Представления и статика

Этот материал имеет отношение к веб-приложениям и, в частности, Express

Представление (**view**) - это динамическое генерируемое нечто, скорее всего по шаблону.
Статика - это файлы рисунков, и прочее.

Термин **view** имеет отношение к паттерну проектирования приложений MVC.

Задача - максимально отделить одно от другого.

Статика может быть выдана сразу (как бы режим файл-сервера), а может появиться "в составе" представления.

Что значит написать в Express фразу типа
res.render('about') ?

Это и значит что мы вызываем **представление**:

1. должен быть выбран механизм шаблонов
2. создан файл типа views/layouts/main.handlebars - в нём есть {{{body}}} и т.п. т.е. placeholder для подстановки HTML-кода конкретного представления
3. создан файл типа views/about.handlebars - т.е. конкретное представление для данного маршрута /about; там уже код HTML, который будет подставлен вместо плейсхолдера {{{}}}
4. В случае handlebars, к примеру, в представлении есть плейсхолдеры своего уровня - с двойными фигурными скобками - это заместители JavaScript-идентификаторов, туда подставляются значения при вызове render

Например, если в представлении написано:
** {{name}} **

То это значит, что для подстановки в него конкретного имени
нужно будет написать

```
app.get('/about',(req, res)=>{
    res.render('about', { name: 'Ilya' });
});
```

Т.е. в {{}} указывается имя параметра из JSON-записи, которая указывается в методе render
… который указывается в коде, который обрабатывает данный маршрут.

Ну а статика - это middleware (см. на следующей странице)

Некий каталог регистрируется как источник для поиска в нём запрошенных статических файлов
(фактически это как бы такое graceful встраивание функциональности традиционного веб-сервера, отдающего файлы)
с помощью следующего **middleware**:
app.use(express.static(__dirname + '/public'));
или
более *цивилизованно*
app.use(express.static(path.join(__dirname, 'public')))
тут мы абстрагируем вульгарную конкатенацию, поручая модулю path сформировать валидный путь
(path = require('path'))

Тогда попытка клиента обратиться к маршруту типа /logo.png
приведёт к поиску этого файла в папке public,
которая находится в корне папки приложения

Это то, что сервера типа Apache делают автоматом, "не задумываясь".

Middleware static фактически устанавливает отображение, которое абстрагирует от пользователя "механизм порождения или возникновения" того, что запрошено. Нет возможности узнать, существовал ли файл logo.png или был как-то сгенерирован на лету, и, если существовал, то где именно.

Поэтому статика - частный, упрощённый случай представления.