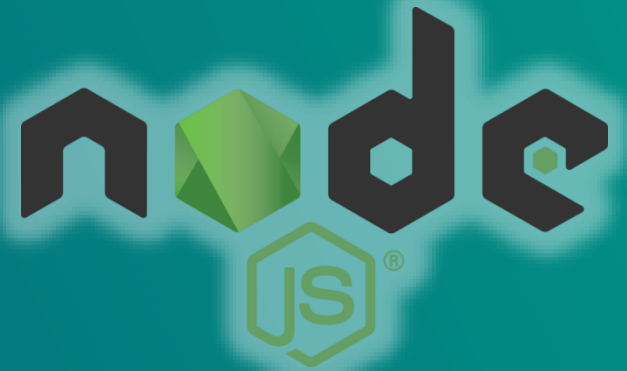


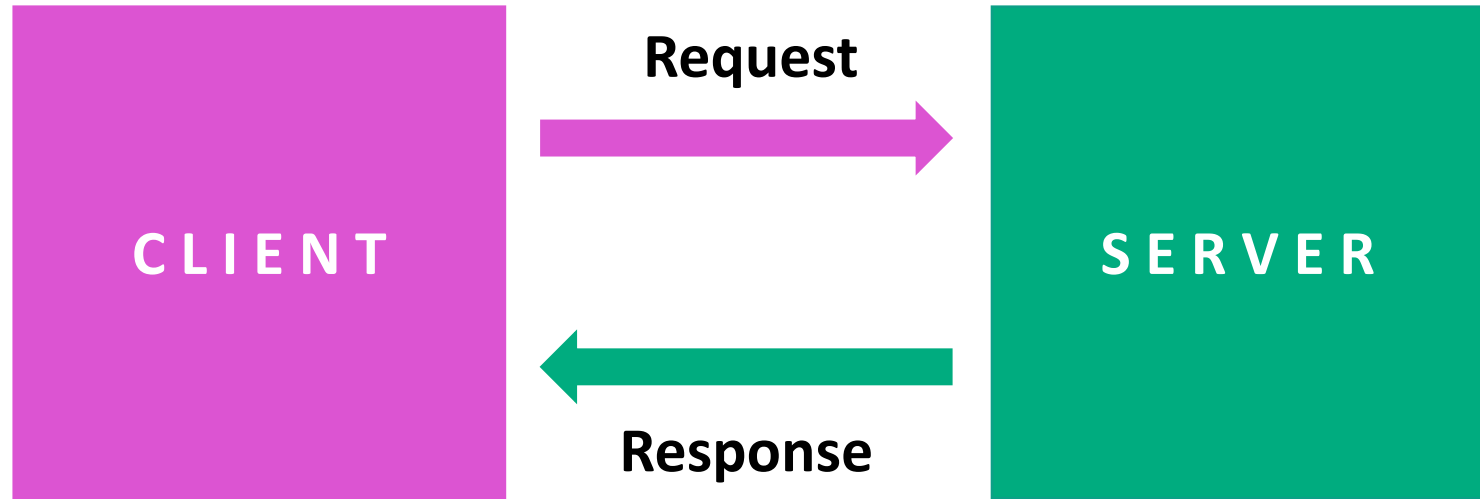
Back-End Development

Exploring

#3 HTTP Module



Clients & Servers



When we surf in a website sometimes we do something, for instance click a button, it will request data from the server & we'll see its response, like opening a new webpage.

Client & server communicate each other, using *protocols*: a set of communication rules, that both client & server agree to use when communicating.

HTTP Methods

- **HTTP** (*The Hypertext Transfer Protocol*) is designed to enable communications between clients & servers. It works as a request & response protocol between a client & server.
- A web browser may be the client, and an application on a computer that hosts a web site may be the server.
- The most commonly used HTTP Methods are POST, GET, PUT, PATCH & DELETE.

Create a Server

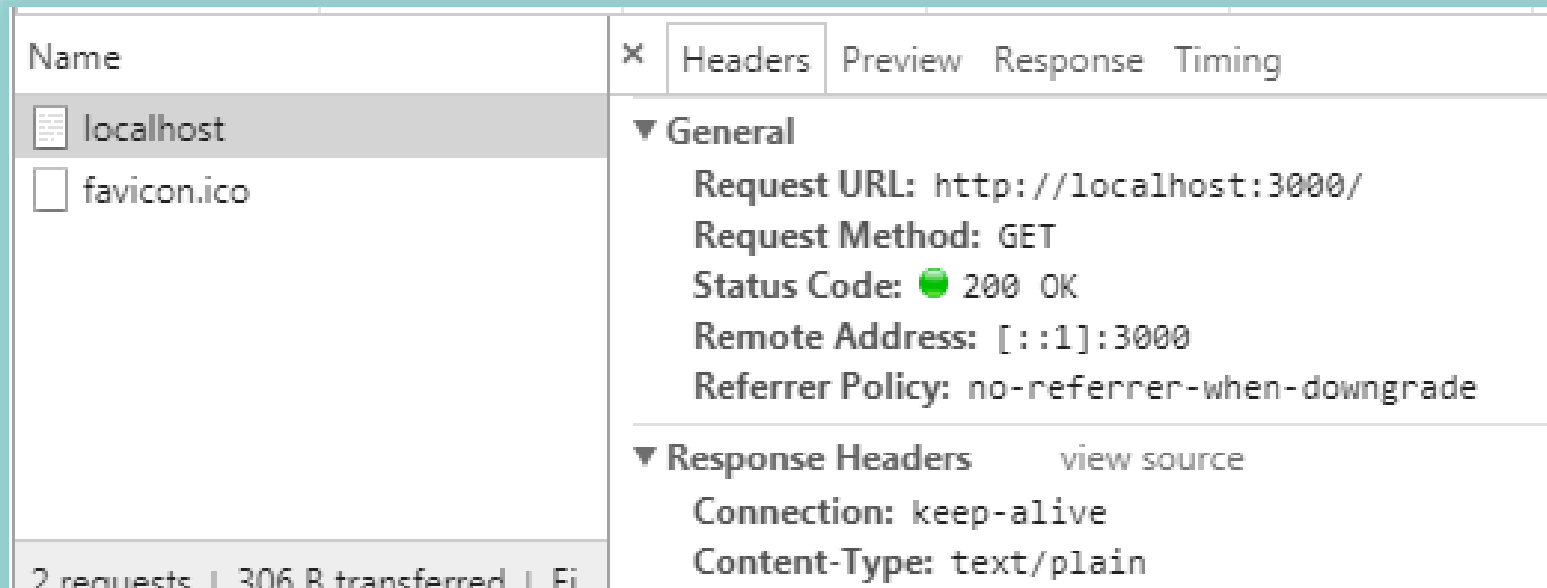
```
var http = require('http');
```

```
var server = http.createServer(function(req, res){  
    res.writeHead(200, {'Content-Type' :  
    'text/plain'});  
    res.end('Halo semuanya!');  
});
```

```
server.listen(3000);
```

```
// buka di browser = localhost : 3000
```

```
console.log('Server aktif. Listening port 3000.');
```



Create a Server

set url

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

```
var server = http.createServer(function(req, res){  
    res.writeHead(200, {'Content-Type' : 'text/plain'});  
    res.end('Halo semuanya!');  
});
```

```
server.listen(1234, '127.127.127.127');
```

```
// buka di browser = 127.127.127.127 : 1234
```

```
console.log('Server aktif. Listening port 1234.');
```

Name	×	Headers	Preview	Response	Timing
127.127.127.127					
favicon.ico					

2 requests | 306 B transferred | Fi...

▼ General

Request URL: http://127.127.127.127:1234/
Request Method: GET
Status Code: 200 OK
Remote Address: 127.127.127.127:1234
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers [view source](#)

Connection: keep-alive
Content-Type: text/plain
Date: Sat, 02 Dec 2017 13:09:32 GMT

Create a Server

set response

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

var server = http.createServer(function(req, res){
    res.writeHead(404, {'Content-Type' : 'text/html'});
    res.end('<h1>Halo semuanya!</h1>');
});

server.listen(1234, '127.127.127.127');
// buka di browser = 127.127.127.127 : 1234
console.log('Server aktif. Listening port 1234.');
```

The screenshot shows a web browser's developer console with the 'Network' tab selected. The left pane shows a list of requests: '127.127.127.127' (selected) and 'favicon.ico'. The right pane shows the details for the selected request, including the 'General' tab with the following information:

- Request URL: http://127.127.127.127:1234/
- Request Method: GET
- Status Code: 404 Not Found
- Remote Address: 127.127.127.127:1234
- Referrer Policy: no-referrer-when-downgrade

Below the 'General' tab, the 'Response Headers' section is visible, showing:

- Connection: keep-alive
- Content-Type: text/html
- Date: Sat, 02 Dec 2017 13:13:10 GMT

At the bottom of the console, it indicates '2 requests | 337 B transferred | Fi...'

Create a Server

print request url

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

var server = http.createServer(function(req, res){
    console.log('Request : ' + req.url);
    res.writeHead(404, {'Content-Type' : 'text/html'});
    res.end('<h1>Halo semuanya!</h1>');
});

server.listen(1234, '127.127.127.127');
console.log('Server aktif. Listening port 1234.');
```

// buka di browser, tambahkan url target /sesuatu

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: node



Server aktif. Listening port 1234.

Request : /

Request : /favicon.ico

Request : /halo

Request : /halo/yuk

SELAMAT DATANG!

~ Ini webpage paling unyu sedunia ~


```
<!DOCTYPE html>
<html>
  <head><title>Lintang</title>
    <style>
      body{background:green; font-family:verdana;
color:white; padding:150px;}
      h1{font-size:72px; text-transform:uppercase;
letter-spacing:2px; text-align:center;}
      p{font-size:32px; text-align:center;}
    </style>
  </head>
  <body>
    <h1>Selamat Datang!</h1>
    <p>~ Ini webpage paling unyu sedunia ~</p>
  </body>
</html>
```

Serving HTML File #1 using FS readFileSync

satu.js

```
var http = require('http');
var fs = require('fs');

var x = fs.readFileSync('satu.html', 'utf8');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end(x);
});

server.listen(3000);
console.log('Server aktif. Listening port 3000.');
```

Serving HTML File #2 using FS readStream

satu.js

```
var http = require('http');
var fs = require('fs');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  res.writeHead(200, {'Content-Type': 'text/html'});

  var bacaStream
  =fs.createReadStream(__dirname+'/satu.html', 'utf8');
  bacaStream.pipe(res);
});

server.listen(3000);
console.log('Server aktif. Listening port 3000.');
```

JSON

- **JSON (*Javascript Object Notation*)** is a lightweight data-interchange format that based on a subset of JS. It's easy to read, write and can be used with any modern language.

- **JSON is not a JavaScript Object!**

JSON rules:

- Uses property/value pairs = `{"nama" : "Andi"}`
- Uses **doublequotes** on its prop & val (except number)
- Must use specified data type
- File type is **".json"**
- Mime type is **"application/json"**

Serving JSON Data

satu.js

```
var http = require('http');
var fs = require('fs');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  res.writeHead(200, {'Content-Type': 'application/json'});
  var dataku = {
    nama: 'Lintang',
    usia: 24,
    jomblo: true
  };
  res.end(JSON.stringify(dataku));
});

server.listen(3000);
console.log('Server aktif. Listening port 3000.');
```

satu.json

```
{  
  "nama": "Lintang",  
  "usia": 24,  
  "status": "jomblo"  
}
```

Serving JSON File #2

satu.js

```
var http = require('http');
var fs = require('fs');

var x = fs.readFileSync('satu.json', 'utf8');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end(x);
});

server.listen(3000);
console.log('Server aktif. Listening port 3000.');
```

Routing



/beranda



/profil



/berita



/galeri



/tentang



/kontak

home.js

Basic Routing #1 Home Page

```
var http = require('http');
var fs = require('fs');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  if(req.url === '/home' || req.url === '/'){
    res.writeHead(200, {'Content-Type': 'text/html'});
    fs.createReadStream(__dirname + '/home.html').pipe(res);
  }
});

server.listen(3000);
console.log('Server aktif, port 3000.');
```

** Buat home.html*

** Dengan cara serupa, dapat dibuat route lainnya!*

home.js

Basic Routing #2 Error 404 Not Found

```
var http = require('http');
var fs = require('fs');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  if(req.url === '/home' || req.url === '/'){
    res.writeHead(200, {'Content-Type': 'text/html'});
    fs.createReadStream(__dirname + '/home.html').pipe(res);
  } else {
    res.writeHead(404, {'Content-Type': 'text/html'});
    fs.createReadStream(__dirname + '/404.html').pipe(res);
  }
});

server.listen(3000);
console.log('Server aktif, port 3000.');
```

** Buat home.html & 404.html*

** Dengan cara serupa, dapat dibuat route lainnya!*

```
var http = require('http');  
console.log(http.METHODS);
```

HTTP Methods

```
[ 'ACL',  
  'BIND',  
  'CHECKOUT',  
  'CONNECT',  
  'COPY',  
  'DELETE',  
  'GET',  
  'HEAD',  
  'LINK',  
  'LOCK',  
  'M-SEARCH',  
  'MERGE',  
  'MKACTIVITY',  
  'MKCALENDAR',  
  'MKCOL',  
  'MOVE',  
  'NOTIFY',  
  'OPTIONS',  
  'PATCH',  
  'POST',  
  'PROPFIND',  
  'PROPPATCH',  
  'PURGE',  
  'PUT',  
  'REBIND',  
  'REPORT',  
  'SEARCH',  
  'SUBSCRIBE',  
  'TRACE',  
  'UNBIND',  
  'UNLINK',  
  'UNLOCK',  
  'UNSUBSCRIBE',  
 ]
```

Status Codes

```
var http = require('http');  
console.log(http.STATUS_CODES);
```

```
{ '100': 'Continue',  
  '101': 'Switching Protocols',  
  '102': 'Processing',  
  '200': 'OK',  
  '201': 'Created',  
  '202': 'Accepted',  
  '203': 'Non-Authoritative  
Information',  
  '204': 'No Content',  
  '205': 'Reset Content',  
  '206': 'Partial Content',  
  '207': 'Multi-Status',  
  '208': 'Already Reported',  
  '226': 'IM Used',  
  '300': 'Multiple Choices',  
  '301': 'Moved Permanently',  
  '302': 'Found',  
  '303': 'See Other',  
  '304': 'Not Modified',  
  '305': 'Use Proxy',  
  '307': 'Temporary Redirect',  
  '308': 'Permanent Redirect',  
  '400': 'Bad Request',  
  '401': 'Unauthorized',  
  '402': 'Payment Required',  
  '403': 'Forbidden',  
  '404': 'Not Found',  
  '405': 'Method Not Allowed',  
  '406': 'Not Acceptable',  
  '407': 'Proxy Authentication  
Required',  
  '408': 'Request Timeout',  
  '409': 'Conflict',  
  '410': 'Gone',  
  '411': 'Length Required',  
  '412': 'Precondition Failed',  
  '413': 'Payload Too Large',  
  '414': 'URI Too Long',  
  '415': 'Unsupported Media Type',  
  '416': 'Range Not Satisfiable',  
  '417': 'Expectation Failed',  
  '418': 'I\'m a teapot',  
  '421': 'Misdirected Request',  
  '422': 'Unprocessable Entity',  
  '423': 'Locked',  
  '424': 'Failed Dependency',  
  '425': 'Unordered Collection',  
  '426': 'Upgrade Required',  
  '428': 'Precondition Required',  
  '429': 'Too Many Requests',  
  '431': 'Request Header Fields Too  
Large',  
  '451': 'Unavailable For Legal  
Reasons',  
  '500': 'Internal Server Error',  
  '501': 'Not Implemented',  
  '502': 'Bad Gateway',  
  '503': 'Service Unavailable',  
  '504': 'Gateway Timeout',  
  '505': 'HTTP Version Not  
Supported',  
  '506': 'Variant Also Negotiates',  
  '507': 'Insufficient Storage',  
  '508': 'Loop Detected',  
  '509': 'Bandwidth Limit Exceeded',  
  '510': 'Not Extended',  
  '511': 'Network Authentication  
Required' }
```

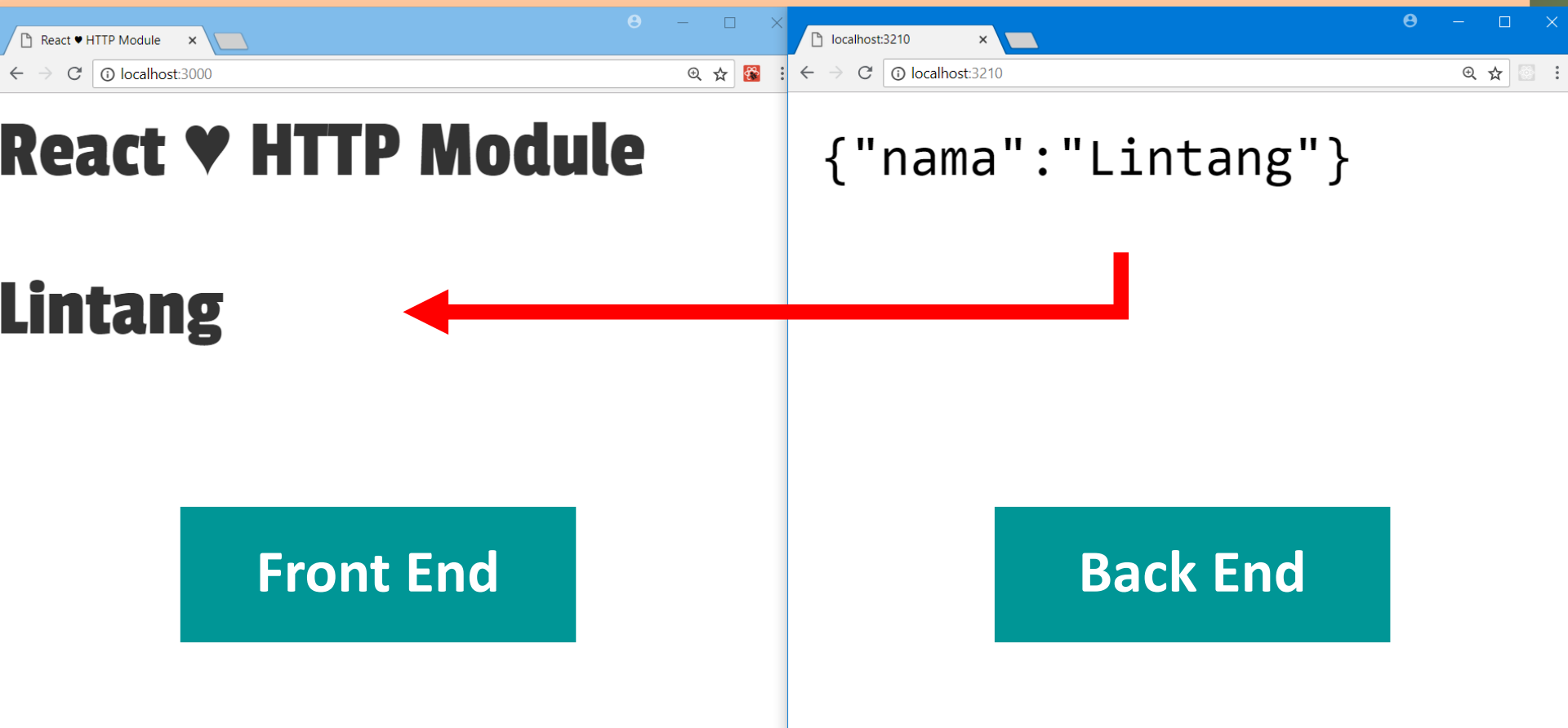
```
var http = require('http');

var server = http.createServer(function(req, res){
  console.log('Request : ' + req.url);
  res.writeHead(200, {
    'Content-Type': 'text/plain',
    'Access-Control-Allow-Origin': '*'
  });
  var data = {'nama': 'Lintang'};
  res.end(JSON.stringify(data));
});

server.listen(3000, ()=>{
  console.log('Server aktif di port 3000.');
```

** Cobalah GET dari React!*

Allow Access



Back-End Development

Exploring

#3 HTTP Module

