Using template engines with Express

A *template engine* enables you to use static template files in your application. At runtime, the template engine replaces variables in a template file with actual values, and transforms the template into an HTML file sent to the client. This approach makes it easier to design an HTML page.

The Express application generator uses Pug as its default, but it also supports Handlebars, and EJS, among others.

To render template files, set the following application setting properties, in the default app.js created by the generator:

- views, the directory where the template files are located. Eg: app.set('views', './views'). This defaults to the views directory in the application root directory.
- view engine, the template engine to use. For example, to use the Pug template engine: app.set('view engine', 'pug').

Then install the corresponding template engine npm package; for example to install Pug:

```
$ npm install pug --save
```

Express-compliant template engines such as Pug export a function named __express(filePath, options, callback), which res.render() calls to render the template code.

Some template engines do not follow this convention. The @ladjs/consolidate library follows this convention by mapping all of the popular Node.js template engines, and therefore works seamlessly within Express.

After the view engine is set, you don't have to specify the engine or load the template engine module in your app; Express loads the module internally, for example:

```
app.set('view engine', 'pug')
```

Then, create a Pug template file named index.pug in the views directory, with the following content:

```
html
  head
    title= title
  body
    h1= message
```

Create a route to render the index.pug file. If the view engine property is not set, you must specify the extension of the view file. Otherwise, you can omit it.

```
app.get('/', (req, res) => {
  res.render('index', { title: 'Hey', message: 'Hello there!' })
})
```

When you make a request to the home page, the index.pug file will be rendered as HTML.

The view engine cache does not cache the contents of the template's output, only the underlying template itself. The view is still re-rendered with every request even when the cache is on.