

7. Express.js

Untuk memudahkan pembuatan aplikasi web anda bisa menggunakan framework [ExpressJS](#) daripada harus menggunakan module http bawaan Node.js. Framework ini menawarkan beberapa fitur seperti routing, rendering view dan mendukung middleware dengan kata lain anda akan banyak menghemat waktu dalam pengembangan aplikasi Node.js.

ExpressJS merupakan framework minimal yang sangat fleksibel. Anda bisa membuat web server HTML, server file statik, aplikasi chat, search engine, sosial media, layanan web dengan akses melalui REST API atau aplikasi hybrid yaitu selain pengguna mempunyai akses melalui REST API juga mempunyai akses ke HTML page.

7.1. Server File

Sebelumnya bikin project kecil dengan mengetikkan perintah berikut pada direktori project

```
$ npm init
```

dan berikan nama project sebagai `server-file-statik`. Direktori project dari server file bisa dilihat pada susunan tree dibawah ini.

Susunan file dan direktori

```
server-file-statis
├── package.json
├── app.js
├── node_modules
├── publik
│   └── index.html
```

Dengan adanya npm instalasi ExpressJS sangat mudah, ketik perintah berikut pada direktori project.

```
$ npm install express --save
```

Catatan :

Perintah diatas akan menginstall ExpressJS dengan versi yang paling terbaru (pada saat buku ini ditulis versi terbaru adalah 4.x). Jika membutuhkan versi tertentu cukup dengan menambahkan '@' dan nomer versi yang akan di inginkan seperti contoh berikut

```
$ npm install express@3 --save
```

Untuk kedepannya bahasan mengenai ExpressJS ini akan memakai versi 4.x.

Kode

Jika anda ingat server file yang memakai modul http pada bab sebelumnya berikut merupakan versi yang memakai ExpressJS

app. js

```
'use strict';
```

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

```
var server = express();
```

```
var logger = require('morgan');
```

```
server.use(logger('dev'));
```

```
server.use(express.static(__dirname+'/publik'));
```

```
server.listen(4000, function(){
```

```
  console.log('Server file sudah berjalan bos!');
```

```
});
```

Seperti yang dijelaskan pada bab sebelumnya untuk memakai module Node.js di gunakan keyword `require`.

Modul `express` akan menangani tiap request dari user dan kemudian akan memberikan response berupa file yang diinginkan. Pada kode diatas file yang akan diberikan ke pengguna disimpan pada folder `publik`.

7.1.1. Middleware

Fungsionalitas yang diberikan oleh ExpressJS di bantu oleh yang namanya `middleware` yaitu fungsi asinkron yang bisa mengubah request dan respon di server.

Pada kode diatas contoh middleware yaitu modul `morgan`. Cara pemakaian `middleware` yaitu melalui `app.use()`.

Module `morgan` merupakan modul untuk logger yang berfungsi untuk pencatatan tiap request ke server. Pencatatan ini atau istilahnya `logging` akan ditunjukkan di console terminal.

Untuk menginstall modul ini ketik perintah berikut

```
$ npm install morgan --save
```

Middleware `middleware` ini bisa anda lihat di link berikut

<https://github.com/senchalabs/connect#middleware>

7.1.2. Akses Server

Untuk menjalankan server file statik ini

```
$ node app.js
```

Anda bisa meletakkan file apa saja pada direktori 'publik'. Untuk mengakses server ini ketik alamat berikut pada browser

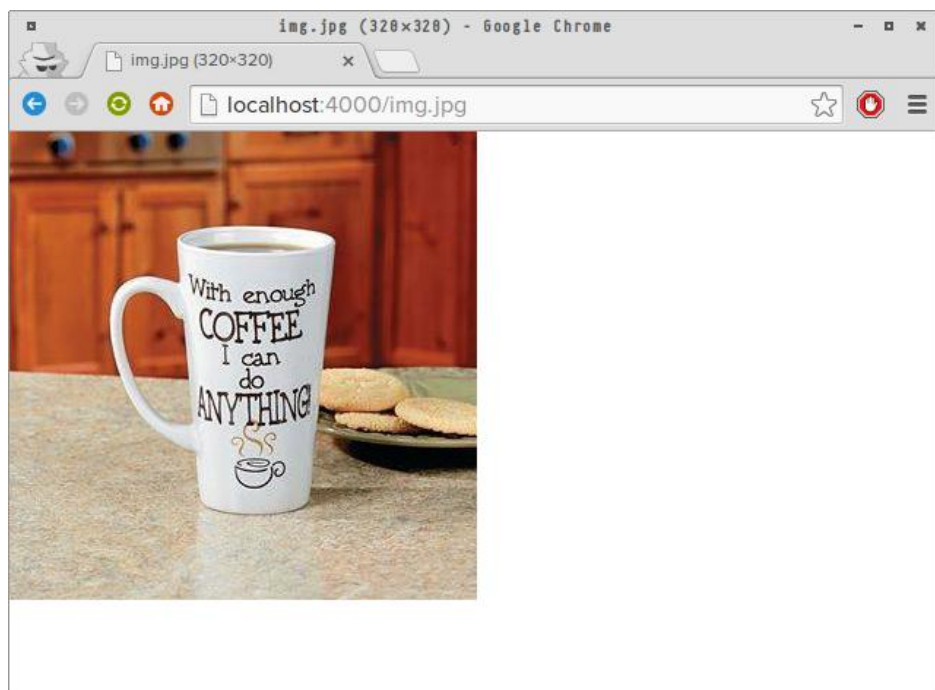
```
http://localhost:4000
```

dan secara default akan menampilkan file `index.html` di browser.

Untuk mengakses file yang lain format URL adalah

```
http://localhost:4000/[nama-file]
```

misalnya untuk mengakses file `img.jpg`



7.2. Server REST

Framework ExpressJS sangat banyak digunakan sebagai aplikasi RESTful. Bagi anda yang belum tahu, REST atau Representational State Transfer adalah arsitektur yang digunakan dalam desain aplikasi network. Identya yaitu daripada memakai teknik seperti CORBA, SOAP atau RPC untuk membuat Web Service lebih baik jika memakai protokol HTTP yang lebih mudah.

Aplikasi RESTful memakai request HTTP untuk operasi Create, Read, Update dan Delete data. REST itu sendiri bukanlah suatu standard jadi tidak ada yang namanya spec dari W3C. Sehingga sangat mudah bagi bahasa pemrograman untuk menggunakan arsitektur ini. Keuntungan lainnya yaitu dengan arsitektur ini bisa dibangun berbagai macam teknologi klien seperti web atau mobile ataupun dekstop untuk mengakses aplikasi RESTful tersebut.

Untuk contoh aplikasi RESTful memakai ExpressJS akan diberikan pada Bab [Person REST API](#).