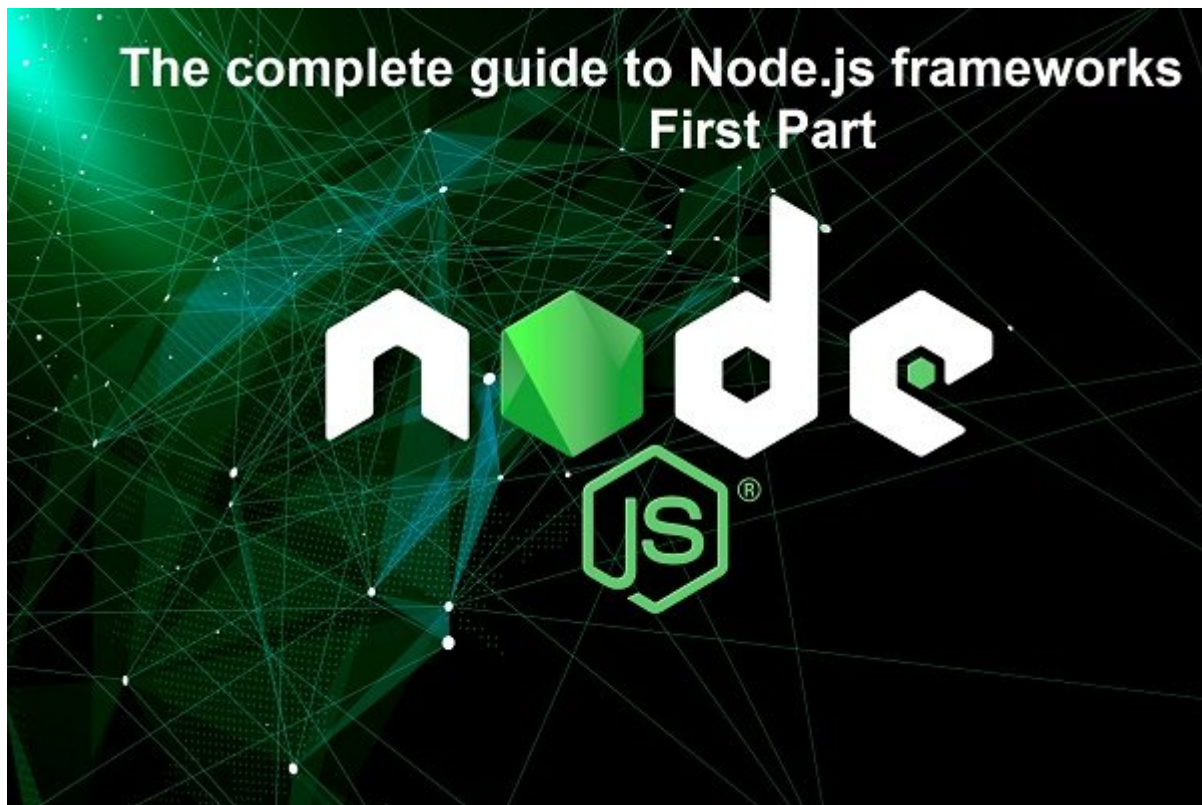


[Node.js 網路開發系列書籍目錄](#)



本書是「Node.js 網路開發」系列的第一本，介紹 Node.js 的基礎知識。Node.js 是一個基於 V8 引擎的 JavaScript 執行環境，它允許在伺服器端執行 JavaScript。Node.js 的架構是 non-blocking 的，這使得它能夠處理大量的並行請求。本書將介紹 Node.js 的安裝、配置、以及如何使用 Express 和 Sails.js 等框架來開發網路應用程式。此外，本書還將介紹 REST API 的開發，以及如何使用 Meteor 和 LoopBack 等框架來開發網路應用程式。最後，本書還將介紹 Node.js MVC 的開發。

本書是「Node.js 網路開發」系列的第一本，介紹 Node.js 的基礎知識。Node.js 是一個基於 V8 引擎的 JavaScript 執行環境，它允許在伺服器端執行 JavaScript。Node.js 的架構是 non-blocking 的，這使得它能夠處理大量的並行請求。本書將介紹 Node.js 的安裝、配置、以及如何使用 Express 和 Sails.js 等框架來開發網路應用程式。此外，本書還將介紹 REST API 的開發，以及如何使用 Meteor 和 LoopBack 等框架來開發網路應用程式。最後，本書還將介紹 Node.js MVC 的開發。

Node.js における MVC パターン

MVC (Model-View-Controller) 是软件设计中的一种设计模式，用于组织应用程序的结构。它通常分为三个部分：Model（模型）、View（视图）和 Controller（控制器）。Model 负责处理数据，View 负责显示数据，Controller 负责处理用户输入并协调 Model 和 View 的操作。

MVC 模式在许多编程语言和框架中都有实现。例如，在 Ruby 语言中，Rails 框架就采用了 MVC 模式。在 JavaScript 语言中，Node.js 和 Express.js 框架也采用了 MVC 模式。在 Python 语言中，Django 框架也采用了 MVC 模式。

MVC 模式的优势在于它能够将应用程序的不同部分分离开来，使得代码更加模块化、易于维护和扩展。同时，它也能够提高代码的可测试性，因为每个部分都可以独立地进行测试。

□ □ □ □ □ □ □ □ □ □ □ □



☐☐☐☐☐☐☐☐ ☐☐☐☐☐☐ ☐☐☐☐ ☐☐

Node.js

Node.js 框架 Sinatra 框架 MVC 框架

Diet

API 的 数据 格式 是 基于 Node.js 的 数据 格式 Diet . 使用 Express 的 数据 格式 Diet 的 数据 格式 . 使用 数据 格式

```
// Create an app
var server = require('dier')
var app = server()
app.listen('http://localhost:8000')

// When http://localhost:8000/ is requested, respond with "Hello World!"
app.get('/', function($) {
  $.end('Hello World!')
})
```

1

000000 00000000 00 000000 0000000000 00 000000 00000000 00 0000 0000 00 000 0000000000 00000000 00 Diet
 0000 00 000 0000 0 000000 0000000000 000000 0000000000 00000000 000000 00 Diet 0000 .00000 0000000000

```
// Register middleware functions for the upload path
app.post('/upload/picture', upload, crop, save, finish)
```

Express

이 단락을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 . Nodejitsu Node 이 책을 읽는 동안 Flatiron 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 . 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 . 이 책을 읽는 동안 Sinatra 이 책을 읽는 동안 Flatiron . 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 http 이 책을 읽는 동안 (require) . 이 책을 읽는 동안 Broadway 이 책을 읽는 동안 Flatiron 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 API 이 책을 읽는 동안 Connect 이 책을 읽는 동안 Union . Node MVC 이 책을 읽는 동안 http 이 책을 읽는 동안 Union .

```
// create a flatiron application

var flatiron = require('flatiron'),
    app = flatiron.app;

app.use(flatiron.plugins.http);

app.router.get('/', function () {
  this.res.writeHead(200, { 'Content-Type': 'text/plain' });
  this.res.end('Hello world!\n');
});

app.start(8080);
```

4 이 책을 읽는 동안

Hapi

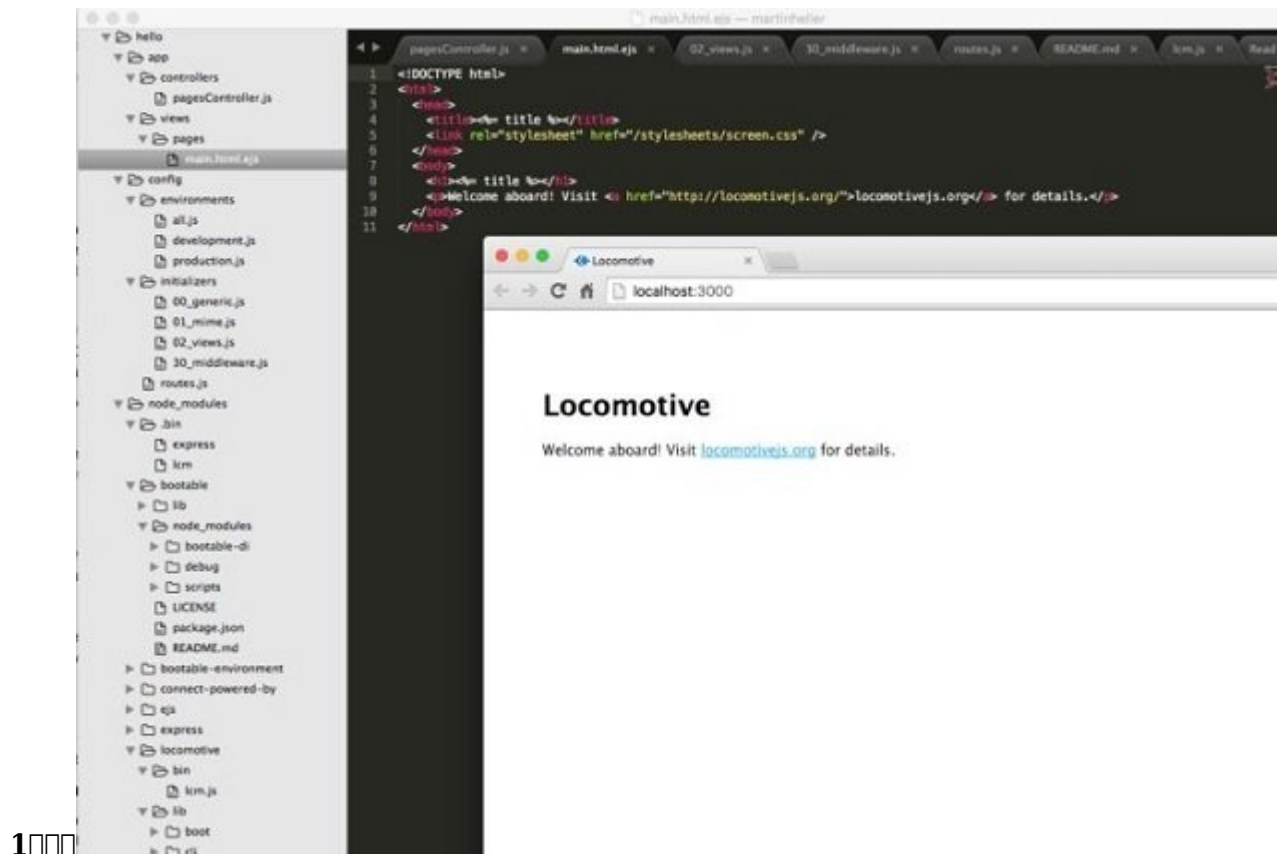
이 단락을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 Hapi 이 책을 읽는 동안 Hapi . 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 이 책을 읽는 동안 Express 이 책을 읽는 동안 Hapi . 이 책을 읽는 동안 Hapi 이 책을 읽는 동안 Hapi . 이 책을 읽는 동안 Hapi .

```
// create a hapi server

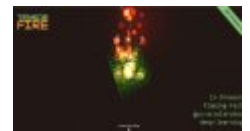
var Hapi = require('hapi');
var server = new Hapi.Server(3000);

server.route([
  {
    method: 'GET',
    path: '/api/items',
    handler: function(request, reply) {
      reply('Get item id');
    }
  }
]);
```


Express.js 是 Node.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。Express.js 使用 Jade 模板引擎，而 Locomotive 是一个基于 Express.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。Express.js 使用 Jade 模板引擎，而 Locomotive 是一个基于 Express.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。



Express.js 是 Node.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。



Express.js 是 Node.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。

Express.js 是 Node.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。

Total.js

Express.js 是 Node.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。Total.js 是一个基于 Express.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。Total.js 是一个基于 Express.js 的 Web 应用框架，它提供了许多功能，如路由、中间件、模板引擎等。

```
require('total.js');
```

```
F.route('/', function() {  
  this.plain('total.js is really good!');  
});
```


Kraken

[illegible]

Lusca (🐙🐙🐙🐙) 🐙 Kappa (NPM Proxy) 🐙
 Makara (LinkedIn Dust.js i18N) 🐙
 Adaro (LinkedIn Dust.js Templating).

ꠘꠕꠘ Locomotive ꠘꠕꠘ .ꠘꠕꠘ ꠘꠕꠕꠘ yo ꠘꠕ ꠘꠕꠕꠕꠕꠕ ꠘꠕꠕꠘ ꠘꠕꠕꠘ Kraken ꠘꠕꠕꠕꠕꠕ ꠘꠕꠕꠕꠘ 2 ꠘꠕꠘ ꠘꠕ ꠘꠕꠕꠘ ꠘꠕꠕꠘ
ꠘꠕꠕꠕꠘ ꠘꠕꠕꠕꠕꠕꠕꠘ Rails ꠘꠕꠕꠘ ꠘꠕꠕꠕꠕꠘ ꠘꠕ ꠘꠕꠘ ꠘꠕꠕꠕꠕꠕꠘ ꠘꠕꠘ ꠘꠕꠕꠕꠕꠕꠘ

```
$ yo kraken

      ,mm\`
hh  /  _  _  \
    |(@)(@)|   Release the Kraken!
    )  _  (
    /,'))((`.\
  (( (( )) ))
  `\' `)( ' /'

Tell me a bit about your application:

[?] Name: HelloWorld
[?] Description: A test kraken application
[?] Author: YourName GoesHere
...
```

2 □□□

Meteor

[illegible]

3 □□□

Nodal

[illegible]

++++C Napa.js

Node.js

Sails

[illegible]

ThinkJS



(○○○ ○○○) Node.js ○○○○○○○○○○○○○○○○○○○

□□□□□□□□□□ - MVC □□□ □□□□ - Node.js □□□□ MVC □□□□□□□□□□ - □□□ □□□□ - Node.JS

<https://www.shabakeh-mag.com/workshop/10740/%D8%B1%D8%A7%D9%87%D9%86%D9%85:%D8%A7%DB%8C-%D8%AC%D8%A7%D9%85%D8%B9-%DA%86%D9%87%D8%A7%D8%B1%DA%86%D9%88%D8%A8%E2%80%8C%D9%87%D8%A7%DB>

[%8C-%D8%A8%D8%B1%D8%AA%D8%B1-nodejs](#)