`



Training Document

Van Lang Admissions

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Reason for changes | Version |
| Tai Nguyen | 27/10/2016 | Training Node-JS | 1.0 |
|  |  |  |  |

Table of Contents

[Revision Table 1](#_Toc465367864)

[Lệnh Window Command Line Thông Dụng 3](#_Toc465367865)

[Bài 1: viết chương trình quản lý sinh viên đơn giản bằng NodeJS gồm các chức năng sau: 3](#_Toc465367866)

[Load thư viện: 5](#_Toc465367867)

[Lấy danh sách sinh viên: 6](#_Toc465367868)

[Lấy chi tiết một sinh viên: 6](#_Toc465367869)

[Thêm một sinh viên: 7](#_Toc465367870)

[Xóa sinh viên: 8](#_Toc465367871)

[Sửa sinh viên: 9](#_Toc465367872)

[Hiển thị danh sách sinh viên: 9](#_Toc465367873)

[Cách sử dụng thư viện trên 10](#_Toc465367874)

[Bài 2: Thực tập với webserver: 11](#_Toc465367875)

[Mã code: 11](#_Toc465367876)

[Chạy Server: 13](#_Toc465367877)

[Chạy trên browser: 13](#_Toc465367878)

[Truy cập một file trên Server 14](#_Toc465367879)

[Referece: http://freetuts.net/ 17](#_Toc465367880)

# Lệnh Window Command Line Thông Dụng

|  |  |
| --- | --- |
| Lệnh | Ý nghĩa |
| cd | Change Directory: Di chuyển qua lại giữa các Folder. nếu di chuyển vào folder con thì chỉ cần gõ cd tenfolder, còn di chuyển ngược lại thì gõ cd ... Nếu bạn muốn đổi ổ đĩa thì gõ tenodia:, ví dụ D:. |
| dir | Liệt kê danh sách các folder và file nằm trong folder hiện tại |
| mkdir | Tạo một folder con mới. Ví dụ mkdir freetuts thì sẽ tạo folder mới tên là freetuts. |
| rmdir hoặc rd | Xóa folder con. Ví dụ rmdir freetuts thì sẽ xóa folder freetuts. |
| echo | In một giá trị ra ngoài màn hình. Ví dụ echo freetuts thì sẽ in ra chữ freetuts. |
| rename | Đổi tên của folder hoặc file. Ví dụ rename freetuts hoc\_php thì sẽ đổi tên folder freetuts sang hoc\_php. |
| type | Tạo một file mới. Ví dụ TYPE nul >filename.log sẽ tạo file filename.log. |
| del | Xóa file. Ví dụ del demo.txt thì sẽ xóa file demo.txt |

Bài 1: viết chương trình quản lý sinh viên đơn giản bằng NodeJS gồm các chức năng sau:

* Hiển thị danh sách sinh viên
* Thêm mới sinh viên
* Xóa sinh viên
* Sửa sinh viên

Để đơn giản thì mỗi sinh viên ta sẽ lưu mã sinh viên và tên sinh viên.

**Hướng dẫn**:

Nếu bạn chưa [cài đặt node-persist](http://freetuts.net/node-persist-local-storage-tren-server-589.html" \o "cài đặt node-persist) thì hãy cài đặt nó trước đi nhé.

Bạn tạo một file tên là node-persist.js nằm ngoài thư mục gốc D:/nodejs-freetuts/, lúc này để chạy file này thì trong Command Prompt bạn cd đến thư mục này và gõ lệnh sau:

|  |  |
| --- | --- |
| 1 | node node-persist.js |

Ok, bây giờ ta sẽ lần lượt liệt kê các hàm theo yêu cầu của đề bài.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | // Load module node-persist  var storage = require('node-persist');    // Hàm khởi tạo  // Load dữ liệu đã lưu trên ổ đĩa  storage.initSync({      dir : "students" // cấu hình nơi lưu trữ dữ liệu nằm trong thư mục students  });    // Hàm lấy danh sách sinh viên  function getAllStudents()  {    }      // Hàm lấy chi tiết sinh viên  function getStudent(studentId)  {    }      // Hàm thêm một sinh viên  function addStudent(id, fullname)  {    }    // Hàm xóa sinh viên  function removeStudent(studentId)  {    }    // Hàm sửa sinh viên  function editStuent(studentId, studentName)  {    }    // Hàm hiển thị danh sách sinh viên  function showStudents()  {    } |

Và sau đây là nội dung của từng hàm.

### Load thư viện:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | // Load module node-persist  var storage = require('node-persist');    // Hàm khởi tạo  // Load dữ liệu đã lưu trên ổ đĩa  storage.initSync({      dir : "students"  }); |

### Lấy danh sách sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | // Hàm lấy danh sách sinh viên  function getAllStudents()  {      // Lấy sinh viên từ nơi lưu trữ      var students = storage.getItemSync('students');        // Nếu không có sinh viên nào thì trả về một mảng rỗng      if (typeof students === "undefined"){          return [];      }        // Ngược lại sẽ trả về danh sách sinh viên      return students;  } |

### Lấy chi tiết một sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | // Hàm lấy chi tiết sinh viên theo id của sinh viên  function getStudent(studentId)  {      // Lấy danh sách sinh viên      var students = getAllStudents();        // Biến lưu trữ sinh viên được tìm thấy      var matchedStudent = null;        // Lặp để tìm sinh viên      for (var i = 0; i < students.length; i++){          if (students[i].id === studentId){              matchedStudent = students[i];              break;          }      }        return matchedStudent;  } |

### Thêm một sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // Hàm thêm một sinh viên  function addStudent(id, fullname)  {      var students = getAllStudents();        students.push({          id : id,          fullname : fullname      });        storage.setItemSync('students', students);  } |

### Xóa sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | // Hàm xóa sinh viên  function removeStudent(studentId)  {      var students = getAllStudents();        for (var i = 0; i < students.length; i++){          if (students[i].id === studentId){              students.splice(i, 1);          }      }        storage.setItemSync('students', students);  } |

### Sửa sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | // Hàm sửa sinh viên  function editStuent(studentId, studentName)  {      var students = getAllStudents();        for (var i = 0; i < students.length; i++){          if (students[i].id === studentId){              students[i].fullname = studentName;          }      }        storage.setItemSync('students', students);  } |

### Hiển thị danh sách sinh viên:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | // Hàm hiển thị danh sách sinh viên  function showStudents()  {      var students = getAllStudents();      students.forEach(function(student){          console.log('Student: ' + student.fullname + ' (' + student.id + ')');      });  } |

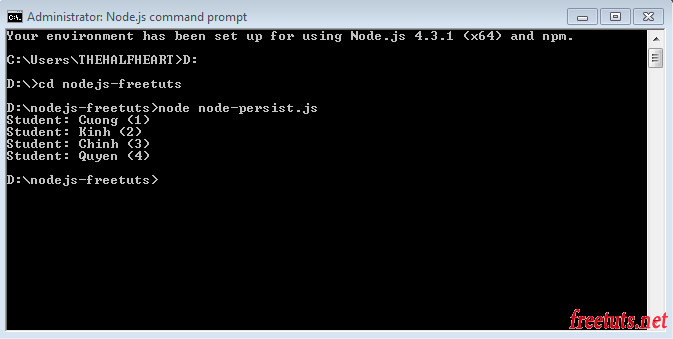
Trong các hàm trên bạn thấy mình sử dụng vòng lặp để duyệt qua từng sinh viên và kiểm tra điều kiện và xử lý theo đề bài. Hy vọng bạn hiểu các hàm trên.

### Cách sử dụng thư viện trên

Bây giờ bạn bổ sung phía dưới cùng một đoạn code như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | // Thêm sinh viên  addStudent(1, 'Cuong');  addStudent(2, 'Kinh');  addStudent(3, 'Chinh');  addStudent(4, 'Quyen');    // Hiển thị danh sách sinh viên  showStudents(); |

Chạy lên kết quả sẽ như hình dưới đây:



Tương tự để xóa một học sinh có id là 1 thì bạn sử dụng hàm sau:

|  |  |
| --- | --- |
| 1 | removeStudent('1'); |

Tương tự cho các hàm còn lại.

## Bài 2: Thực tập với webserver:

Thông thường chúng ta có hai giao thức đó là http và https nên trong NodeJS có cung cấp cho chúng ta hai module tương ứng với hai cái tên đó luôn đó là http và https. Như vậy để tạo server giúp việc giao tiếp giữa client và server thì ta sẽ thông qua hai module này, tuy nhiên trong phạm vi bài viết này mình chỉ hướng dẫn sử dụng http thôi nhé.

### Mã code:

Trước tiên bạn mở file server.js lên và nhập vào nội dung sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | // Bước 1: Import module http  var http = require('http');    // Bước 2: Khởi tạo server  var server = http.createServer(function(request, response){      // Biến request: là biến lưu trữ thông tin gửi lên của client      // Biến response: là biến lưu trữ các thông tin trả về cho client        // Thiết lập Header      response.writeHead(200, {          "Context-type" : "text/plain"      });        // Show thông tin      response.write('Your URL is ' + request.url);        // Kết thúc      response.end();  });    // Bước 3: Lắng nghe cổng 300 thì thực hiện chương trình  server.listen(3000, function(){      console.log('Connected Successfull!');  }); |

Để **tạo NodeJS Server** thì bạn phải thực hiện ba bước gồm:

* **Bước 1**: Import thư viện http
* **Bước 2**: Khởi tạo server. Chúng ta sử dụng hàm http.createServer và tham số truyền vào là một callback function, trong hàm callbackfunction có hai tham số truyền vào nữa đó là request và response,
* **Bước 3**: Thiết lập cổng kết nối

Trong bước 2 bạn để ý đoạn code thiết lập header:

|  |  |
| --- | --- |
| 1  2  3  4 | // Thiết lập Header  response.writeHead(200, {      "Context-type" : "text/plain"  }); |

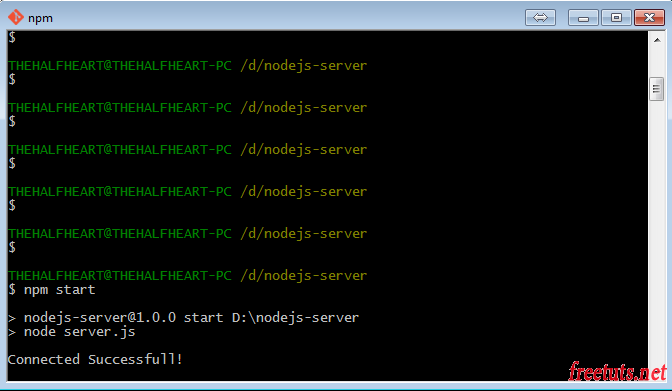
Đây chính là phần khai báo các tham số của trang web như thông tin định dạng file trả về, kích thước file ... Còn mã code 200 tức là truy cập thành công.

### Chạy Server:

Bây giờ bạn bật Git Bash lên và gõ lệnh sau:

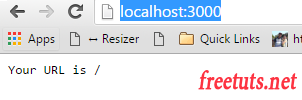
|  |  |
| --- | --- |
| 1 | npm start |

Và nếu giao diện xuất hiện như hình dưới đây thì tức là bạn đã khởi chạy Server thành công.



### Chạy trên browser:

Bây giờ bạn mở trình duyệt ra và gõ vào URL http://localhost:3000/ thì sẽ xuất hiện giao diện như sau:



Bạn hãy thay đổi phần URL phía dau để xem thành quả nhé.

### Truy cập một file trên Server

Bây giờ bạn tạo một file tên là about.html nằm trong project, sau đó pass nội dung dưới đây vào.

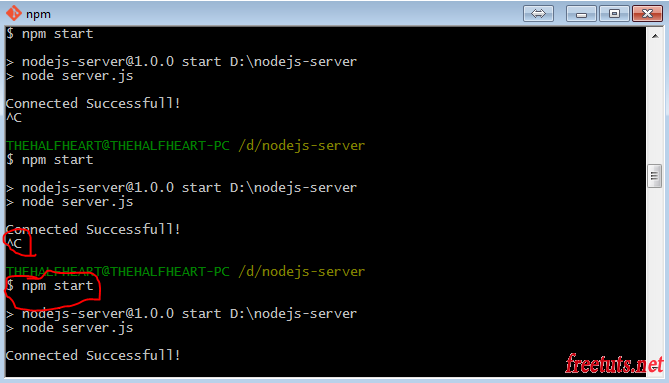
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <!DOCTYPE html>  <html>      <head>          <title>freetuts.net - NodeJS Server căn bản</title>          <meta charset="UTF-8">          <meta name="viewport" content="width=device-width, initial-scale=1.0">      </head>      <body>          <div style="padding: 50px; background: blue; color: #FFF">Xin chào, rất vui khi được chia sẻ kiến thức NodeJS</div>      </body>  </html> |

Tiếp theo bạn sửa lại file server.js như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | // Bước 1: Import module http và fs  var http = require('http');  var fs = require('fs');    // Bước 2: Khởi tạo server  var server = http.createServer(function(request, response){      // Biến request: là biến lưu trữ thông tin gửi lên của client      // Biến response: là biến lưu trữ các thông tin trả về cho client        // Kiểm tra URL truy cập phải trang about ko      if (request.url == '/about.html')      {          // Thiết lập Header          response.writeHead(200, {              "Context-type" : "text/html"          });            // Show thông tin trang about          fs.createReadStream('./about.html').pipe(response);      }      else // trường hợp ngược lại ko tìm thấy file      {          // Thiết lập Header          response.writeHead(404, {              "Context-type" : "text/plain"          });            // Show lỗi không tìm thấy trang          response.write('404 Not Found ' + request.url);            // Kết thúc          response.end();      }  });    // Bước 3: Lắng nghe cổng 300 thì thực hiện chương trình  server.listen(3000, function(){      console.log('Connected Successfull!');  }); |

Trong đoạn code này mình có sử dụng một module tên là fs, đây là module thao tác với file của NodeJS.

Bạn vào Git Bash và nhấn Ctr + C để hủy bỏ những thiết lập trước đó, sau đó chạy lại lệnh npm start nhé.



Bây giờ bạn chạy lên và nhập vào URL như sau http://localhost:3000/about.html thì sẽ thấy kết quả như hình là mọi thứ OK rồi đấy.

# Referece: http://freetuts.net/