CS 4220

- Current Trends in Web Development -

Application Design and Development with Node.js

Prof. Cydney Auman

AGENDA

01 Prompting in CLIs

02 CommonJS Module System

03 module.exports & require

04 ES Module System

Timeline

Week 08 - 3/15

- Adding features to CLIs
- Midterm Demo + Discussion
- Homework 2 Review

Week 09 - 3/22

- Midterm Due - Sunday 3/26 at 11:59 PM
- Remote Zoom Class Wednesday 3/22 - focused on Q&A for Midterm

Week 10

- Spring Break (https://www.calstatela.edu/academicresources/academic-calendar)

prompts

Prompts is a Node Module that can be found on NPM
`https://www.npmjs.com/package/prompts`

Prompts helps with the process of **prompting for inputs** in CLI applications.   It is also able to, parse these inputs and perform validation.

* Prompts differs from Yargs as it provides the user interface and the inquiry session flow. It is not command line program utility like Yargs. Instead it provides various styles of input prompts - yes/no, selects, raw input and more for a command line application.

prompts

`prompts(`*`questions`*`) returns a Promise`

questions: Array of prompt objects. These are the questions the user will be prompted. There are several support prompt types.

Prompts Objects are JavaScript objects that define the "questions". Each contains several keys to compose the question prompt - these keys indicate the type of question, name, message, choices array, validate function, etc.

When creating a `select` or `multiselect` - we need to provide a choices array. This array can be built as an array of object which contains a title (to display in list) and a value that gets returned.

Common JS Module System

The CommonJS Module System is built into Node.js. This allowed for backend application to modularize code by solving the problem of importing and exporting. CommonJS was around before the release of ES Modules.

Exporting

`module.exports` is used as the instruction that tells Node.js which parts of code - such as functions to export from a file to be used in other files .

Importing

`require(String)` is the function used to import Node.js core modules, third-party modules or files.

Common JS Module System

When using the require function - it uses the following rules:

- Does the string argument point to a core module (ex: fs, path, http)
    - if so then use it
- Does the string argument point to an *installed* node package (ex: superagent, yargs)
    - if so then use it
- Does the string argument point to a filename by relative path
    - if so then use it

- If all these fail then throw an error - `MODULE NOT FOUND`

ES Module System

The ES Module System was created as a way to standardize the JavaScript module system. And it has now become the standard format for encapsulating JavaScript code for reuse in browsers application.

Because Node.js was built using the CommonJS Module System this means users have to opt-in to using the ES Module System. Meaning we have to configure our Node.js application in a certain way to interact and work with the ES Module System syntax.

Typically this means we must set an option in out package.json (`{ "type": "module" }`). And sometimes we use file extensions with `.mjs` to indicate that the source code file contains an ES Module for use with a Node.js.

Resources

Mozilla Developer Docs

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules

Prompts

- https://www.npmjs.com/package/prompts
- https://www.npmjs.com/package/prompts#-types

API

- https://deckofcardsapi.com/

Review and Prep

Review

- Review Slides
- Review and Run Code Examples

Finish

API + Team Selection

- Pick a team of 2 - 3 for Midterm
- Pick a team captain
- Select an API
- Fill out the Google Form

```
console.log(‘Week 08’);
console.log(‘Code Examples’);
```