Node.js Week 3

Building a REST API

SOCIAL
HACKERS
ACADEMY

Agenda

- Recap from Week 2
- Previous homework
- Q&A
- Express.js vs native http
- Testing with cURL and Postman
- Building a REST API
- Homework

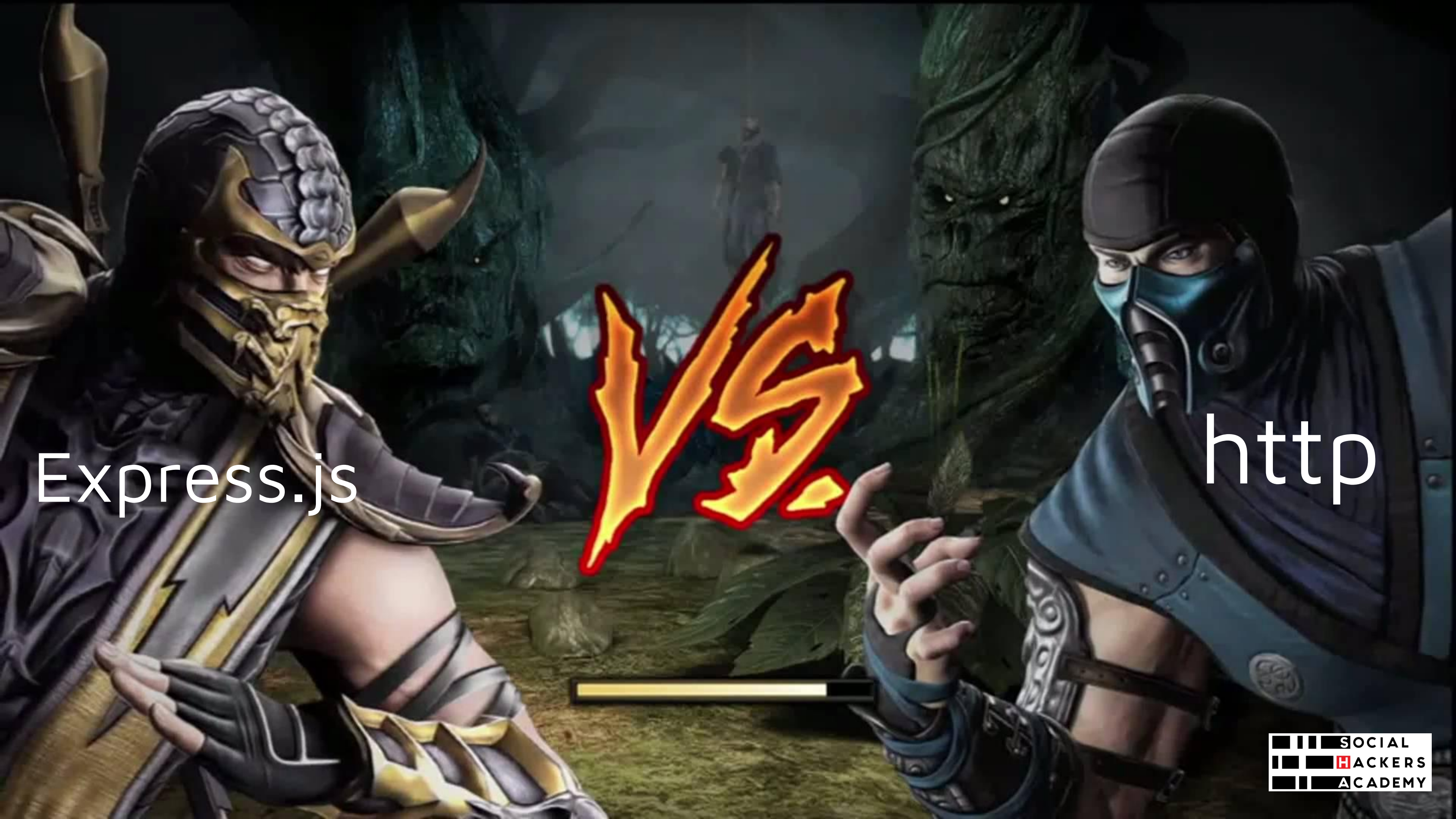
Week 2 Recap

- process.argv
- fs
- CLI
- Sync vs Async
- CRUD

Previous Homework

- Write a Node.js command line application
- The user must be able to run the file using *node index.js* or *node* . in the project directory
- There must be a help section that lists all the commands and a short description for each of them
- The user must be able to add, remove and list to-dos
- The user must be able to remove all to-dos at once
- The following commands must be present: *help*, *list*, *add*, *remove*, *reset*
- Bonus assignment: JSON, actions in different files, use *commander* library, add the *update* command

Q&A



SOCIAL HACKERS ACADEMY

Remember commander?

SOCIAL HACKERS ACADEMY

Express.js is to **http**
what **commander** is to **process.argv**

SOCIAL HACKERS ACADEMY

curl://

Meet you best friend cURL!

SOCIAL HACKERS ACADEMY

But you can make life easier with [Postman]

Building a REST API

A great description of what REST (Representational State Transfer) is

Also check out the HTTP status definitions

Homework

- Write a Node.js REST API application based on the previous week’s homework
- Add four more actions: *readTodo*, *clearTodos*, *markAsDone*, *markAsNotDone*
- All requests that need a body should be in JSON format, and follow the request structure of the other actions
- All responses should be in JSON format, and follow the response structure of the other actions
- Follow the anatomy of the project
- Make sure your code is DRY
- Follow the REST design principles: use the proper method, response status codes, and consistent URL paths
- Test your API using Postman