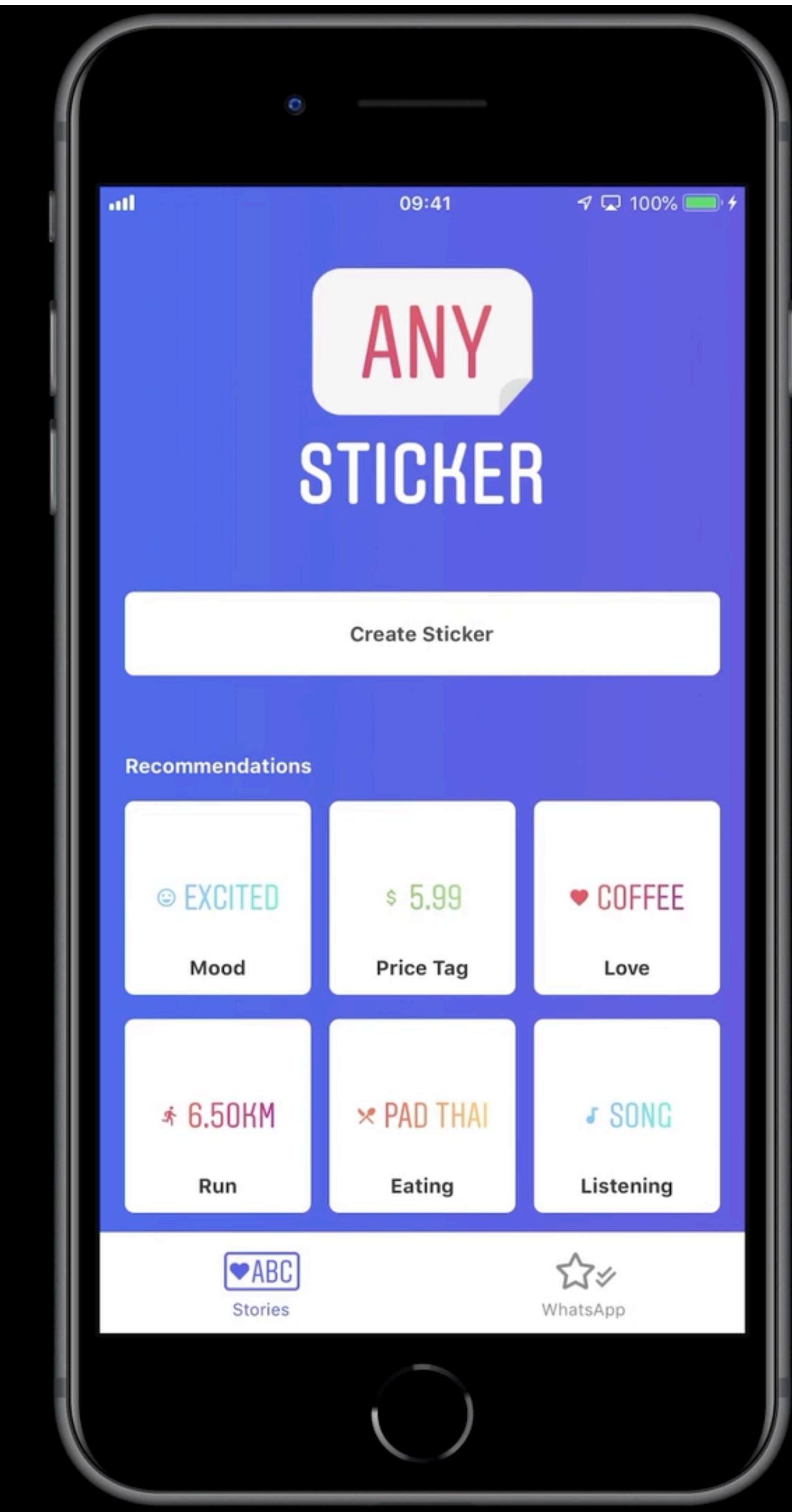
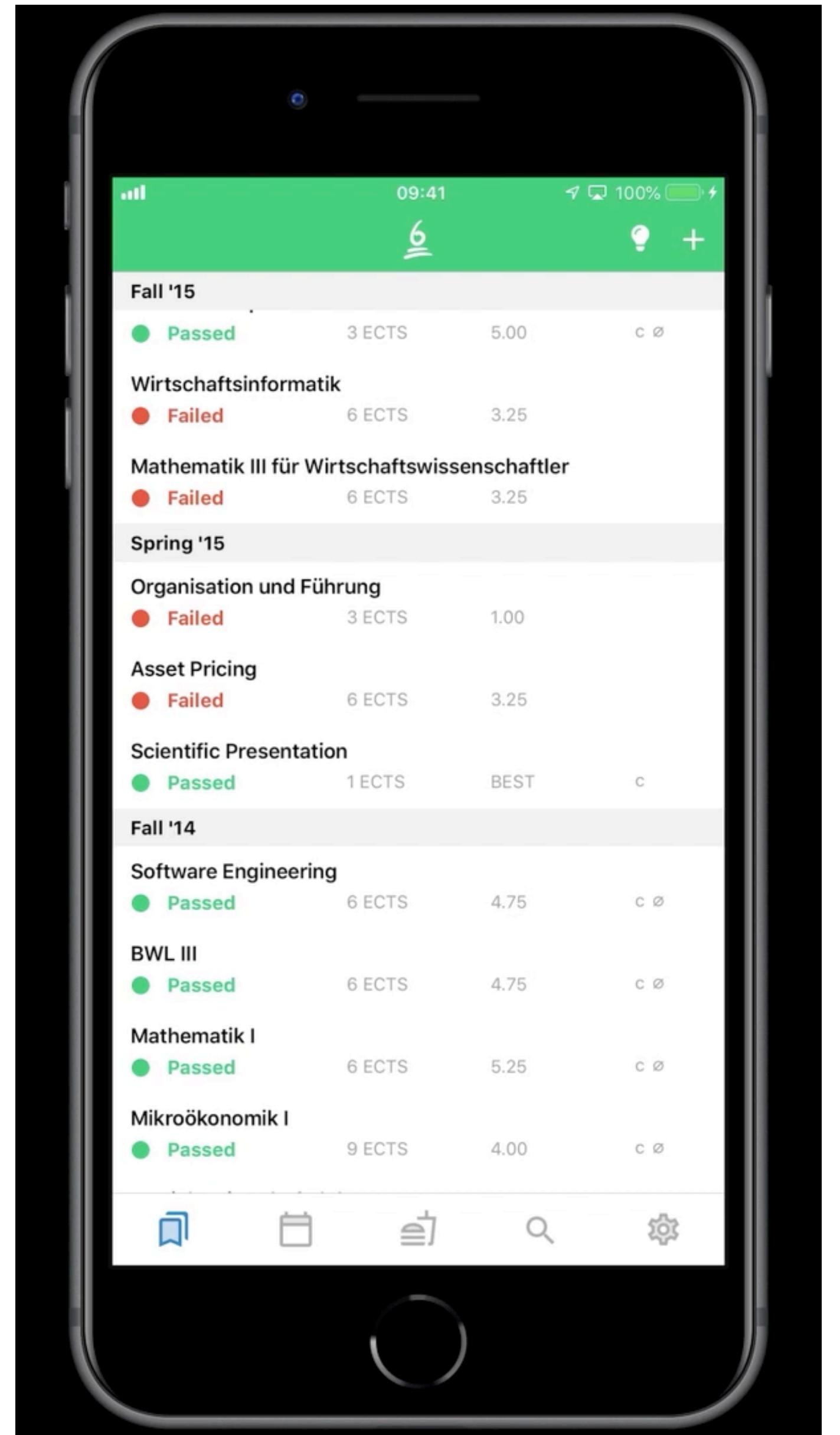


How to write ultra-concise code in Express.js

Jonny Burger



About me

Software Developer at Axelra
(axelra.com)





@JNYBGR

← **Jonny Burger**
1.343 Tweets



Profil bearbeiten

Jonny Burger
@JNYBGR

-2.53x hacker, nullx engineer - working on @BestandeApp @AnystickerApp.
Ran 1251 days in a row: [jonny.run](#)
Part time hacker at [@Axelra_AG](#)

⌚ Zürich, Schweiz ⌚ [jonny.io](#) 📅 Seit Oktober 2009 bei Twitter

692 Folge ich 441 Follower

[Tweets](#) [Tweets und Antworten](#) [Medien](#) [Gefällt mir](#)

Angehörteter Tweet

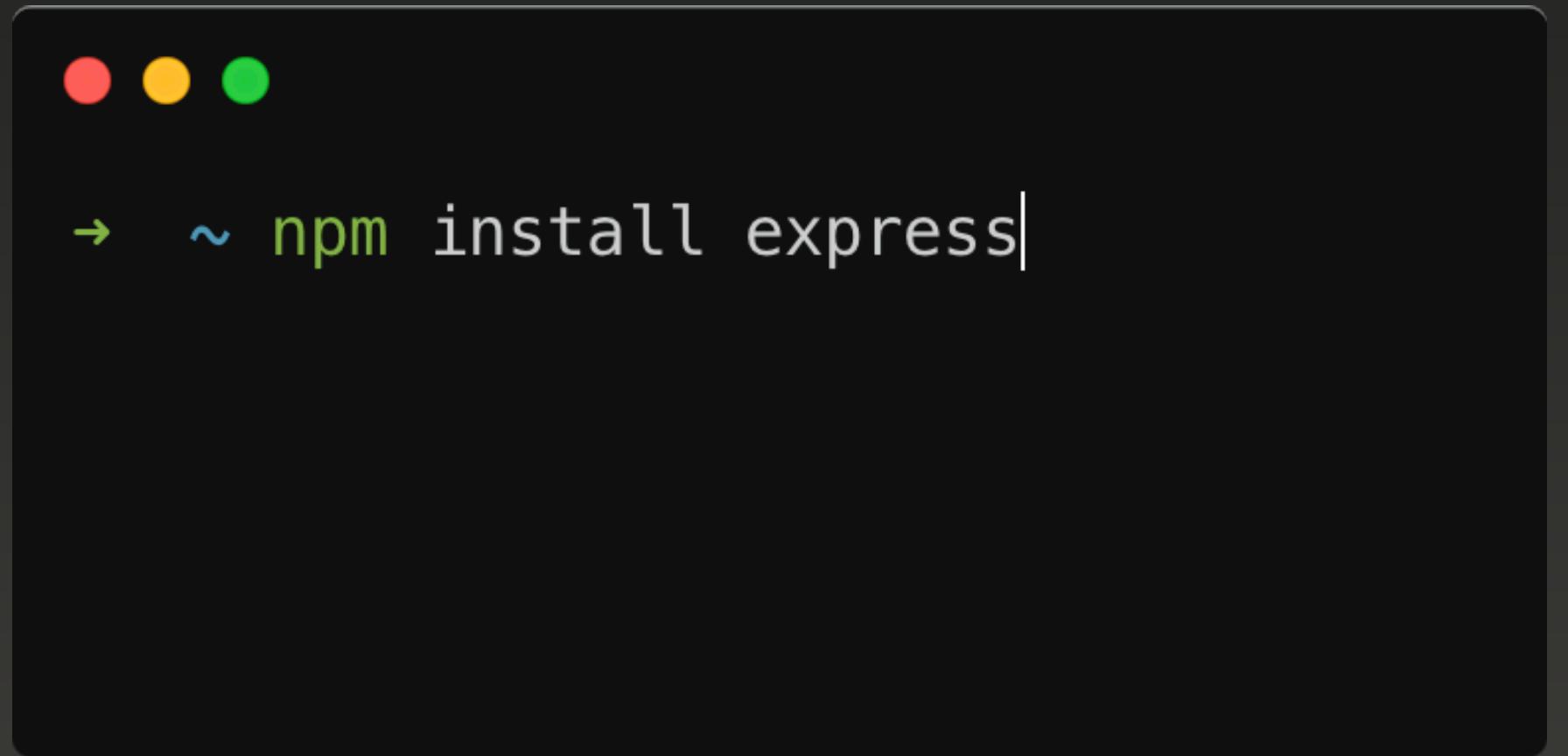
Jonny Burger @JNYBGR · 9. Juni
We are finally live with [@AnyStickerApp!](#)

Thank you for following the journey on Twitter 😊 It was and will continue to be a lot of fun!

See the result here: [Link](#)

Express.js

- 📡 Framework for creating HTTP servers for Node.JS
- 👴 One of the oldest packages on NPM
- ⭐ Over 9'000'000 downloads per week



```
● ● ●
→ ~ npm install express|
```

A screenshot of a terminal window on a dark background. At the top, there are three small colored circles (red, yellow, green) representing window control buttons. Below them, the text '→ ~ npm install express|' is displayed in white, indicating an active command line session.



index.js

```
1 const express = require('express');
2
3 const app = express();
4
5 app.get('/hello', (request, response) => {
6     response.json({
7         hello: 'world'
8     });
9 });
10
11 app.listen(8000);
12 console.log('Server has started on port 8000');
13
```



```
→ 1_hello_world git:(master) ✘ node index.js  
Server has started on port 8000
```

|

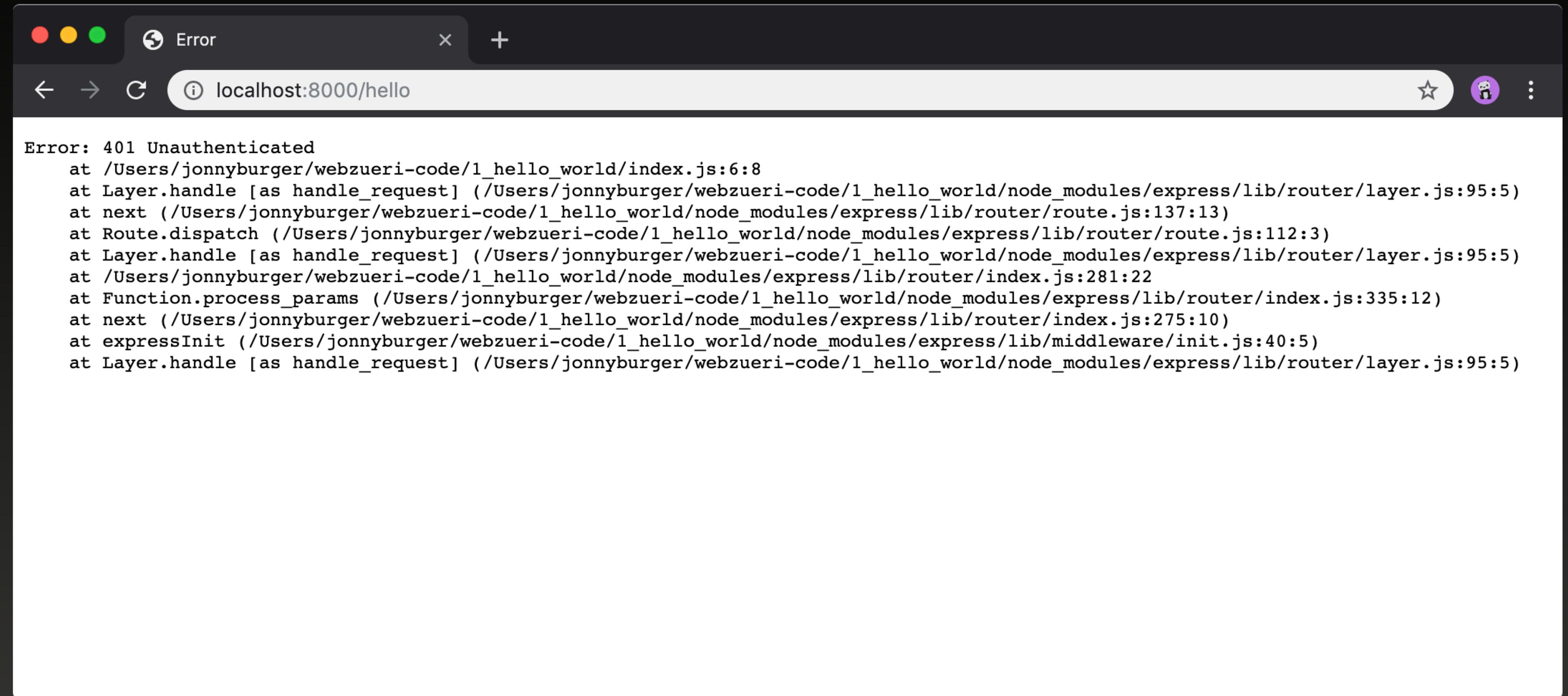
A screenshot of a web browser window displaying a JSON object. The browser has a dark mode interface with a light gray header bar. The address bar shows the URL `localhost:8000/hello`. The main content area displays the following JSON:

```
{  
  "hello": "world"  
}
```



index.js

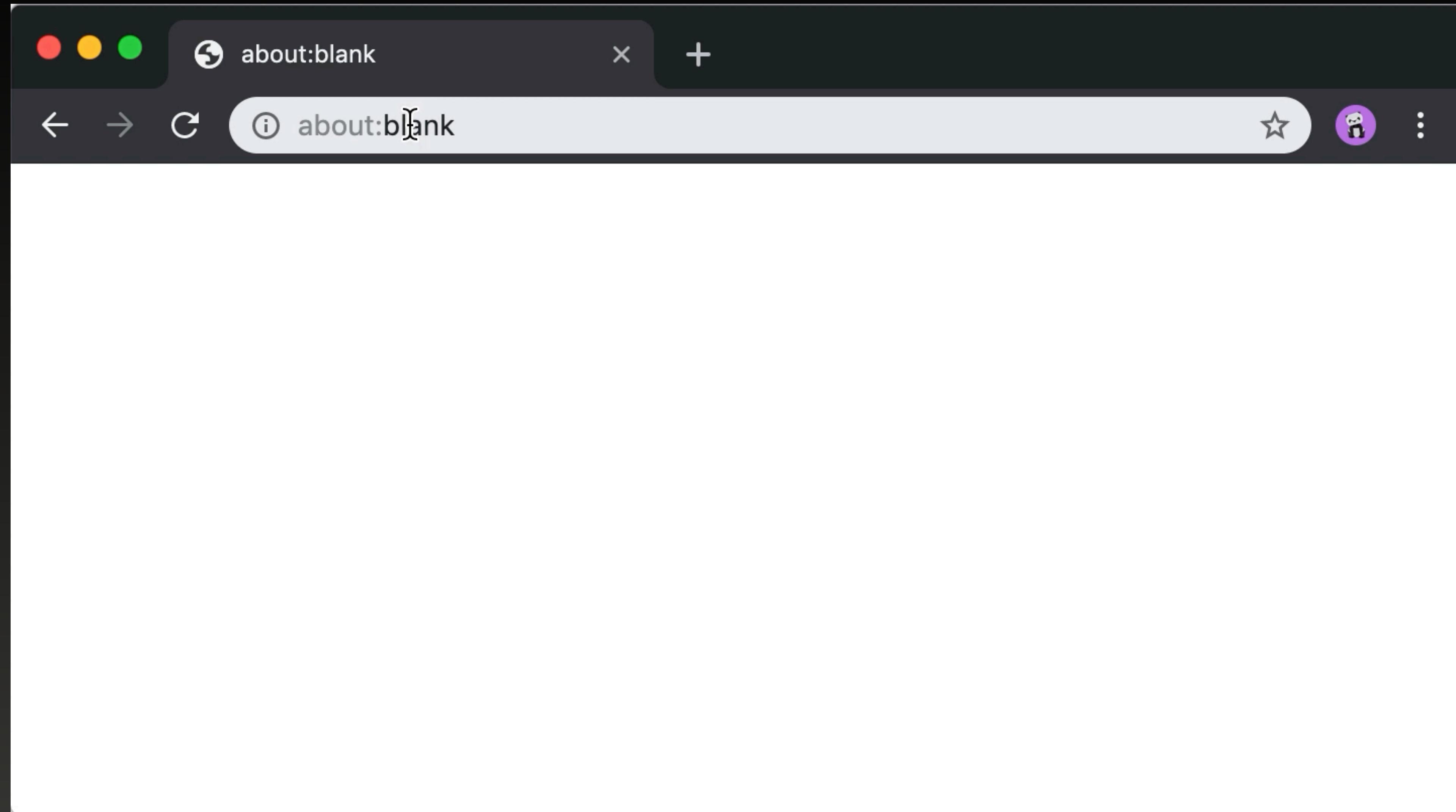
```
1 const express = require('express');
2
3 const app = express();
4
5 app.get('/hello', (request, response) => {
6     throw new Error('401 Unauthenticated')
7 });
8
9 app.listen(8000);
10 console.log('Server has started on port 8000');
11
```





index.js

```
1 const express = require('express');
2
3 const app = express();
4
5 app.get('/hello', async (request, response) => {
6     throw new Error('401 Unauthenticated')
7 });
8
9 app.listen(8000);
10 console.log('Server has started on port 8000');
11
```



Introducing the “async handler”



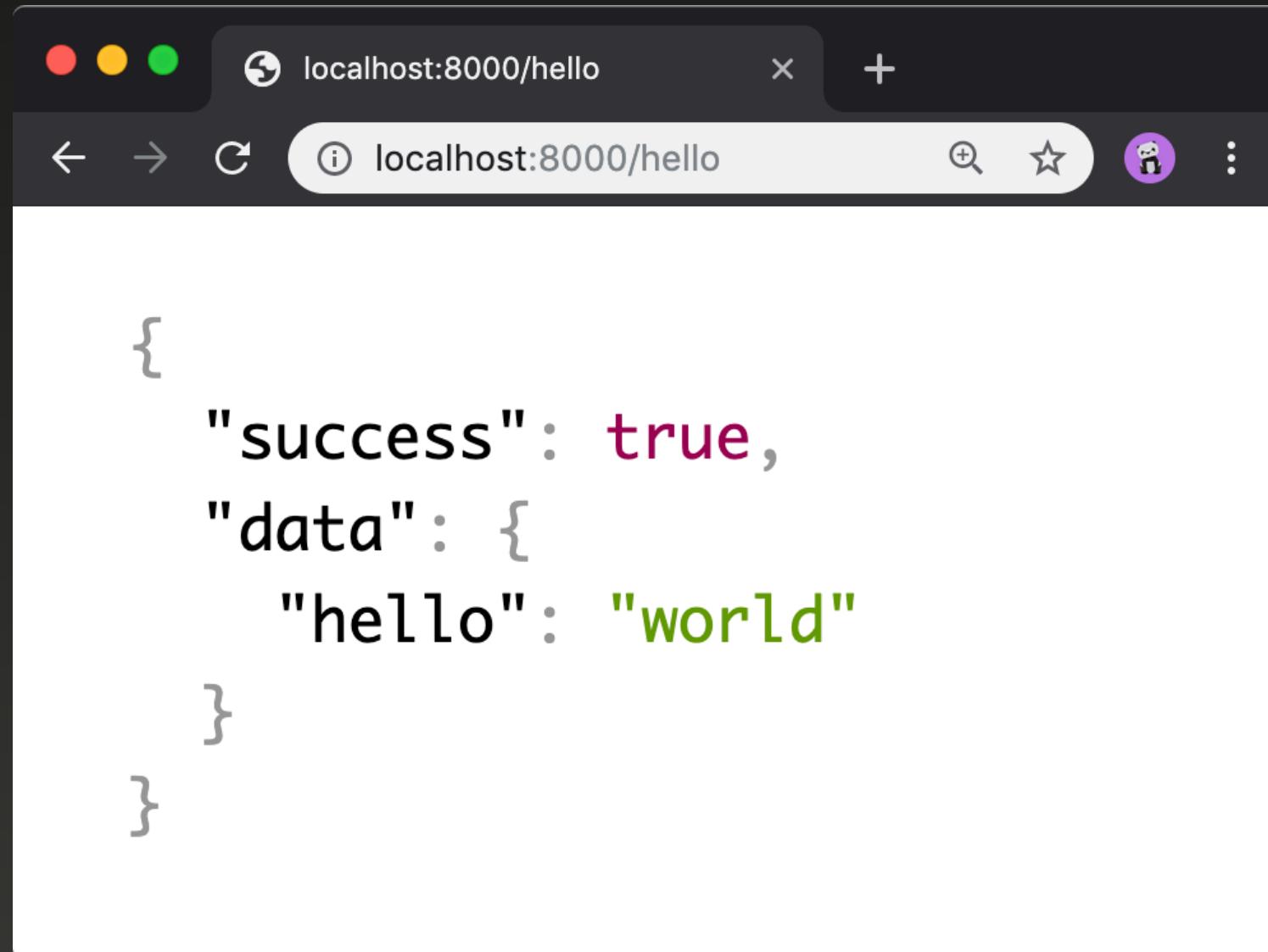
async-handler.js

```
const asyncHandler = fn => {
  return async function(request, response) {
    try {
      const data = await fn(request, response);
      response.json({
        success: true,
        data
      });
    } catch (err) {
      const statusCode = err.status || 500;
      response.status(statusCode).json({
        success: false,
        error: err.message
      });
    }
  };
};
```

```
app.get('/hello',  
  asyncHandler(() => {  
    return {  
      hello: 'world'  
    };  
  })  
);
```

```
app.get('/hello',  
  asyncHandler(async () => {  
    throw new Error('Oops!')  
  })  
)
```

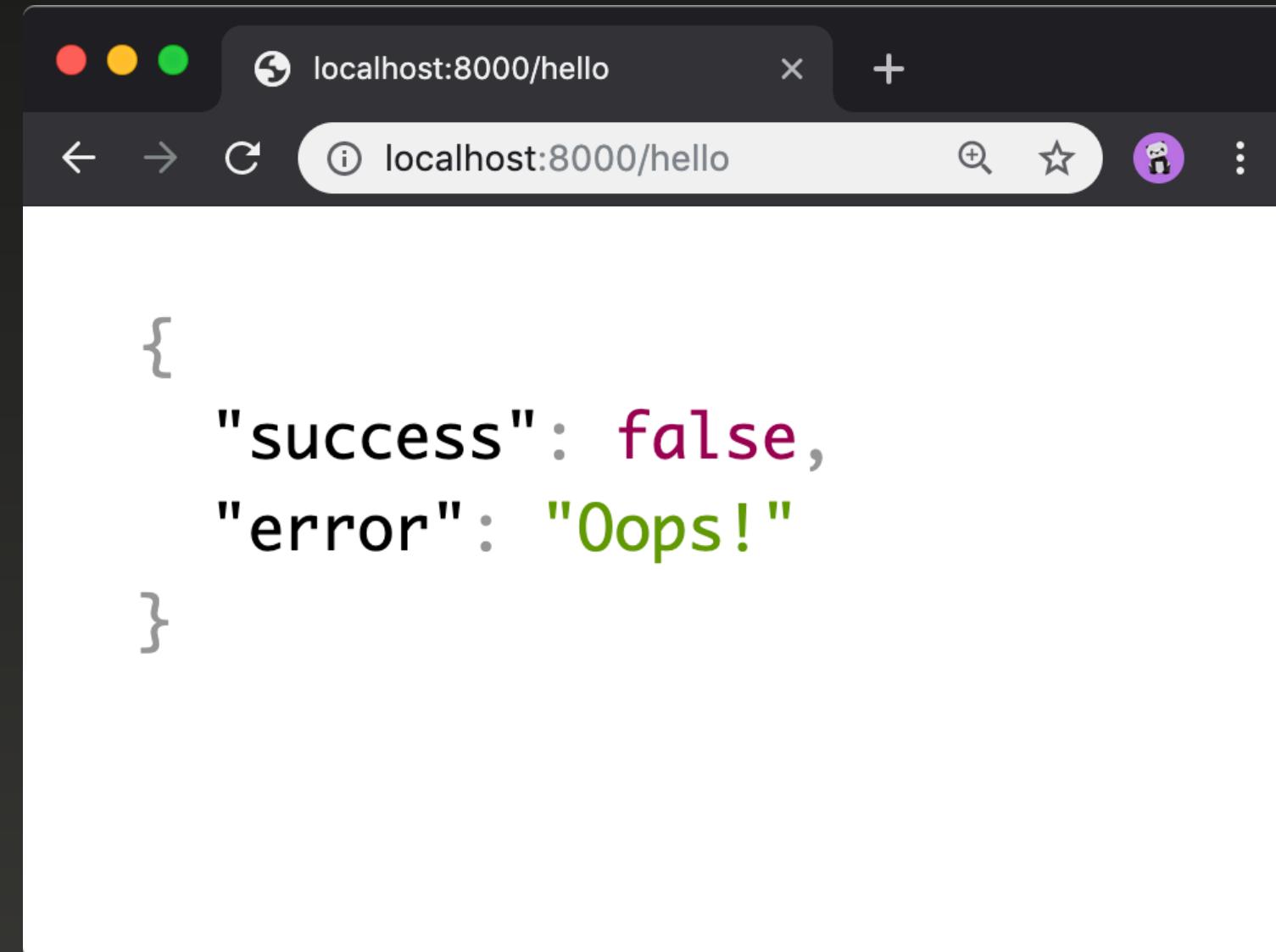
```
const httpError = require('http-errors')  
  
app.get('/hello',  
  asyncHandler(() => {  
    throw new httpError(401, 'Unauthenticated')  
  })  
);
```



A screenshot of a web browser window titled "localhost:8000/hello". The address bar shows the same URL. The page content is a JSON object:

```
{  
  "success": true,  
  "data": {  
    "hello": "world"  
  }  
}
```

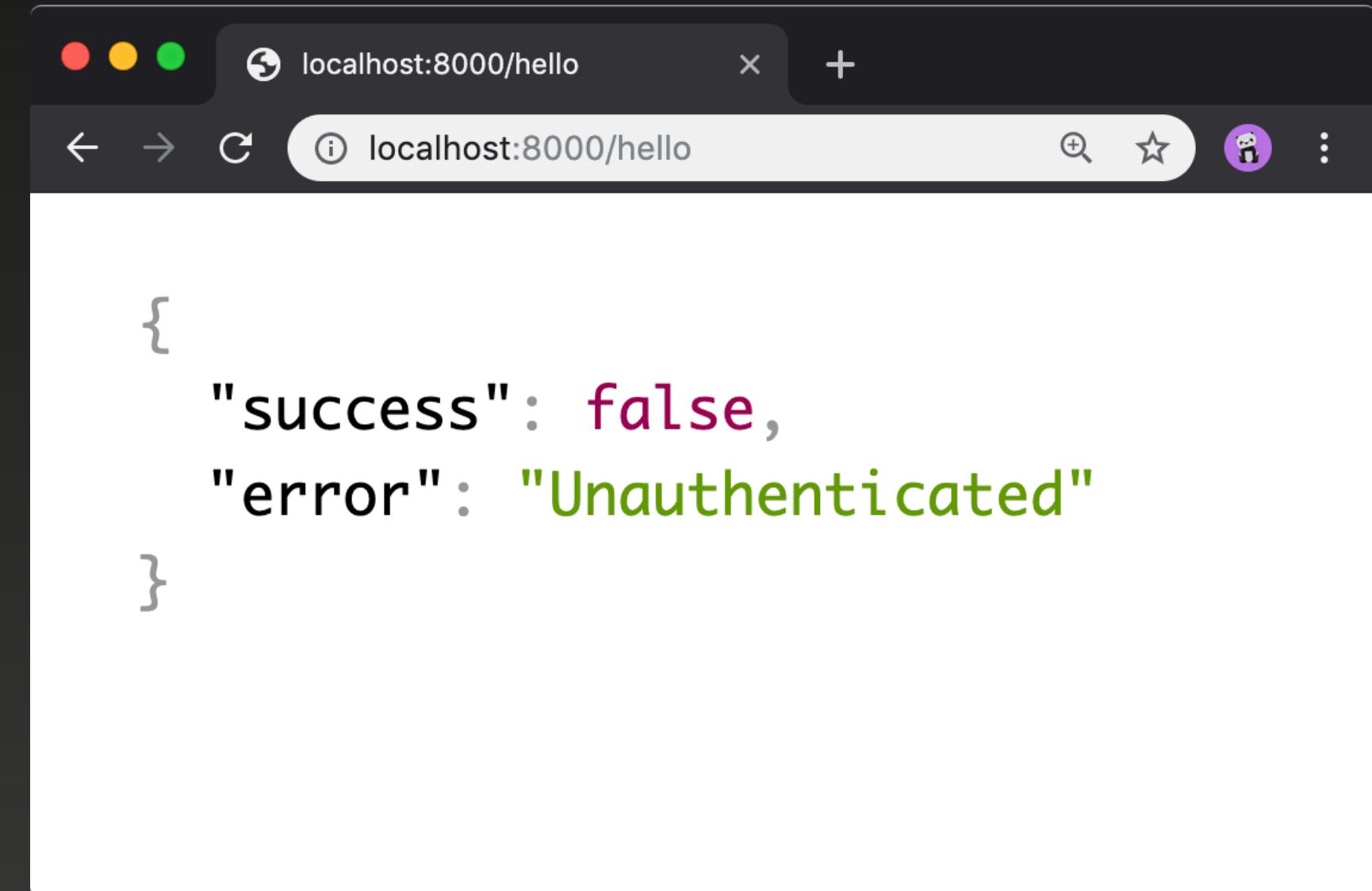
200 OK



A screenshot of a web browser window titled "localhost:8000/hello". The address bar shows the same URL. The page content is a JSON object indicating an error:

```
{  
  "success": false,  
  "error": "Oops!"  
}
```

500 Internal Server Error



A screenshot of a web browser window titled "localhost:8000/hello". The address bar shows the same URL. The page content is a JSON object indicating an authentication error:

```
{  
  "success": false,  
  "error": "Unauthenticated"  
}
```

401 Unauthenticated

End of slides - let's code!

Code available on Github:
github.com/JonnyBurger