

шаблонизаторы

общая идея

фиксируется каркас из захардкоженной разметки

те места, которые "сменные", заменяются на "гнезда" или "placeholders"

которые традиционно обозначаются фигурными скобками, обычно двумя

и далее схема: вызываем метод **render**, которому передаём имя **представления (view, вьюха)**

и JSON Options, где перечислены пары ключ:значение как имяГнезда:значениеПодстановки

представление - это динамическое генерируемое нечто, скорее всего по шаблону

handlebars	pug (ранее jade)				
должен быть <i>выбран</i> в качестве механизм шаблонов	шаблонизатор по умолчанию http://html2jade.org/				
npm i -S express-handlebars let handlebars = require('express-handlebars') .create({ defaultLayout:'main' }); app.engine('handlebars', handlebars.engine); app.set('view engine', 'handlebars');	npm i -S pug устанавливаем, далее он сам подцепляет				
<div><div><i>макет - меташаблон, шаблон вьюшек</i> файл типа views/layouts/main.handlebars - в нём есть {{{body}}} и т.п. т.е. placeholder для подстановки тегов для представлений If you don't want Handlebars to escape a value, use the "triple-stash", {{{.</div><div><i>представление, вьюшка - шаблон страницы</i> файл типа views/about.handlebars - т.е. конкретика для маршрута; там HTML и {{..}} куда подставляются не-теги</div></div> <div>дефолтный макет используется всеми вьюшками если не указано иное</div> <table><tr><td>views/layouts/main.handlebars</td><td>views/root.handlebars</td></tr><tr><td><!DOCTYPE html> <html> <head> <title>Express Site</title><meta charset="utf-8"> </head> <body> {{{body}}} </body> </html></td><td><h1>Welcome, friends!</h1> <p>You are at {{text}} page!</p></td></tr></table>	views/layouts/main.handlebars	views/root.handlebars	<!DOCTYPE html> <html> <head> <title>Express Site</title><meta charset="utf-8"> </head> <body> {{{body}}} </body> </html>	<h1>Welcome, friends!</h1> <p>You are at {{text}} page!</p>	<i>вот простая вьюха</i> views/main.pug html head title= title body h1= title p Hello everyone... a(href='/') Homepage!
views/layouts/main.handlebars	views/root.handlebars				
<!DOCTYPE html> <html> <head> <title>Express Site</title><meta charset="utf-8"> </head> <body> {{{body}}} </body> </html>	<h1>Welcome, friends!</h1> <p>You are at {{text}} page!</p>				
app.get('/:key', (req, res) => { res.render('root', {text: req.params.key} }); });	подключаем app.get('/:key', (req, res) => { res.render('main.pug', {title: 'page ' + req.params.key + ' is in development!'}); }); `page \${req.params.key} ...`				

без шаблона	с шаблоном
<code>res.status(404).send('<h2>Not yet here, pardon!</h2>');</code>	<code>res.status(404).render('404.pug',{....});</code>

Выше мы видели, что макет забирает в свой `{{body}}` код из представлений. Очевидно, может быть нужно менять также и другие разделы, вроде `head`.

server.js	
<pre>handlebars = require('express-handlebars') .create({ defaultLayout:'main', helpers: { section: function(name, options){ if(!this._sections) this._sections = {}; this._sections[name] = options.fn(this); return null; } } }); get('/:key', (req, res) => { res.render('root', {text: req.params.key}); });</pre>	
views/layouts/main.handlebars	views/root.handlebars
<pre><!DOCTYPE html> <html> <head> {{{_sections.head}}} </head> <body> {{{body}}} </body> </html></pre>	<pre>{{#section 'head'}} <title>Root view: {{text}}</title> <meta charset="utf-8"><meta name="author" content="Alan Turing"> {{/section}} <h1>Welcome, friends!</h1> <p>You are at {{text}} page!</p></pre>