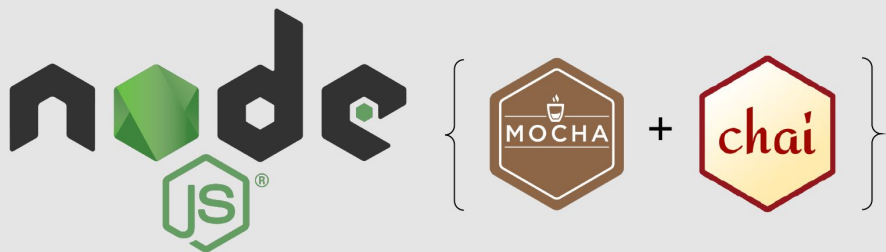


W2D1 - TDD With Mocha and Chai



**NOT SURE IF I WROTE GOOD
CODE**

OR SHITTY UNIT TESTS

made on imgur

AGENDA

Automated Testing

Test-Driven Development

Demo

NodeJS Assert

Mocha


Chai

Automated Testing

Automated testing is the practice of writing code to ***programmatically test*** the actual code we want to write.

slido

Why automated testing? What are the benefits?

 Start presenting to display the poll results on this slide.

Automated Testing

*The main goal of automated testing is to **improve overall code quality!***


Automated Testing

Other benefits include:

- Save time compared to manual testing
- Easy to see if new changes to our code is breaking anything
- Think about potential bugs up-front
- Write more modular code
- Could be integrated into build workflow

slido

What is Test Driven Development (TDD)?

 Start presenting to display the poll results on this slide.

Test-Driven Development

- **Test-driven development (TDD) relies on tests to drive development.**
- **In other words, write the test first and write the code to make the test pass.**

Test-Driven Development

1. **Write a failing test** to indicate which functionality needs to be added and how it should behave.
2. Write the **minimal amount of code** to make the test pass. At this stage, the code doesn't have to be elegant or clean.
3. **Refactor the code.** Clean up the code to make it more readable and maintainable.

Demo

1. **NodeJS Assert**
2. **Mocha**
3. **Chai**

Demo

Try Catch handle errors more gracefully.

```
try {  
  // What we need to execute  
  
} catch (error) {  
  // Executes if an error is  
  // thrown  
  
}
```

Details of the error are stored in the error variable in the form of an object.

- error.name
- error.actual
- error.expected
- error.message

modules

Modules are keeping a set of functions in separate files. We can think of modules as JavaScript libraries.

1. Core Modules (net, http, assert, etc)
2. Third-Party Modules (npm)
3. Local Modules

modules

Modules are keeping a set of functions in **separate files**. We can think of modules as **JavaScript libraries**.

- Keeping modules separate ensures that our code remains **well organized and modular**
- These functions can be added to our app **reused any number of times**

Mocha



Mocha is a feature-rich **JavaScript test framework** running on [Node.js](#) and in the browser, making asynchronous testing *simple* and *fun*.

mochajs.org

Chai



Chai is a **BDD / TDD assertion library** for node and the browser that can be delightfully paired with any javascript testing framework.

Chai



Behavior Driven Development (BDD)

- Describes the behavior of the system from the customer's point of view
- Based on users' stories
- Uses a more natural language that everyone can understand

Chai Cheatsheet: <https://devhints.io/chai>

Questions?

