

Синтаксис

Запуск консоли (REPL)

Запустить сервер

остановить сервер

Подключить пакет

Выполнить функцию подключения пакета

npm

Установка пакета

```
npm install пакет
```

Установка с сохранением в dependencies (package.json)

```
npm install пакет --save
```

ejs

Установка ej

```
npm install ej --save
```

Синтаксис

Весь код пишется как html, куски js-кода вставляются в конструкции:

`<%= код %>` - когда получаемое значение выводится в html (например вывод переменных)

`<% код %>` - когда код в конструкции не выводится (условия, циклы и т.п.)

Условие

```
<% if(условие) { %>
  <p>Выведется, только если выполняется условие</p>
<% } %>
```

Цикл

```
<% for(var i=0;i< posts.length ;i++) { %>
  <div><%= posts[ i ].content %></div>
<% } %>
```

ejs (cont)

Передать переменные в ej

```
res.render("файл.ejs", {переменные});
```

- второй параметр - объект

Шаблоны (partials)

Хранятся в папке "views/partials"

```
Пример шаблона header.ejs: <!DOCTYPE html>
<html>
  <head>
```

Включить шаблон

```
<% include partials/ header %>
```

package.json

package.json хранит метаданные проекта

В разделе 'dependencies' хранятся используемые модули и версии

Сгенерировать package.json: `npm init`

entry point - начальный файл (обычно `app.js`)

пропустить пункт - нажать enter

Express

Подключение express

```
var express = require('express');
var app = express();
```

- сохр. в переменную `app`

Routes

обрабатывают http-запросы

Пример:

```
app.get("/", function(req, res) {
  res.send("text");
});
```

`req, res` - передаем объекты запроса и ответа

Ожидать запросы

```
app.listen(порт, process.env.PORT || 3000, function() {
  // ...
});
```

Порт обычно 3000 или `process.env.PORT`

Express (cont)

Остальные запросы (404)

Так как выполняется первый подходящий to route для 404 должен идти последним

Редирект

```
res.redirect("/путь");
```

Параметры (переменные)

```
app.get("r/:post/id/:title", function(req, res) {
  // ...
});
```

:post и :title - параметры

Доступ к параметру

```
var параметр = res.params.paraмeтp;
// ...
res.render("файл", {параметры});
```

Параметр в число

```
Number(res.params.paraмeтp);
```

Ответ

```
.send() - ответ, выполняется один раз
.render("файл") - ответ файлом (ejs/html)
```

Расширение файлов для render

```
app.set("view engine", "ejs");
```

функции `render` можно будет указывать то имя файла, без расширения (ejs)

Стили и скрипты

css и js файлы хранить в папке `public` (можно в подпапках)

Папку нужно подключить в `app.js`: `app.use(express.static("public"))`; - в `app.js` файлов не нужно будет указывать `public` ('`main.js`', '`css/app.css`')

POST

```
app.post("/url", function(req, res) {
  // ...
});
```

Получить атрибут name тег input (POST)

```
req.body.input_name; - для доступа к body нужен пакет body-parser
```

Express (cont)

Подключение body-parser

```
var bodyParser = require("body-parser");  
app.use(bodyParser.urlencoded({extended: true}));
```



By **Yur375**

cheatography.com/yur375/

Not published yet.

Last updated 24th October, 2018.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>