COURSEWARE

- Professional Skills
- Agile Fundamentals
- Jira
- Git
- Databases Introduction
- Java Beginner
- Maven
- Testing (Foundation)
- Java Intermediate
- HTML
- CSS
- Javascript
- Spring Boot
- Selenium
- Sonarqube
- Advanced Testing (Theory)
- Cucumber
- MongoDB
- Express
- NodeJS
- React
- Express-Testing
  - Mocha
  - Chai-HTTP
  - Istanbul
- Networking
- Security
- Cloud Fundamentals
- AWS Foundations
- AWS Intermediate

Chai-HTTP

Contents

- Overview
- Installation
- Importing
- Making requests
- HTTP Methods
- Testing with Mocha
- Tutorial
- Exercises

Overview

Chai plugin for integration-testing web-applications.

Installation

```
npm i chai-http
```

Importing

```
const chai = require('chai');
const chaiHttp = require('chai-http');
chai.use(chaiHttp);
```

Making requests

When using **chai** requests can be made using either an absolute URL

```
chai.request('localhost:4494').get('/hello');
```

or by passing in your Express application

```
const app = require('./server');

chai.request(app).get('/hello');
```

The second method tends to be easier as it removes the chance of our tests breaking just because the program is running on a new port - it also allows chai to spin up a new server using `app.listen()` if one isn't already running.

HTTP Methods

The different methods can be accessed by simply calling the function with the same name:

```
chai.request(app).get('/hello');

chai.request(app).put('/replace');

chai.request(app).post('/create');

chai.request(app).patch('/update');

chai.request(app).delete('/remove');
```

You can provide a request body using `send()`

Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

```
chai.request(app).post('/create').send({'key': 'value'});
```

and query parameters using `query()`

```
chai.request(app).put('/update').query({'key': 'value'});
```

Testing with Mocha

The result of the request can be retrieved using `end()` - this is an *asynchronous* function so to signal the end of the test it is necessary to use the `done()` function provided by **Mocha** tests.

```
chai
  .request(app)
  .get('/hello')
  .end(function(error, response) {
    expect(err).to.be.null;
    expect(response).to.have.status(200);
    expect(response).to.have.body('Hello, World!');
  });
```

For the above code to work, you need to ensure you have the command `const expect = chai.expect`

Tutorial

There is no tutorial for this module.

Exercises

1. Import the Chai Library and test the CRUD functionality for your application.