmf3Ht-XsQ
tdyTNLsLMED1afKZ2w
5244c5d7a2c71af42difa0904e93a029ff8bf188544a986852dd62fc440a
{ user_id: '113984770204727075572', iat: 1609837138, exp: 1609923538 }
eyJhbGciOiJIUzIwMjlsIn5C161kpXVCJ9.eylc2vysWQj0i1xMTM50DQ3Nzay
MDQ3N1cwMjU1Nz1lLC1jYX10JE20k4mccMzgs1w4c16NTw0Tkwz2eH0.-w146Lwntsksks0UGEH1yk7kmQy34kL340-9P5yak3w0IMN1G71VpPmf3Ht-XsQ
tdyTNLsLMED1afKZ2w
5244c5d7a2c71af42difa0904e93a029ff8bf188544a986852dd62fc440a
{ user_id: '113984770204727075572', iat: 1609837138, exp: 1609923538 }
eyJhbGciOiJIUzIwMjlsIn5C161kpXVCJ9.eylc2vysWQj0i1xMTM50DQ3Nzay
MDQ3N1cwMjU1Nz1lLC1jYX10JE20k4mccMzgs1w4c16NTw0Tkwz2eH0.-w146Lwntsksks0UGEH1yk7kmQy34kL340-9P5yak3w0IMN1G71VpPmf3Ht-XsQ
tdyTNLsLMED1afKZ2w
5244c5d7a2c71af42difa0904e93a029ff8bf188544a986852dd62fc440a
{ user_id: '113984770204727075572', iat: 1609837138, exp: 1609923538 }

MySQL>
mysql> drop table post;
ERROR 1046 (3D000): No database selected
mysql> use dogso;
Database changed
mysql>
```

Web Browser:

bork bork!

Create New Post

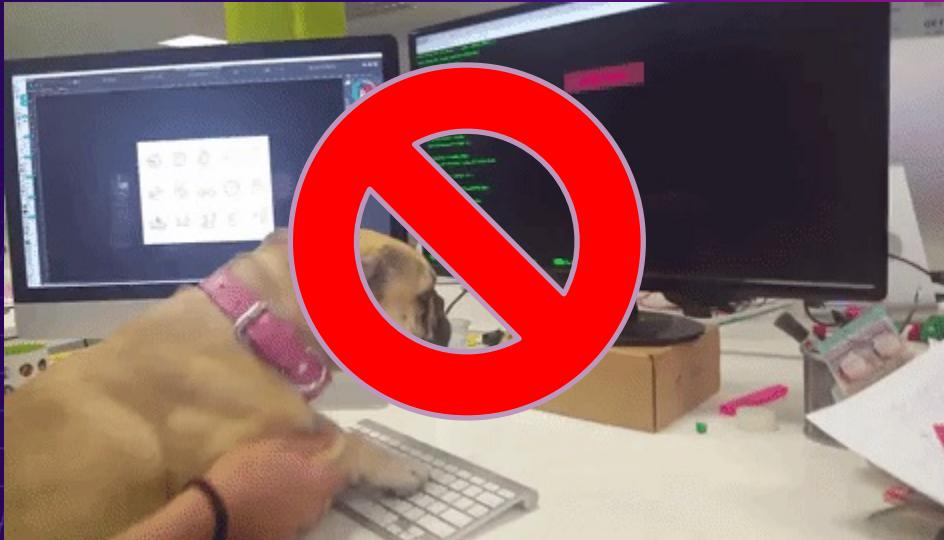
User: 73819773124

Happiest pug ever.

User: 73819773124

5:04 PM 1/5/2021 ENG

Database takedown risk



More threats looming every corner



Thank you!

Any Questions?

