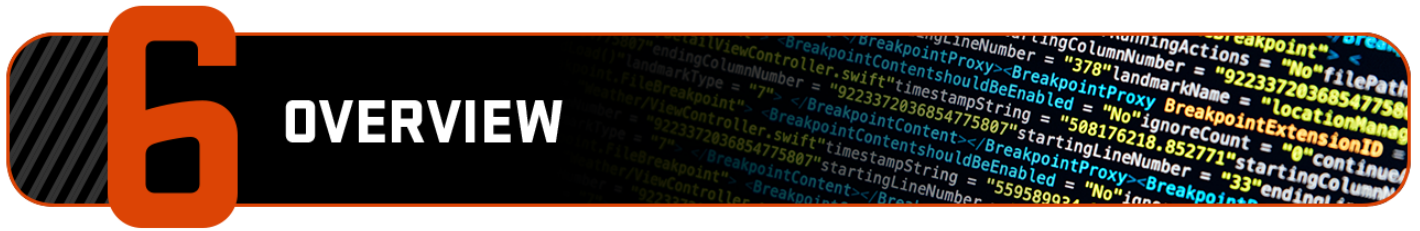


Module 6 — Overview



Introduction

In this module, we return to JavaScript and Express to study additional features which are important for developing web applications. We start with motivating the need using asynchronous programming techniques when developing web applications. We will then study how to write asynchronous code in JavaScript. Following that we look at how JavaScript provides supports for creating modules. We finish the exploration by studying the concept of middleware functions in Express, and how we can write and use middleware functions in our Express apps.

Module Learning Outcomes

After successful completion of this module, you should be able to do the following (in addition to answering the questions listed below)



1. What is synchronous vs. asynchronous programming? (CLO 1, CLO 2, CLO 3)
 - How are synchronous functions executed and how does that different from the execution of asynchronous functions?
 - Why should we avoid calling long running synchronous functions in our JavaScript code?
2. How to write asynchronous JavaScript programs using promises? (CLO 1, CLO 2, CLO 3)
 - What are promises?
 - What are the different states a promise can be in?
 - What does it mean for a promise to be settled?
 - When does it mean for a promise to be fulfilled and resolve to a value?
 - When does it mean for a promise to be rejected?
 - How can you obtain the result for a fulfilled promise?
 - How should you handle a rejected promise?
 - How can you chain promises?
3. How to write asynchronous JavaScript programs using the `await` and `async` keywords? (CLO 1, CLO 2, CLO 3)
 - How can you use the `await` keyword to wait for a promise to be settled?
 - Why can the `await` keyword only be used in an `async` function?

- What are the different ways a calling function can get a value from an `async` function?
4. How is the concept of modules supported in JavaScript? (CLO 1, CLO 2)
- Why is the concept of modules useful?
 - How do we export features from ES modules?
 - What is meant by named exports and what are default exports in ES modules?
 - How can we import and use the features exported by other modules?
 - What is the use of the `require` function when using CommonJS modules?
5. What is Express middleware and what are its uses? (CLO 1, CLO 2)
- What is Express middleware?
 - How can we add middleware using the `app.METHOD` API?
 - How can we add middleware using the `app.use` API?
 - What is meant by middleware pipeline?
 - What is the use of the function `next` in the middleware pipeline?
 - What is error-handling middleware and how does it differ from other middleware?
6. What is the Express API for HTTP requests and responses? (CLO 2)
- How can we get the query parameters from the request object using `req.query`?
 - How can we get the body of the request from the request object using `req.body`?
 - How can we get the request headers from the request object using `req.headers`?
 - How can we set response headers using `res.set`?
 - How can we set the status code of the response using `res.status`?
 - How can we send back a response body using `res.send`?
 - How can we send back JSON in the response by using `res.json`?
7. What is the use of the npm packages `dotenv`, `node-fetch` and `express-async-handler`?

Explorations

Use the pages within this module to explore the following concepts:

- **[Exploration — Asynchronous Programming: Motivation](https://canvas.oregonstate.edu/courses/1879154/pages/exploration-asynchronous-programming-motivation)**
(<https://canvas.oregonstate.edu/courses/1879154/pages/exploration-asynchronous-programming-motivation>)
- **[Exploration — Writing Asynchronous Code](https://canvas.oregonstate.edu/courses/1879154/pages/exploration-writing-asynchronous-code)**
(<https://canvas.oregonstate.edu/courses/1879154/pages/exploration-writing-asynchronous-code>)
- **[Exploration — Modules in JavaScript](https://canvas.oregonstate.edu/courses/1879154/pages/exploration-modules-in-javascript)**
(<https://canvas.oregonstate.edu/courses/1879154/pages/exploration-modules-in-javascript>)
- **[Exploration — Express Middleware](https://canvas.oregonstate.edu/courses/1879154/pages/exploration-express-middleware)**
(<https://canvas.oregonstate.edu/courses/1879154/pages/exploration-express-middleware>)
- **[Exploration — Express API for HTTP & Some Useful npm Packages](https://canvas.oregonstate.edu/courses/1879154/pages/exploration-express-api-for-http-and-some-useful-npm-packages)**
(<https://canvas.oregonstate.edu/courses/1879154/pages/exploration-express-api-for-http-and-some-useful-npm-packages>)

Required Reading

Read the following chapters from the textbook.

- Chapter 9. Read sections 9.1, 9.4, 9.5, 9.6, 9.9, 9.10, 9.12.
- Chapter 10. You may skip section 10.5.

Read the following articles from the Express website.

- **Using Middleware** [_\(https://expressjs.com/en/guide/using-middleware.html\)_](https://expressjs.com/en/guide/using-middleware.html)
- **Writing Middleware** [_\(https://expressjs.com/en/guide/writing-middleware.html\)_](https://expressjs.com/en/guide/writing-middleware.html)

Task List

Please make sure to complete the following assignments and other tasks:

- Complete the exercises provided in the Explorations.
- Take the Module quiz.
- Complete Assignment 5.
- Complete the Module Feedback Survey.

 **Return to Modules** [_\(https://canvas.oregonstate.edu/courses/1879154/modules\)_](https://canvas.oregonstate.edu/courses/1879154/modules)