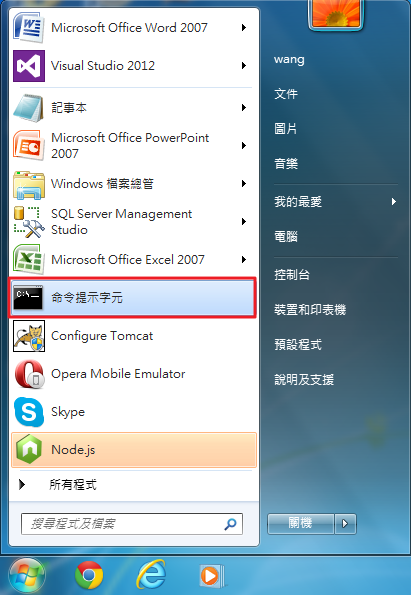
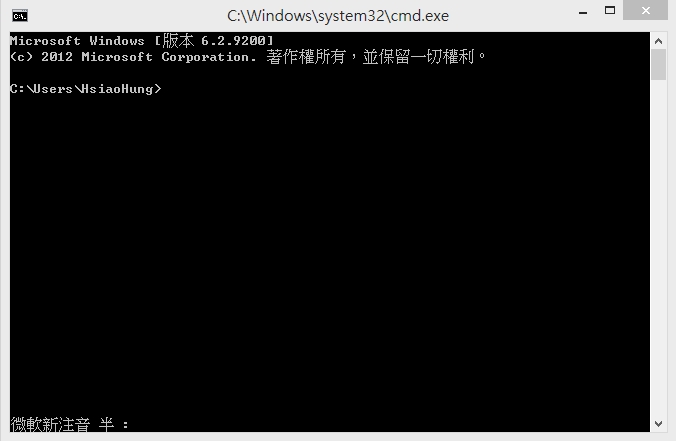
**建立一個Express應用程式，安裝相關模組**

開啟**命令提示字元**

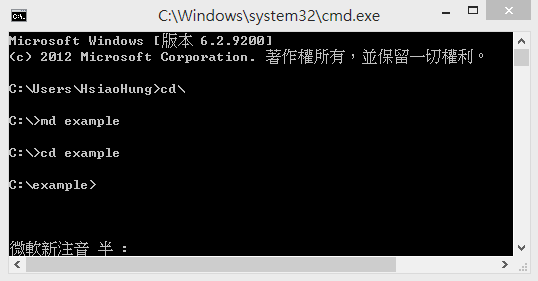
****



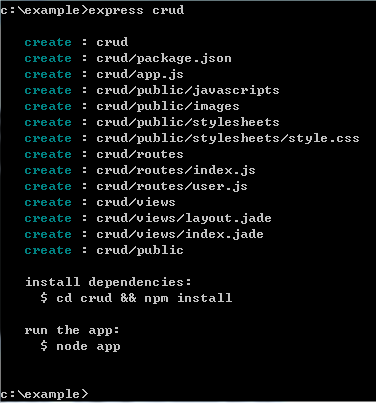
回到根目錄 [cd\]

建立example的資料夾 [md example]

切換到example目錄 [cd example]

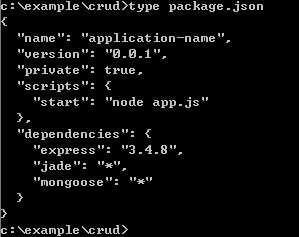


建立一個Express應用程式，名稱叫做crud [express crud]

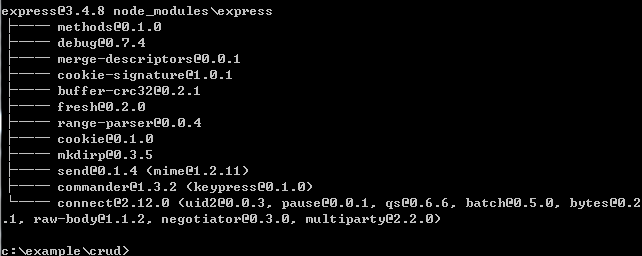
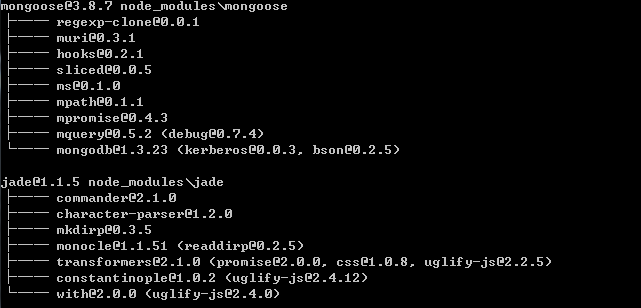
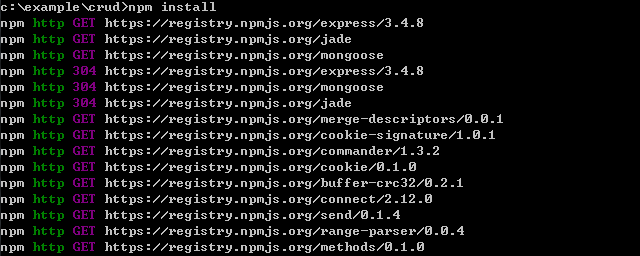


切換到crud目錄 [cd crud]

在這個練習中我們要使用mongoose 模組對MongoDB進行新增、刪除、修改的功能，所以我們先使用記事本開啟C:\example\crud\package.json在dependencies的區段中加上 "mongoose":"\*"，然後再根據package.json安裝相關模組，先開啟packagejson看看修改後的結果[type package.json]



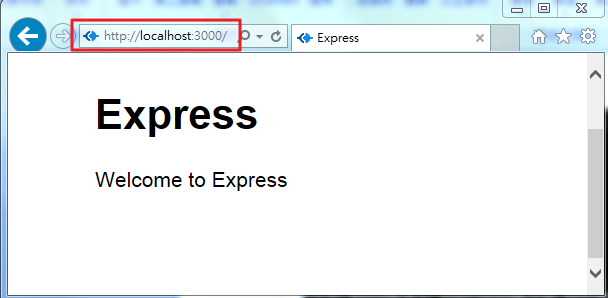
使用 npm 安裝相關模組 [npm install]



執行 crud app [node app]



打開瀏覽器輸入 http://localhost:3000，看到下面頁面就表示crud app建置成功了



**修改共用版型及樣式**

我們先針對網頁做一些設計，建立共用樣版，開啟C:\example\crud\views\layout.jade

將原有頁面改成如下所示

doctype html

html

block head

title= title

link(rel='stylesheet', href='/stylesheets/style.css')

body

header#banner

h1 MongoDB-CRUD

block content

footer @iiiedu

加上一些樣式，讓網頁看起來美觀一點

開啟及修改C:\example\crud\public\stylesheets\style.css如下

html {

height: 100%;

}

body {

font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;

margin: 0px;

padding: 0px;

height: -moz-calc(100% - 20px);

height: -webkit-calc(100% - 20px);

height: calc(100% - 20px);

}

section#chatroom {

height: -moz-calc(100% - 80px);

height: -webkit-calc(100% - 80px);

height: calc(100% - 80px);

background-color: #EFFFEC;

}

div#messages {

height: -moz-calc(100% - 35px);

height: -webkit-calc(100% - 35px);

height: calc(100% - 35px);

padding: 10px;

-moz-box-sizing:border-box;

-webkit-box-sizing:border-box;

box-sizing:border-box;

}

input#message {

width: -moz-calc(100% - 80px);

width: -webkit-calc(100% - 80px);

width: calc(100% - 80px);

}

header{

background-color:#4192C1;

text-align: right;

margin-top: 15px;

}

header h1{

padding: 5px;

padding-right: 15px;

color: #FFFFFF;

margin: 0px;

}

footer{

padding: 6px;

background-color:#4192C1;

color: #FFFFFF;

bottom: 0;

position: absolute;

width: 100%;

margin: 0px;

margin-bottom: 10px;

-moz-box-sizing:border-box;

-webkit-box-sizing:border-box;

box-sizing:border-box;

text-align:center;

}

footer a{

color:purple;

}

a {

color: #00B7FF;

}

div#content{

margin:0 auto;

width:500px;

}

#table1{

width:400px;

border:1px solid green;

border-collapse:collapse;

}

#table1 th,#table1 td{

border:1px solid green;

padding:4px;

}

/\*form\*/

#userform div

{

width: 300px;

border-bottom: 1px solid #d2dadd;

padding-bottom: 3px;

margin-bottom: 10px;

}

#userform div:nth-child(4)

{

text-align: center;

}

#userform label

{

width: 70px;

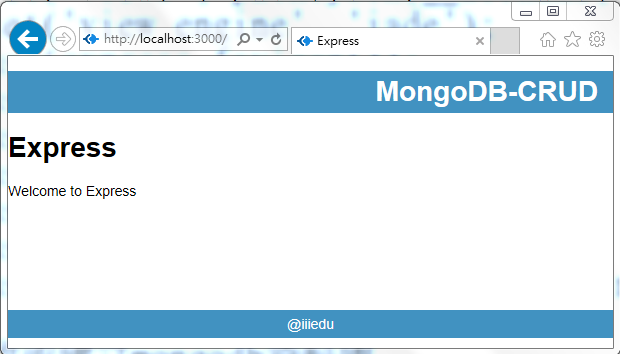
float: left;

text-align: right;

padding-right: 10px;

}

重新整理(F5)瀏覽器，會看到下面頁面，是不是漂亮多了



**使用mongoose 對 mongoDB 完成CRUD的操作**

**新增**

我們先來完成一個新增的頁面，建立一個new.jade的檔案，輸入下列程式，存到C:\example\crud\views目錄底下

extends layout

block content

div#content

h1 新增使用者

form#userform(method="post" action="/new")

div

label(for="name") 姓名:

input#name(type="text" name="name")

div

label(for="email") 電子郵件:

input#email(type="email" name="email")

div

label(for="age") 年紀:

input#age(type="number" name="age")

div: button(type="submit") 新增

設定route才能瀏覽到這個新增的頁面，開啟C:\example\crud\views\app.js 在app.get('/users', user.list);程式之下，加入下面程式

app.get('/new', function(req,res){

res.render("new");

})

將app.js存檔後，重新啟動crud pp，開啟瀏覽器，在網址列輸入http://localhost:3000/new，您將會看到如下頁面



現在我們準備使用mongoose協助我們針對mongodb資料庫建立新增、修改、刪除以及讀取的操作，開啟C:\example\crud\app.js，在var path = require('path');的下一行

引用mongoose模組

var mongoose = require('mongoose');

在app.use(express.static(path.join(\_\_dirname, 'public')));的下面，使用 mongoose 開啟或建立mongodb資料庫，以及mongoose schema

mongoose.connect('mongodb://localhost/MyDB');

var UserSchema = new mongoose.Schema({

name:String,

email:String,

age:Number

}),

Users = mongoose.model('Users',UserSchema);

新增使用者

app.post("/new",function(req,res){

var b = req.body;

new Users({

name: b.name,

email:b.email,

age:b.age

}).save(function(err, user){

if(err) res.json(err);

res.redirect("/");

})

})

**瀏覽**

我們要接著完成瀏覽使用者資料的功能，才能看到剛剛新增完成是否成功

開啟並修改C:\example\crud\views\index.jade的檔案，程式碼如下

extends layout

block content

div#content

h1 所有使用者

a(href="/new") 新增

table(id="table1")

thead

tr

th 姓名

th 年紀

th 電子郵件

th 編輯

th 刪除

tbody

each user in users

tr

td #{user.name}

td #{user.age}

td #{user.email}

td

a(href="/edit/#{user.name}") 編輯

td

a(href="/delete/#{user.name}") 刪除

在app.js的檔案中註解及加入下面程式

//app.get('/', routes.index);

app.get("/",function(req,res){

Users.find({},function(err,docs){

res.render('index',{users:docs});

});

})

將app.js存檔後，重新啟動crud pp，開啟瀏覽器，在網址列輸入http://localhost:3000/，您將會看到如下頁面



按下新增連結，會連到新增使用者頁面



輸入使用者資料後，再按下新增按鈕，新增完成後，會再轉回顯示所有使用者的頁面，如果新增有成功就會看到剛剛新增的使用者資料



**修改**

我們先來完成一個編輯使用者資料的頁面，建立一個edit.jade的檔案，輸入下列程式，存到C:\example\crud\views目錄底下

block content

div#content

h1 修改使用者

form#userform(method="POST" action="/#{user.name}")

div

label(for="name") 姓名:

input#name(type="text" name="name" value="#{user.name}")

div

label(for="email") 電子郵件:

input#email(type="email" name="email" value="#{user.email}")

div

label(for="age") 年紀:

input#age(type="number" name="age" value="#{user.age}")

div

button(type="submit") 修改

input(type="hidden", name="\_method", value="PUT")

在app.js的檔案中加入下面route程式

app.get('/edit/:name', function(req,res){

Users.find({"name":req.params.name},function(err,users){

res.render('edit',{user:users[0]});

});

})

將app.js存檔後，重新啟動crud pp，開啟瀏覽器，在網址列輸入http://localhost:3000/，您將會看到如下頁面



現在您可以點選[編輯]連結，會轉到修改使用者的頁面，並且將此使用者的資料帶入到文字輸入的欄位中



在app.js的檔案中加入將使用者資料更新到資料庫中程式

app.put('/:name', function(req,res){

var b = req.body;

Users.update(

{name:req.params.name},

{ name: b.name,

email:b.email,

age:b.age},

function(err){

res.redirect("/");

});

});

將app.js存檔後，重新啟動crud pp，開啟瀏覽器，進到修改使用者頁面，將原有的Jack、Jack@gmail.com、23的資料修改成Jack1、Jack1@gmail.com、25

按下[修改]按鈕，修改完成後轉回顯示所有使用者的頁面，您將看到修改後的結果



**刪除**

這裡我們不需要建立刪除的頁面，直接在顯示所有使用者的頁面中，按下[刪除]的連結來進行刪除，開啟app.js輸入下列程式

app.get('/delete/:name', function(req,res){

Users.remove({name:req.params.name},

function(err){

res.redirect("/");

});

})

將app.js存檔後，重新啟動crud pp，開啟瀏覽器進入http://localhost:3000



按下[刪除]連結，刪除完成後，您會看到看看您要刪除的使用者，在網頁上已經看不到了



**完成了**

以下貼上app.js所有程式碼，粗體字部分的程式，就是在此練習中加入的

/\*\*

\* Module dependencies.

\*/

var express = require('express');

var routes = require('./routes');

var user = require('./routes/user');

var http = require('http');

var path = require('path');

**var mongoose = require('mongoose');**

var app = express();

// all environments

app.set('port', process.env.PORT || 3000);

app.set('views', path.join(\_\_dirname, 'views'));

app.set('view engine', 'jade');

app.use(express.favicon());

app.use(express.logger('dev'));

app.use(express.json());

app.use(express.urlencoded());

app.use(express.methodOverride());

app.use(app.router);

app.use(express.static(path.join(\_\_dirname, 'public')));

//開啟或建立mongodb資料庫

**mongoose.connect('mongodb://localhost/MyDB');**

//建立Schema

**var UserSchema = new mongoose.Schema({**

**name:String,**

**email:String,**

**age:Number**

**}),**

**Users = mongoose.model('Users',UserSchema);**

// development only

if ('development' == app.get('env')) {

app.use(express.errorHandler());

}

**//**app.get('/', routes.index);

app.get('/users', user.list);

//顯示所有使用者

**app.get("/",function(req,res){**

**//搜尋**

**Users.find({},function(err,docs){ //{}表示取出所有Users**

**res.render('index',{users:docs});**

**});**

**})**

//新增頁面

**app.get('/new', function(req,res){**

**res.render("new");**

**})**

//新增

**app.post("/new",function(req,res){**

**var b = req.body;**

**new Users({**

**name: b.name,**

**email:b.email,**

**age:b.age**

**}).save(function(err, user){**

**if(err) res.json(err);**

**res.redirect("/");**

**})**

**})**

//編輯

**app.get('/edit/:name', function(req,res){**

**Users.find({"name":req.params.name},function(err,users){**

**res.render('edit',{user:users[0]});**

**});**

**})**

//修改

**app.put('/:name', function(req,res){**

**var b = req.body;**

**Users.update(**

**{name:req.params.name},**

**{ name: b.name,**

**email:b.email,**

**age:b.age},**

**function(err){**

**res.redirect("/");**

**});**

**});**

//刪除

**app.get('/delete/:name', function(req,res){**

**Users.remove({name:req.params.name},**

**function(err){**

**res.redirect("/");**

**});**

**})**

http.createServer(app).listen(app.get('port'), function(){

console.log('Express server listening on port ' + app.get('port'));

});