Express: Advanced Topics

So, in this section, you learned that:

- A middleware function is a function that takes a request object and either terminates the request/response cycle or passes control to another middleware function.
- Express has a few built-in middleware functions:
  - **json()**: to parse the body of requests with a JSON payload
  - **urlencoded()**: to parse the body of requests with URL-encoded payload
  - **static()**: to serve static files
- You can create custom middleware for cross-cutting concerns, such as logging, authentication, etc.

```
// Custom middleware (applied on all routes)
app.use(function(req, res, next)) {
    // …
    next();
}
```

```
// Custom middleware (applied on routes starting with /api/admin)
app.use('/api/admin', function(req, res, next)) {
    // …
    next();
}
```

- We can detect the environment in which our Node application is running (development, production, etc) using **process.env.NODE_ENV** and **app.get('env').**

- The **config** package gives us an elegant way to store configuration settings for our applications.
- We can use the **debug** package to add debugging information to an application. Prefer this approach to console.log() statements.
- To return HTML markup to the client, use a templating engine. There are various templating engines available out there. Pug, EJS and Mustache are the most popular ones.