

Javascript

is **Cool** and the market is **Hot**

You can mint money with it.

But it is not for those who have **bandwidth, CPU**
and **memory** issues.

Setting up



Download the current version from <http://nodejs.org/en>

I am using V6.3.0

The download is Windows (X64)

An node-v6.3.0-x64 MSI file will be downloaded

Right click it and install it.

During installation change the path from C:\Program Files\nodejs\, to c:\nodejs

Installation is pretty simple.

Now create a folder on c:\ named as nodeeg for our initial examples

Add [c:\nodejs](#) to path environment variable (Control panel/System/Advance settings/Environment Variables). We have done it earlier.

now in our [c:\nodeeg](#) folder create your example js files and run it as done in classroom session.

eg1.js

```
console.log("God is great\nWe are thankful to God.");
```

eg2.js

```
console.log("To err, is human
```

eg3.js

```
var x;  
console.log(x);  
x=0;  
console.log(x);  
x=0.01;  
console.log(x);  
x="Cool";  
console.log(x);  
x='Cool';  
console.log(x);  
x=true;  
console.log(x);  
x=false;  
console.log(x);  
x=null;  
console.log(x);  
x=undefined;  
console.log(x);
```

eg4.js

```
var x  
console.log(x)  
x=0  
console.log(x)  
x=0.01  
console.log(x)  
x="Cool"  
console.log(x)  
x='Cool'  
console.log(x)  
x=true  
console.log(x)  
x=false  
console.log(x)  
x=null
```

```
console.log(x)
x=undefined
console.log(x)
```

```
eg5.js
var x
=5
console.log("Great")
console.log(x)
```

```
eg6.js
var x
=5
var y
y
=10
=20
=30
console.log("Great")
console.log(x)
```

```
eg7.js
var x;
x=10
var y
y=20
x
++
y
console.log(x);
console.log(y);
```

```
eg8.js
console.log(Number.MAX_VALUE);
console.log(Number.MIN_VALUE);
console.log(Number.MAX_VALUE+1);
console.log(Number.MIN_VALUE-1);
var x;
x=Number.MAX_VALUE+10;
console.log(x);
x=Number.MAX_VALUE-10;
console.log(x);
x=0/0;
console.log(x);
x=10/0;
console.log(x);
x=-1/Infinity;
console.log(x);
x=-2/Infinity;
console.log(x);
```

```
x=-0;  
console.log(x);
```

```
eg9.js
```

```
var x;  
x=0;  
console.log(x);  
console.log(y);  
if(x==undefined)  
{  
  console.log("x is not defined");  
}  
else  
{  
  console.log("x is defined");  
}  
if(y==undefined)  
{  
  console.log("y is not defined");  
}  
else  
{  
  console.log("y is defined");  
}
```

```
eg10.js
```

```
var x,y;  
x=0;  
console.log(x);  
console.log(y);  
if(x==undefined)  
{  
  console.log("x is not defined");  
}  
else  
{  
  console.log("x is defined");  
}  
if(y==undefined)  
{  
  console.log("y is not defined");  
}  
else  
{  
  console.log("y is defined");  
}
```

```
eg11.js
```

```
var x;
```

```
x=10;
if(x=="10")
{
console.log('Value of x is 10 as a string');
}
else
{
console.log('Value of x is not 10 as a string');
}
```

```
eg12.js
var x;
x=10;
if(x==="10")
{
console.log('Value of x is 10 as a string');
}
else
{
console.log('Value of x is not 10 as a string');
}
```

```
eg13.js
var x;
x=10;
if(x!="10")
{
console.log('Value of x is not 10 as a string');
}
else
{
console.log('Value of x is 10 as a string');
}
```

```
eg14.js
var x;
x=10;
if(x!==="10")
{
console.log('Value of x is not 10 as a string');
}
else
{
console.log('Value of x is 10 as a string');
}
```

```
eg15.js
var a;
a="Amar";
var b;
```

```
b="Boy";
if(a<b)
{
console.log(a+" should be placed before "+b+" in a dictionary");
}
else
{
console.log(a+" should not be placed before "+b+" in a dictionary");
}
```

```
eg15.js
var a;
a="Good";
var b;
b="Good";
if(a==b)
{
console.log(a+" is equal to "+b);
}
else
{
console.log(a+" is not equal to "+b);
}
b=new String("Good");
if(a==b)
{
console.log(a+" is equal to "+b);
}
else
{
console.log(a+" is not equal to "+b);
}
```

```
eg16.js
var x=-1;
console.log("Value of x is "+x);
if(x)
{
console.log(x+" stands for true");
}
else
{
console.log(x+" stands for false");
}
x=-0;
console.log("Value of x is "+x);
if(x)
{
```

```
console.log(x+" stands for true");
}
else
{
console.log(x+" stands for false");
}
var y;
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
y="Cool";
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
y="";
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
y="  ";
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
y=10/0;
```



```
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
y=NaN;
console.log("Value of y is "+y);
if(y)
{
console.log(y+" stands for true");
}
else
{
console.log(y+" stands for false");
}
}
```

eg17.js

```
var x;
x="10"+20;
console.log(x);
x="10"*20;
console.log(x);
x="10"-20;
console.log(x);
x="10"/20;
console.log(x);
x=Number("10")+20;
console.log(x);
x="10";
if(x===10)
{
console.log("Value of x is 10 as a number");
}
else
{
console.log("Value of x is not 10 as a number");
}
if(Number(x)===10)
{
console.log("Value of x is 10 as a number");
}
else
{
```

```
console.log("Value of x is not 10 as a number");
}
// try examples with String(10) Boolean("false")
```

```
eg18.js
function add(x,y)
{
return x+y;
}
var x;
x=add(10,20);
console.log(x);
```

```
eg19.js
var x;
x=100;
function sam()
{
x=10;
}
console.log(x);
sam();
console.log(x);
```

```
eg20.js
var x;
x=100;
function sam()
{
var x;
x=10;
}
console.log(x);
sam();
console.log(x);
```

```
eg21.js
function sam()
{
x=10;
console.log(x);
}
function tom()
{
console.log(x);
}
sam();
tom();
```

```
eg22.js
function sam()
```

```
{
var x;
x=10;
console.log(x);
}
function tom()
{
console.log(x);
}
sam();
tom();
```

```
eg23.js
var x;
function sam(x,y)
{
return x*y;
};
x=sam;
var y;
y=x(10,30);
console.log(sam);
console.log(x);
console.log(y);
```

```
eg24.js
var x;
x=function(x,y)
{
return x*y;
};
var y;
y=x(10,30);
console.log(x);
console.log(y);
```

```
eg25.js
function sam(f)
{
var x;
x=f(10,20);
console.log(x);
}
function tom(p,q)
{
return p*q;
}
sam(tom);
```

```
eg26.js
```

```
var x;
x=function(r){
return r(10,20);
};
var y=x(function(p,q){
return p+q;
});
console.log(y);
```

```
eg27.js
function sam(p,q,r)
{
console.log(p);
console.log(q);
console.log(r);
}
sam();
sam(1);
sam(1,2);
sam(1,2,3);
```

```
eg28.js
function sam(p,q,r)
{
if(p===undefined && q===undefined && r===undefined)
{
console.log('No arguments');
}
else if(q===undefined && r===undefined)
{
console.log(p);
}
else if(r===undefined)
{
console.log(p+" "+q);
}
else
{
console.log(p+" "+q+" "+r);
}
}
sam();
sam(1);
sam(1,2);
sam(1,2,3);
```

```
eg29.js
function sam(p,q,r)
{
```

```
var x=0;
while(x<arguments.length)
{
console.log(arguments[x]);
x++;
}
console.log("-----");
}
sam();
sam(1);
sam(1,2);
sam(1,2,3);
sam(1,2,3,4);
sam(1,2,3,4,5);
```

```
eg30.js
function sam(p,q,r)
{
console.log(sam.toString());
}
var x=function(a,b,c,d) /* this is a cool function */
{
// Some cool functionality
return a+b+c+d;
}
sam();
console.log(x.toString());
```

```
eg31.
var k="God is great";
console.log(k.search("great"));
console.log(k.search("Great"));
var p=/GREAT/i;
console.log(k.search(p));
```

Phase Ends

Javascript Regular Expression is a Cool Thing.

eg32.js

```
var x;  
x="God is great, We are grateful to god. May great god help you."  
var p;  
p=/great/;  
x=x.replace(p,"GREAT");  
console.log(x);  
p=/great/g;  
x=x.replace(p,"GREAT");  
console.log(x);
```

eg33.js

```
var k="God is great";  
var p=/GREAT/i;  
console.log(p.test(k));  
p=/GREAT/  
console.log(p.test(k));  
var x;  
x="God is great, We are grateful to god. May great god help you."  
var p;  
p=/great/;  
console.log(p.test(x));  
p=/great/g;  
console.log(p.test(x));
```

eg34.js

```
var x;  
x="Ujjain\nis the city of gods.\nUjjain is a cool place";  
var p;  
p=/cool/;
```

```
console.log(p.test(x));
p=/^Ujjain/;
console.log(p.test(x));
p=/place$/;
console.log(p.test(x));
p=/^ujjain/;
console.log(p.test(x));
p=/Place$/;
console.log(p.test(x));
p=/^ujjain/i;
console.log(p.test(x));
p=/Place$/i;
console.log(p.test(x));
p=/Ujjain$/;
console.log(p.test(x));
p=/^place/;
console.log(p.test(x));
p=/^Ujjain$/;
console.log(p.test(x));
x="Ujjain is a cool place. I live in Ujjain";
p=/^Ujjain$/;
console.log(p.test(x));
p=/^Ujjain.*$/;
console.log(p.test(x));
```

```
eg35.js
var k="God is great";
console.log(k.search("great"));
console.log(k.search("Great"));
var p=new RegExp("GREAT","i");
console.log(k.search(p));
```

Assignment (convert eg32 to eg34 to new RegExp version)

```
eg36.js
var p;
p=/Great/i;
var x;
x="God is great. We are thankful to god for his greatness";
console.log(p.test(x));
console.log(p.exec(x));
```

```
eg37.js
var p;
p=/Great/;
var x;
x="God is great. We are thankful to god for his greatness";
console.log(p.test(x));
console.log(p.exec(x));
```

```
eg38.js
var p=/Great/ig;
var x="God is great, we are thankful to god for his greatness";
var k;
while(true)
{
k=p.exec(x);
if(!k) break;
console.log(k);
}
```

```
eg39.js
var p=/Great/ig;
var x="God is great, we are thankful to god for his greatness";
var k;
while(k=p.exec(x))
{
console.log(k);
}
```

```
eg40.js
var x="God is great. we are thankful to god for his greatness";
var p=/Great/ig;
var m;
m=p.exec(x);
console.log(m);
console.log(p.lastIndex);
m=p.exec(x);
console.log(m);
console.log(p.lastIndex);
```

```
eg41.js
var x="God is great. we are thankful to god for his greatness";
var p=/Great/ig;
var m;
m=p.exec(x);
console.log(m);
console.log(p.index);
m=p.exec(x);
console.log(m);
console.log(p.index);
```

```
eg42.js
var x;
x="God is great. He is really great.";
var p;
p=/(\gr)eat/;
var m=p.exec(x);
console.log(m);
```

```
eg43.js
var x,y;
x="This is coolest programming stuff.";
y="That is coolest programming stuff.";
p=/(?This)/;
console.log(p.exec(x));
console.log(p.exec(y));
```

```
eg44.js
var x,y;
x="This is coolest programming stuff.";
y="That is coolest programming stuff.";
p=/(?This|That)/;
console.log(p.exec(x));
console.log(p.exec(y));
```

```
eg45.js
var x,y;
x="This is coolest programming stuff";
y="That is cooest programming language";
p=/(?STUFF|LANGUAGE)/;
console.log(p.exec(x));
console.log(p.exec(y));
p=/(?STUFF|LANGUAGE)$/i;
console.log(p.exec(x));
console.log(p.exec(y));
```

Many more examples of regular expression are pending. This was just to give you a head start. Refer any document and try to learn other rules. We will continue this topic after going through other features of javascript.

Array – A Powerful tool – We need to exploit it properly

```
eg46.js
var x=[10,20,"good",true];
for(var i=0;i<x.length;i++)
{
console.log(x[i]);
}
```

```
eg47.js
var x=['a',2,];
console.log("Length"+x.length);
for(var i=0;i<x.length;i++)
{
console.log(x[i]);
}
```

```
eg48.js
var x=[10,,20];
```

```
for(var i=0;i<x.length;i++)
{
console.log(x[i]);
}
```

```
eg49.js
var k;
k[0]=10;
```

```
eg50.js
var x=[10,20,30,40];
x[20]=93;
for(var i=0;i<x.length;i++)
{
console.log(x[i]);
}
console.log("-----");
console.log(x[100]);
```

```
eg51.js
var x=new Array();
console.log("Length : "+x.length);
x[0]=20;
x[4]=30;
var i;
for(i=0;i<x.length;i++) console.log(x[i]);
console.log("-----");
x=new Array(50);
console.log("Length : "+x.length);
x[100]=30;
console.log("Length : "+x.length);
x=new Array(10,20,"good,bad");
console.log("Length : "+x.length);
```

```
eg52.js
var x;
x=new Array(5);
console.log(Array.isArray(x));
x=20;
console.log(Array.isArray(x));
x=[];
console.log(Array.isArray(x));
x=[10,20];
console.log(Array.isArray(x));
```

```
eg53.js
var x;
x=new Array();
x[0]=new Array(2);
x[1]=22;
x[2]=33;
```

```
x[4]=new Array("gold","Silver","dust");
x[5]=new Array(2);
x[5][0]=new Array(3);
x[5][0][0]=new Array(4);
x[5][0][0]=220;
x[5][0][0][1]=293;
function arrayPrinter(a)
{
console.log("---- arrayPrinter starts -----");
if(Array.isArray(a))
{
console.log(a.length);
for(var i=0;i<a.length;i++)
{
if(Array.isArray(x[i])) arrayPrinter(a[i]);
else console.log(x[i]);
}
}
else
{
console.log(a);
}
}
arrayPrinter(x);
```

```
eg54.js
var x=[10,[1,2,3],20,new Array(1,2,3)];
console.log(Array.isArray(x[0]));
console.log(Array.isArray(x[1]));
console.log(Array.isArray(x[2]));
console.log(Array.isArray(x[3]));
console.log(Array.isArray(x[23]));
```

```
eg55.js
var x=[];
x["0"]=20;
x["1"]=30;
x["7"]=40;
for(var i=0;i<x.length;i++) console.log(x[i]);
console.log(x.length);
console.log("-----");
x["good"]=30;
console.log(x.length);
for(var i=0;i<x.length;i++) console.log(x[i]);
console.log("-----");
console.log(x["bad"]);
console.log("-----");
console.log(x["good"]);
```

```
eg56.js
var x=[];
x["good"]=30;
console.log(x.good);
console.log(x["good"]);
```

```
eg57.js
var x=[10,20,30];
console.log(x.length);
var i;
for(i=0;i<x.length;i++) console.log(x[i]);
console.log('-----');
delete x[1];
console.log(x.length);
for(i=0;i<x.length;i++) console.log(x[i]);
```

```
eg58.js
var x;
x=new Array();
x[0]=new Array(2);
x[1]=22;
x[2]=33;
x[4]=new Array("gold","Silver","dust");
x[5]=new Array(2);
x[5][0]=new Array(3);
x[5][0][0]=new Array(4);
x[5][0][0]=220;
x[5][0][0][1]=293;
console.log(x.toString());
console.log("-----");
console.log(x.join("***"));
```

```
eg59.js
var x=new Array();
x.push(10);
x.push(20);
x.push(30);
x.push(40);
x.push(50);
var i;
for(i=0;i<x.length;i++) console.log(x[i]);
```

```
eg60.js
var x=new Array();
x.push(10);
x.push(20);
x.push(30);
x.push(40);
x.push(50);
var i;
```

```
console.log("Length : "+x.length);
for(i=0;i<x.length;i++) console.log(x[i]);
console.log("-----");
x.pop();
x.pop();
console.log("Length : "+x.length);
for(i=0;i<x.length;i++) console.log(x[i]);
```

```
eg61.js
var queue=new Array();
queue.push(10);
queue.push(20);
queue.push(30);
var y;
y=queue.shift();
console.log(y);
console.log("-----");
for(var i=0;i<queue.length;i++) console.log(queue[i]);
y=queue.shift();
console.log(y);
y=queue.shift();
console.log(y);
y=queue.shift();
console.log(y);
y=queue.shift();
console.log(y);
y=queue.shift();
console.log(y);
```

```
eg62.js
var x=[10,20,30,40];
x.unshift(5);
x.push(50);
var i;
for(i=0;i<x.length;i++) console.log(x[i]);
console.log("-----");
var y;
y=x.shift();
console.log(y);
console.log("-----");
y=x.pop();
console.log(y);
console.log("-----");
for(i=0;i<x.length;i++) console.log(x[i]);
```

```
eg63.js
var x=[10,20,30,40,50];
var j=x.splice(3,0,33,44);
for(var i=0;i<x.length;i++) console.log(x[i]);
```

```
console.log(j);
console.log("-----");
var y=[100,200,300,400,500];
j=y.splice(3,2);
for(var i=0;i<y.length;i++) console.log(y[i]);
console.log(j)
console.log("-----");
var y=[100,200,300,400,500];
j=y.splice(3,2,333,444,460);
for(var i=0;i<y.length;i++) console.log(y[i]);
console.log(j);
console.log("-----");
```

```
eg64.js
var x=[10,20,30,40,50];
x.splice(2,0,[22,23,24]);
for(var i=0;i<x.length;i++) console.log(x[i]);
```

```
eg65.js
var x=[10,20,30,40,50];
var y=[100,200,300,400,500];
var z=x.concat(y);
console.log(x);
console.log(y);
console.log(z);
```

```
eg66.js
var x=[10,20,30,40,50];
var y=x.slice(2);
console.log(x);
console.log(y);
console.log("-----");
x=[10,20,30,40,50,60];
y=x.slice(2,4);
console.log(x);
console.log(y);
```

```
eg67.js
var x=[10,20,30,40];
var y=x.valueOf();
var z=x.toString();
console.log(y);
console.log(z);
```

```
eg68.js
var x=[40,20,10,20,5,32];
var y;
y=x.sort();
console.log(x);
console.log(y);
var a=["Sudhir", "Amit", "Zeeshan", "Pulkit", "Bobby"];
```

```
var b=a.sort();
console.log(a);
console.log(b);
var c=b.reverse();
console.log(a);
console.log(b);
console.log(c);
```

```
eg69.js
var x=[1,22,3,55,33,22,664,123,53,2];
x.sort(function(a,b) { return a-b; });
console.log(x);
x.sort(function(a,b) { return b-a; });
console.log(x);
```

```
eg70.js
var x=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15];
x.sort(.5-Math.random());
console.log(x);
// run the above script 3 times
```

```
eg71.js
var x=[10,20,30,40,50,60,70,80,90,100];
console.log(x);
x.copyWithIn(4,6);
console.log(x);
x=[10,20,30,40,50,60,70,80,90,100];
x.copyWithIn(4,6,7);
console.log(x);
```

```
eg72.js
var x=[2,4,6,8,42,36];
function isEven(n) { return n%2==0; }
console.log(x.every(isEven));
var x=[2,4,6,8,421,36];
console.log(x.every(isEven));
```

```
eg73.js
var x=[10,20,30,40,50,60];
console.log(x);
x.fill(99);
console.log(x);
x=[10,20,30,40,50,60];
x.fill(99,3,5);
console.log(x);
x=[10,20,30,40,50,60];
x.fill(99,3,15);
console.log(x);
```

```
eg74.js
var x=[10,20,3,30,40,3,20,50,62,32];
console.log(x);
```

```
var f=function(m){ return m%10==0; };
x.filter(f);
console.log(x);
var y;
y=x.filter(f);
console.log(x);
console.log(y);
```

```
eg75.js
var x=[103,201,3,30,40,3,20,50,62,32];
console.log(x);
var f=function(m){ return m%10==0; };
var y;
y=x.find(f);
console.log(x);
console.log(y);
```

```
eg76.js
var x=[103,201,3,30,40,3,20,50,62,32];
console.log(x);
var f=function(m){ return m%10==0; };
var y;
y=x.findIndex(f);
console.log(x);
console.log(y);
```

```
eg77.js
var x=[10,20,30,40,50,60,10,20,30,40,50,60];
var y;
y=x.indexOf(30);
console.log(y);
y=x.indexOf(30,6);
console.log(y);
y=x.indexOf(30,11);
console.log(y);
y=x.lastIndexOf(30);
console.log(y);
y=x.lastIndexOf(30,6);
console.log(y);
y=x.lastIndexOf(30,11);
console.log(y);
y=x.lastIndexOf(30,1);
console.log(y);
```

```
eg78.js
var x=[10,20,30,40,50,60,70,80,90,100];
function sam(n,i,a)
{
console.log(n);
console.log(i);
```



```
console.log(a);  
if(n==50)  
{  
a[i]=500;  
return true;  
}  
}  
var y;  
y=x.find(sam);  
console.log(x);  
console.log(y);
```

Phase Ends

```
eg79.js
function sam(n,i,a)
{
console.log(n);
console.log(i);
console.log(a);
console.log(this.m);
if(n==50)
{
a[i]=500;
return true;
}
}
function tony()
{
this.m=243;
var x=[10,20,30,40,50,60,70,80,90,100];
var y;
y=x.find(sam,this);
console.log(x);
console.log(y);
}
tony();
```

```
eg80.js
function sam(n,i,arr) // 2nd and 3rd param are optional
{
console.log(n);
console.log(i);
console.log(arr);
console.log(i+200);
console.log(this.good);
}
function tony()
{
this.good="bad";
var x=[10,20,30,40,50];
x.forEach(sam,this);
}
tony();
```

```
eg81.js
function mapper(c,i,a) // 2nd and 3rd param are optional
{
var mp=["Ujjain","Indore","Dewas","Agar"];
var maharashtra=["Mumbai","Kolhapur","Pune"];
if(mp.indexOf(c)!=-1) return {"city":c,"state":"M.P.,"country": this.country};
if(maharashtra.indexOf(c)!=-1) return {"city":c,"state":"Maharashtra","country": this.country};
```

```
}
function tony()
{
this.country="India"
var x=["Ujjain","Dewas","Mumbai","Agar","Kolhapur","Indore","Pune"];
var y=x.map(mapper,this);
console.log(x);
console.log(y);
}
tony();
```

```
eg82.js
var x=[10,20,30];
function totalCalculator(t,c,i,a) // 3rd and 4th are optional
{
console.log(t);
console.log(c);
console.log(i);
console.log(a);
return t+c;
}
var y=x.reduce(totalCalculator,100);
console.log('-----');
console.log(x);
console.log(y);
```

```
eg83.js
var x=[10,20,30];
function totalCalculator(t,c,i,a) // 3rd and 4th are optional
{
console.log(t);
console.log(c);
console.log(i);
console.log(a);
return t+c;
}
var y=x.reduceRight(totalCalculator,100);
console.log('-----');
console.log(x);
console.log(y);
```

```
eg84.js
function containsCititesOfMP(c,i,a) // 2nd and 3rd param are optional
{
var mp=["Ujjain","Indore","Dewas","Agar"];
var maharashtra=["Mumbai","Kolhapur","Pune"];
if(mp.indexOf(c)!=-1) return true;
}
var x=["Ujjain","Dewas","Mumbai","Agar","Kolhapur","Indore","Pune"];
```

```
console.log(x.some(containsCititesOfMP));  
var y=["Mumbai","Kolhapur","Pune"];  
console.log(y.some(containsCititesOfMP));
```

Coders Should Learn By Coding

```
eg85.js  
function calculateAmount(p,r,t)  
{  
function calculateSimpleInterest(p,r,t)  
{  
return (p*r*t)/100;  
}  
return p+calculateSimpleInterest(p,r,t);  
}  
console.log(calculateAmount(1000,10,2));
```

```
eg86.js  
function calculateAmount(p,r,t)  
{  
function calculateSimpleInterest()  
{  
return (p*r*t)/100;  
}  
return p+calculateSimpleInterest(p,r,t);  
}  
console.log(calculateAmount(1000,10,2));
```

```
eg87.js  
var object={  
p: 0,  
r: 0,  
t: 0,  
getSimpleInterest:function()  
{  
return (this.p*this.r*this.t)/100;  
},  
getAmount:function()  
{  
return this.p+this.getSimpleInterest();  
}  
};  
object.p=1000;  
object.r=10;  
object.t=2;  
console.log(object.getSimpleInterest());  
console.log(object.getAmount());
```

```
eg88.js
var object={
p: 0,
r: 0,
t: 0,
getSimpleInterest:function()
{
return (this.p*this.r*this.t)/100;
},
getAmount:function()
{
return this.p+this.getSimpleInterest();
}
};
object["p"]=1000;
object["r"]=10;
object["t"]=2;
console.log(object["getSimpleInterest"]());
console.log(object["getAmount"]());
```

```
eg89.js
var bulb={
wattage: 0,
setWattage: function(w)
{
this.wattage=w;
},
getWattage: function()
{
function printBrand()
{
console.log("Wattage is "+this.wattage+" , and its brand is Philips");
}
printBrand();
return this.wattage;
}
}
bulb.setWattage(60);
console.log(bulb.getWattage());
```

```
eg90.js
function Bulb()
{
}
var b=new Bulb;
console.log(b);
var k=new Bulb();
console.log(k);
```

```
function Toy()
{
this.inspect=function()
{
return "[Toy]";
}
}
var t=new Toy();
console.log(t);
```

```
eg91.js
var bulb={
wattage: 0,
setWattage: function(w)
{
this.wattage=w;
},
getWattage: function()
{
var that=this;
function printBrand()
{
console.log("Wattage is "+that.wattage+", and its brand is Philips");
}
printBrand();
return this.wattage;
}
}
bulb.setWattage(60);
console.log(bulb.getWattage());
```

```
eg92.js
function computeSimpleInterest(p,r,t)
{
return ((p*r*t)/100);
}
var x=computeSimpleInterest(1000,10,2);
console.log(x);
x=computeSimpleInterest.call(null,1000,10,2);
console.log(x);
x=computeSimpleInterest.apply(null,[1000,10,2]);
console.log(x);
```

```
eg93.js
function Bulb(price)
{
this.price=price;
}
```

```
function Toy(price)
{
this.price=price;
}
function TV(price)
{
this.price=price;
}
function pricePrinter(product)
{
console.log("Price of "+product+" is "+this.price);
}
var bulb=new Bulb(50);
var toy=new Toy(1230);
var tv=new TV(210000);
pricePrinter.call(bulb,"Bulb");
pricePrinter.call(toy,"Toy");
pricePrinter.call(tv,"TV");
```

eg94.js

```
function Bulb(price)
{
this.price=price;
}
function Toy(price)
{
this.price=price;
}
function TV(price)
{
this.price=price;
}
function pricePrinter(product)
{
console.log("Price of "+product+" is "+this.price);
}
var bulb=new Bulb(50);
var toy=new Toy(1230);
var tv=new TV(210000);
pricePrinter.call(bulb,["Bulb"]);
pricePrinter.call(toy,["Toy"]);
pricePrinter.call(tv,["TV"]);
```

eg95.js

```
function Bulb()
{
this.w=0;
}
```

```
function setWattage(w)
{
this.w=w;
}
function getWattage()
{
return this.w;
}
var b1=new Bulb;
setWattage.call(b1,60);
var b2=new Bulb;
setWattage.call(b2,100);
console.log(getWattage.call(b1));
console.log(getWattage.call(b2));
```

```
eg96.js
var k;
k=20;
var m;
m=k || 100;
console.log(m);
m=200 || k;
console.log(m);
m=0 || k;
console.log(m);
```

```
eg97.js
var k;
console.log(k);
var m;
m=k || 100;
console.log(m);
```

```
eg98.js
function add(a,b,c,d)
{
console.log(a+b+c+d);
}
add(10,20,30,40);
add(10,20);
```

```
eg99.js
function add(a,b,c,d)
{
a=a || 0;
b=b || 0;
if(c===undefined) c=0;
d=d || 0;
console.log(a+b+c+d);
}
```



```
add(10,20,30,40);
add(10,20);
```

```
eg100.js
var o1={
name: "Gopal",
age: 24,
gender: "M"
};
function printMembers(o)
{
for(var p in o)
{
console.log(p+" : "+o[p]);
}
}
printMembers(o1);
```

```
eg101.js
var o1={
name: "Gopal",
age: 24,
gender: "M"
};
var o2={
name: "Ganesh",
age: 22,
gender: "M"
};
var g="Ramesh";
var m=20;
var k=[10,20,30];
var d={};
function e()
{
}
console.log(typeof(o1));
console.log(typeof(o2));
console.log(typeof(sc));
console.log(typeof(g));
console.log(typeof(m));
console.log(typeof(k));
console.log(typeof(d));
console.log(typeof(e));
```

```
eg102.js
var o1={
name: "Gopal",
age: 24,
```

```
gender: "M"
};
var o2={
name: "Ganesh",
age: 22,
gender: "M"
};
var sc={
name: "ABC School",
city: "Ujjain",
students:[o1,o2]
};
function printMembers(o)
{
for(var p in o)
{
if(typeof(o[p])=="object") printMembers(o[p]);
else console.log(p+" : "+o[p]);
}
}
printMembers(sc);
```

```
eg103.js
function sam(p,q,r)
{
console.log(arguments.length);
console.log(arguments.callee);
console.log(arguments.callee.caller);
console.log(arguments.callee.caller.arguments.callee.caller);
console.log(arguments.callee.caller.arguments.callee.caller.arguments.callee.caller);
}
function tom()
{
sam(10,20,30);
}
tom();
```

eg104.js

```
var callStack=new Array();
function sam(p,q,r)
{
if(callStack.length==0) callStack.push(arguments.callee.caller);
callStack.push(arguments.callee);
}
function tom()
{
if(callStack.length==0) callStack.push(arguments.callee.caller);
callStack.push(arguments.callee);
sam(10,20,30);
}
function joy(a,b)
{
if(callStack.length==0) callStack.push(arguments.callee.caller);
callStack.push(arguments.callee);
tom();
}
function lion()
{
if(callStack.length==0) callStack.push(arguments.callee.caller);
callStack.push(arguments.callee);
joy(1,2);
}
lion();
console.log(callStack);
```

eg105.js

```
console.log(getSimpleInterest({principalAmount: 1000, rateOfInterest: 10, timePeriod: 2}));
```

```
function getSimpleInterest(o)
```

```
{
return (o.principalAmount*o.rateOfInterest*o.timePeriod)/100;
}
```

eg106.js

```
function getNextItemCode()
{
return getNextItemCode.code++;
}
getNextItemCode.code=1;
console.log(getNextItemCode());
console.log(getNextItemCode());
console.log(getNextItemCode());
console.log(getNextItemCode());
console.log(getNextItemCode());
```

eg107.js

```
function sameer()
{
sameer.x=10;
sameer.doIt=function(){
console.log("Just do it");
};
}
function tony()
{
tony.x=20;
tony.doIt=function(){
console.log("Just do it Tony's way");
};
}
console.log(sameer.x);
console.log(tony.x);
sameer.doIt();
tony.doIt();
```

eg108.js

```
function sameer()
{
sameer.x=10;
sameer.doIt=function(){
console.log("Just do it");
};
}
function tony()
{
tony.x=20;
tony.doIt=function(){
console.log("Just do it Tony's way");
};
}
sameer();
tony();
console.log(sameer.x);
console.log(tony.x);
sameer.doIt();
tony.doIt();
```

eg109.js

```
var sameer=(function()
{
var k={};
k.x=10;
k.doIt=function(){
```

```
console.log("Just do it");
};
return k;
}());
var tony=(function()
{
var g={};
g.x=20;
g.doIt=function(){
console.log("Just do it Tony's way");
};
return g;
}());
console.log(sameer.x);
console.log(tony.x);
sameer.doIt();
tony.doIt();
```

```
eg110.js
var x=10;
function ramesh()
{
var x=20;
function tony()
{
return x;
}
return tony();
}
console.log(ramesh());
```

```
eg111.js
var x=10;
function ramesh()
{
var x=20;
function tony()
{
return x;
}
return tony;
}
console.log(ramesh());
```

```
eg112.js
var nextItemCode=(function()
{
var code=0;
return function() { return ++code; }
```

```
})();  
console.log(nextItemCode());  
console.log(nextItemCode());  
console.log(nextItemCode());  
console.log(nextItemCode());
```

eg113.js

```
function Bulb()  
{  
var w=0;  
return {  
get Wattage() { return w; },  
set Wattage(e) { w=e; }  
};  
}
```

```
var b=Bulb();  
b.Wattage=60;  
console.log(b.Wattage);  
var c=Bulb();  
console.log(c.Wattage);  
c.Wattage=100;  
console.log(c.Wattage);  
console.log(b.Wattage);
```

eg114.js

```
function Bulb(e)  
{  
this.w=e;  
}  
function Toy(e)  
{  
this.p=e;  
}
```

```
var b=new Bulb(100);  
var t=new Toy(50);  
console.log(b instanceof Bulb);  
console.log(t instanceof Bulb);  
console.log(b instanceof Toy);  
console.log(t instanceof Toy);
```

eg115.js

```
function Bulb(e)  
{  
this.w=e;  
this.printBrand=function()  
{  
console.log("Philips");
```

```
};  
}  
var b=new Bulb(60);  
console.log(b);  
console.log(Bulb);  
console.log(b.constructor);  
console.log(b.prototype);  
console.log(Bulb.prototype);  
Bulb.prototype.setWattage=function(e)  
{  
this.w=e;  
};  
Bulb.prototype.getWattage=function()  
{  
return this.w;  
};  
console.log(b);  
var b2=new Bulb();  
console.log(b2);  
console.log(Bulb.prototype);
```

```
eg116.js
var x;
var y;
x=1;
cool: while(x<=5)
{
y=1;
while(y<=5)
{
console.log(x,y);
if(y==3) break cool;
y++;
}
console.log("Ujjain");
x++;
}
console.log("end");
```

```
eg117.js
var bulb1={
w: 0
};
var bulb2={
w: 0
};
function setWattage(w)
{
this.w=w;
}
g=setWattage.bind(bulb1);
g(60);
g=setWattage.bind(bulb2);
g(100);
console.log(bulb1.w);
console.log(bulb2.w);
```

```
eg118.js
function getPower(p,num)
{
var x=1;
for(var e=1;e<=p;e++)
{
x*=num;
}
return x;
}
console.log(getPower(2,3));
console.log(getPower(3,4));
```

eg119.js

```
function getPower(p,num)
{
var x=1;
for(var e=1;e<=p;e++)
{
x*=num;
}
return x;
}
var square=getPower.bind(null,2);
console.log(square(5));
var cube=getPower.bind(null,3);
console.log(cube(5));
```

eg120.js

```
function createPrimeNumberFunction()
{
var parameter1="x";
var functionDefinition="var y,z;for(z=0,y=2;y<=x/2;y++){if(x%y==0){z=1;break;}return z==0;}";
return new Function(parameter1,functionDefinition);
}
var isPrimeNumber=createPrimeNumberFunction();
for(var e=10;e<=50;e++)
{
if(isPrimeNumber(e)) console.log(e);
}
}
```

eg121.js

```
var rectangle={};
rectangle.length=10;
rectangle.breadth=2;
var box=Object.create(rectangle);
box.height=3;
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
rectangle.length=22;
rectangle.breadth=12
console.log("-----");
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
```

eg122.js

```
var rectangle={};
rectangle.length=10;
rectangle.breadth=2;
function EmptyConstructor()
```

```
{
}
EmptyConstructor.prototype=rectangle;
var box=new EmptyConstructor();
box.height=3;
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
```

notepad eg123.js

```
console.log(Object.create);
```

eg124.js

```
var rectangle={};
rectangle.length=10;
rectangle.breadth=2;
function extend(base)
{
if(Object.create) return Object.create(base);
function EmptyConstructor() {}
EmptyConstructor.prototype=rectangle;
return new EmptyConstructor();
}
var box=extend(rectangle);
box.height=3;
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
```

eg125.js

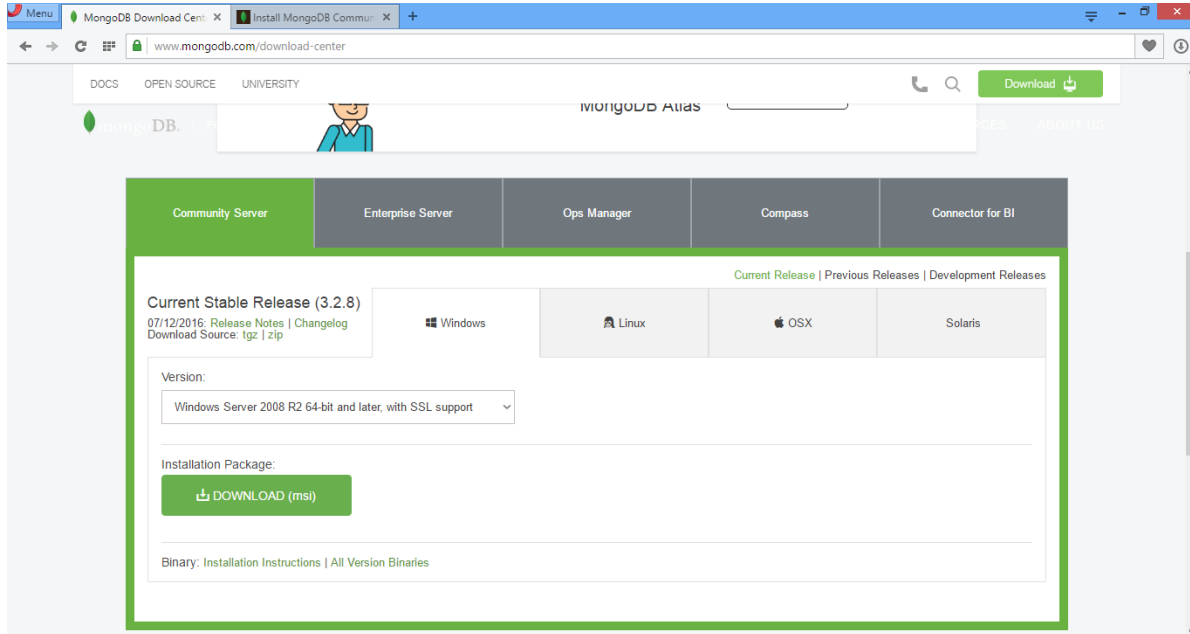
```
var rectangle={};
rectangle.length=10;
rectangle.breadth=2;

function extend(base)
{
if(Object.create) return Object.create(base);
function EmptyConstructor() {}
EmptyConstructor.prototype=rectangle;
return new EmptyConstructor();
}
var box=extend(rectangle);
box.height=3;
box.length=50;
box.breadth=4;
console.log("-----box data -----");
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
```

```
console.log("-----rectangle data -----");
console.log(rectangle.length);
console.log(rectangle.breadth);
rectangle.length=100;
rectangle.breadth=75;
console.log("-----box data-----");
console.log(box.length);
console.log(box.breadth);
console.log(box.height);
console.log("-----rectangle data-----");
console.log(rectangle.length);
console.log(rectangle.breadth);
```

MongoDB – NoSQL Database

Download MongoDB 3.2.8. Use site and option according to following UI



Install Mongo DB. Select Complete features option and install.

It will be installed under program files folder.

On c:\ create c:\data\db folder structure.

Add [c:\Program Files\MongoDB\Server\3.2\bin](#) to PATH environment variable.

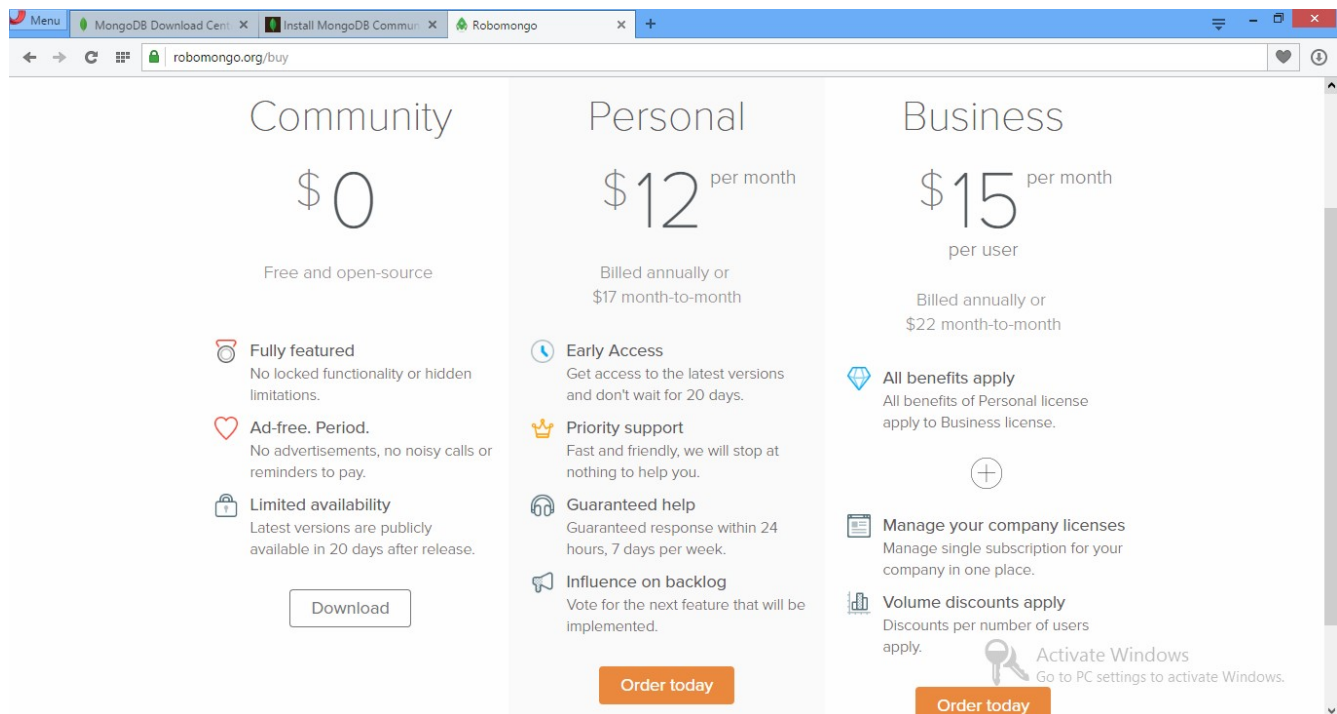
Open command prompt and run (mongod) to start the server.

If the server starts and starts listening on port 27017 then everything is running fine.

Press ctrl C to stop the server.

Robomongo – The MongoDB GUI Client

Now download RoboMongo community version. The UI is as follows



Install the robomongo, the mongodb client. Now go to command prompt and start the mongo db server by running mongod.

Then start the robomongo client and connect to the mongo db server running on local machine and listening on port number 27017.

Right click the root node and select create database and provide a name of your choice (java6pmttsdb)

expand the newly created database node and under the collections node add a new collection by the name of students. Do so by right clicking the collections node.

Then right click on students node and select insert document options, a window will appear with multi line textbox to accept record in JSON form. Feed some records with different attributes.

MongoDB Node JS – Native Driver

Now in the nodeeg folder create a folder named as mongodbeg

Now we need the MongoDB Native Driver for Node js. Go to command prompt, Your system should be connected to internet. We will use Node Package Manager. Go to your projects root folder, in our case the c:\nodeeg\mongodbeg folder and type

```
npm install mongodb
```

now wait for some time. The package manager will install necessary modules by downloading them from appropriate repositories. The node_modules folder will get created in the working directory. Remember if you are creating another project folder and you require to use the mongodb driver then you will have to run `npm install mongodb` from that folder. This is how the modern system to install dependencies work. You may not like it initially but this way you will be getting the latest and bug fixed dependencies.

```
Mongodbeg1.js (The connection test)
var mongodb=require("mongodb");
var mongoClient=mongodb.MongoClient;
var url="mongodb://localhost:27017/test";
mongoClient.connect(url,function(err,db){
if(err)
{
console.log("Error : ",err);
}
else
{
console.log("Connection established");
db.close();
}
});
```

Run the above script and you should see the connection established message.

```
Mongodbeg2.js
var mongodb=require("mongodb");
var mongoClient=mongodb.MongoClient;
var url="mongodb://localhost:27017/test";
mongoClient.connect(url,function(err,db){
if(err)
{
console.log("Error ",err);
return;
}
var student1={
roll: 108,
name: "Sunil",
hobbies: ["Swimming","Movies","Music"]
};
var student2={ roll: 109, name: "Suresh", nationality: "Indian" };
var collection=db.collection("students");
collection.insert([student1,student2],function(err,result){
if(err) console.log("Insert error ",err);
else console.log(result);
});
db.close();
});
```

Run the script and use robomongo to see if the records were inserted

```
Mongodbeg3.js
function errorHandler(err)
{
console.log("Error : ",err);
}
var mongodb=require("mongodb");
var MongoClient=mongodb.MongoClient;
var url="mongodb://localhost:27017/test";
MongoClient.connect(url,function(err,db){
if(err) errorHandler(err); else
var collection=db.collection("students");
collection.find().toArray(function(err,docs){
console.log(docs);
console.log("-----");
for(var e in docs)
{
console.log(docs[e]);
}
});
db.close();
});
```

```
eg126.js
var readline=require("readline");
var ioInterface=readline.createInterface({
input: process.stdin,
output: process.stdout
});
ioInterface.question("Cool ",function(choice){
console.log(choice);
var x={choice : Number(choice)};
console.log(x);
ioInterface.close();
process.stdin.destroy();
});
```

eg127.js

```
var readline=require("readline");
var ioInterface=readline.createInterface(process.stdin,process.stdout);
ioInterface.question("xvxvxx",function(c){
console.log(c);
ioInterface.close();
process.stdin.destroy();
});
```

eg128.js

```
var readline=require("readline");
var ioInterface=readline.createInterface(process.stdin,process.stdout);
ioInterface.setPrompt("$");
ioInterface.prompt();
ioInterface.on('line',function(t){
if(t=="bye") ioInterface.close(); else
{
console.log("type kara : "+t);
ioInterface.prompt();
}
});
ioInterface.on('close',function(){
process.exit(0);
});
```

eg129.js

```
console.log("good");
setTimeout(function(){console.log("bad")},10000);
```

eg130.js

```
var readline=require("readline");
var ioInterface=readline.createInterface(process.stdin,process.stdout);
ioInterface.setPrompt("$");
ioInterface.prompt();
setTimeout(function(){
ioInterface.on('line',function(t){
if(t=="bye") ioInterface.close(); else
{
console.log("type kara : "+t);
ioInterface.prompt();
}
});
ioInterface.on('close',function(){
process.exit(0);
});},10000);
/*
```

keep feeding for 20 seconds, one by one and after some time
you should see output in desired way
*/

eg131.js

```
function EventManager()
{
var eventHandlers=[];
this.setEventHandler=function(event,handler)
{
if(!handler) return;
var eventObject=eventHandlers.find(function(e){return e.event===event;});
if(eventObject)
{
eventObject.handler=handler;
}
else
{
var eventObject={};
eventObject.event=event;
eventObject.handler=handler;
eventHandlers.push(eventObject);
}
}
this.getEvent=function(event)
{
return eventHandlers.find(function(e){return e.event===event;});
}
```

```
}
}

function SomeModule()
{
var events=["initialized","end"];
var eventManager=new EventManager();
this.on=function(event,handler){
if(!events.find(function(e){return e==event;})) return;
eventManager.setEventHandler(event,handler);
};

var object=this;
function initialize()
{
// did some work
raiseEvent("initialized",[object,"Initialized"]);
}
this.start=function()
{
for(var i=1;i<=10;i++)
{
console.log(i);
}
raiseEvent("end",[object,"Some Module Ends","Printed 1 to 10"]);
}
function raiseEvent(event,args)
{
var eventObject=eventManager.getEvent(event);
if(eventObject) eventObject.handler.apply(null,args);
}
setTimeout(initialize,1000);
}
console.log("Constructor ends");
var someModule=new SomeModule();
someModule.on("initialized",function(sender,event){
console.log("In Initialized Event : ",event);
someModule.start();
});
someModule.on("end",function(sender,event,message){
console.log("In end event : ",event,message);
});
}

eg132.js
function EventManager()
{
var eventHandlers=[];
```

```
this.setEventHandler=function(event,handler,callAsync)
{
if(!handler) return;
var eventObject=eventHandlers.find(function(e){return e.event===event;});
if(eventObject)
{
eventObject.handler=handler;
eventObject.callAsync=callAsync;
}
else
{
var eventObject={};
eventObject.event=event;
eventObject.handler=handler;
eventObject.callAsync=callAsync;
eventHandlers.push(eventObject);
}
}
this.getEvent=function(event)
{
return eventHandlers.find(function(e){return e.event===event;});
}
}
```

```
function SomeModule()
{
var events=["initialized","end"];
var eventManager=new EventManager();
this.on=function(event,handler,callAsync){
if(!events.find(function(e){return e===event;})) return;
if(!callAsync) callAsync=false;
eventManager.setEventHandler(event,handler,callAsync);
};
var object=this;
function initialize()
{
// did some work
raiseEvent("initialized",[object,"Initialized"]);
}
this.start=function()
{
for(var i=1;i<=10;i++)
{
console.log(i);
}
raiseEvent("end",[object,"Some Module Ends","Printed 1 to 10"]);
}
}
```

```
function raiseEvent(event,args)
{
var eventObject=eventManager.getEvent(event);
if(eventObject)
{
if(eventObject.callAsync)
{
setTimeout(function(){
eventObject.handler.apply(null,args);
},1);
}
else
{
eventObject.handler.apply(null,args);
}
}
}
setTimeout(initialize,1000);
}
console.log("Constructor ends");
var someModule=new SomeModule();
someModule.on("initialized",function(sender,event){
console.log("In Initialized Event : ",event);
someModule.start();
},true);
someModule.on("end",function(sender,event,message){
console.log("In end event : ",event,message);
});
```

eg133.js

```
function Bulb()
{
this.wattage=0;
this.setWattage=function(w)
{
this.wattage=w;
return this;
}
this.getWattage=function()
{
return this.wattage;
}
}
var k=new Bulb();
console.log(k.setWattage(60).getWattage());
```

http1.js

```
var httpServerFactory=require("http");
```

```
httpServerFactory.createServer(function(request,response){
response.writeHead(200,{"content-type":"text/html"});
response.end("Cool response");
}).listen(3000);
console.log("http server is listening on port 3000");
```

```
http2.js
var httpServerFactory=require("http");
var httpServer=httpServerFactory.createServer();
httpServer.on('request',function(request,response){
response.writeHead(200,{"content-type":"text/html"});
response.end("Another cool response");
});
httpServer.listen(3000);
console.log("Server is listening on port 3000");
```

index.html required in context to http3.js

```
<html>
<head>
<title>A first foot step</title>
</head>
<body>
<h1>My first foot steps towards node js http server programming.</h1>
<a href='Chapter1'>Next step</a>
</body>
</html>
```

```
http3.js
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var server=httpServerFactory.createServer();
server.on('request',function(request,response){
response.writeHead(200,{"content-type":"text/html"});
fileSystemUtilities.createReadStream('index.html').pipe(response);
});
server.listen(3000);
console.log("http server is listening on port number 3000");
```

```
GenericJSONCRUDFactory.js
module.exports={
entities:[
{name: "student", key:"rollNumber"},
{name: "employee",key:"employeeId"}
],
isValidEntity: function(entity)
{
return this.entities.find(function(e){return entity===e.name;});
},
mongo: {
db: require("mongodb"),
```

```
client: function(){
return this.db.MongoClient;
},
url:"mongodb://localhost:27017/test"
},
performOperation: function(object,callback)
{
var jql=object.jql;
var prts=jql.split(" ");
var command=prts[0];
var entityName=prts[1];
console.log("Command name : "+command);
console.log("Entity name : "+entityName);
var entity=this.isValidEntity(entityName);
if(!entity)
{
callback({"success":false,"error":"Invalid entity name : "+entityName},null);
return;
}
this[command](entity,object.object,callback);
},
select: function(entity,object,callback)
{
this.mongo.client().connect(this.mongo.url,function(err,db){
if(err)
{
callback({"success":false,"error":err},null);
return;
}
var collection=db.collection(entity.name+"Collection");
collection.find().toArray(function(err,docs){
if(err)
{
db.close();
callback({"success":false,"error":err},null);
return;
}
if(docs.length==0)
{
callback({"success":false,"error":"No records."},null);
db.close();
return;
}
var json={"success":true,"result": entity.name,"responseType":"json"};
json[entity.name]=docs.map(function(o){delete o._id; return o;});
callback(null,json);
return;
}
```

```
});
});
},
insert: function(entity,object,callback)
{
this.mongo.client().connect(this.mongo.url,function(err,db){
if(err)
{
callback({"success":false,"error":err},null);
return;
}
var collection=db.collection(entity.name+"Collection");
collection.find({rollNumber: object[entity.key]}).toArray(function(err,docs){
if(err)
{
db.close();
callback({"success":false,"error":err},null);
return;
}
if(docs.length>0)
{
callback({"success":false,"error":entity.key+" : "+object[entity.key]+" exists."},null);
db.close();
return;
}
collection.insert(object,function(err,result){
db.close();
if(err)
{
callback({"success":false,"error":err},null);
}
else
{
callback(null,{"success":true,"successMessage":entity.name+"
added.", "result":"successMessage", "responseType":"json"});
}
});
});
},
update:function(collection,object)
{
},
delete: function(collection,object)
{
}
```



```
}  
http4.js  
var crudFactory=require("./GenericJSONCRUDFactory.js");  
var httpServerFactory=require("http");  
var fileSystemUtilities=require("fs");  
var urlUtilities=require("url");  
var pathUtilities=require("path");  
var httpServer=httpServerFactory.createServer();  
httpServer.on('request',function(request,response){  
var requestURI=urlUtilities.parse(request.url).pathname;  
console.log(requestURI);  
var currentWorkingFolder=process.cwd();  
console.log(currentWorkingFolder);  
var fullPath=pathUtilities.join(currentWorkingFolder,requestURI);  
console.log(fullPath);  
// In this example we will not be dealing with the fullPath, I  
// We will use it later on.  
console.log("Method : "+request.method);  
response.setHeader('Access-Control-Allow-Origin', '*');  
response.setHeader('Access-Control-Allow-Credentials', true);  
response.setHeader('Access-Control-Allow-Methods', 'POST, GET, PUT, DELETE, OPTIONS');  
response.setHeader('Access-Control-Allow-Headers', 'Content-Type');  
if(request.method=="OPTIONS")  
{  
response.end();  
}  
if (request.method == 'POST')  
{  
console.log("Request data : "+request.toString());  
var buffer = "";  
request.setEncoding('utf8');  
request.on('data', function(chunk){ buffer += chunk });  
request.on('end', function() {  
var jqLObject=JSON.parse(buffer);  
console.log("Request received : "+JSON.stringify(jqLObject));  
crudFactory.performOperation(jqLObject,function(err,result){  
response.writeHead(200,{"Content-Type":"application/json" });  
if(err)  
{  
console.log("Response : "+JSON.stringify(err));  
response.end(JSON.stringify(err)+"\n");  
}  
else  
{  
console.log("Response : "+JSON.stringify(result));  
response.end(JSON.stringify(result)+"\n");
```

```

}
});
});
} // if POST condition ends
});
httpServer.listen(3000);
console.log("http Server is listening on port 3000");

```

The Client Module for the Above Server Module

ObjectBrowser.html

```

<!doctype html>
<html lang='en' ng-app='TMApplication'>
<head>
<meta charset='utf-8'>
<title>TIFFIT - Obect Browser</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap-theme.min.css">
<script src="jquery/jquery-2.1.3.min.js"></script>
<script src="bootstrap/js/bootstrap.min.js"></script>
<script src='angular/angular.min.js'></script>
<script src='js/Model.js'></script>
<script src='service/Service.js'></script>
<script src='js/ObjectBrowserController.js'></script>
<style>
  p{text-align: center;}
  html, body{height:100%;}
  #content{
    min-height:100%;
    height:auto !important;
    height:100%;
    /* Negative indent header and footer by its height */
    margin:-80px auto -60px;
    /* Pad bottom by header and footer height */
    padding:80px 0 60px;
    font-size:14pt;
  }
  /* Set the fixed height of the header here */
  #header{height:5%; font-size:16pt; margin:2px auto 0px; }
  /* Set the fixed height of the footer here */
  #footer{height:5%; font-size: 10pt; color:green;font-weight: bold;}
  .query { height : 30vh }
  .result { height: 60vh }
  .queryFullHeight { height: 100% }
  .resultFullHeight { height: 100% }
</style>
</head>

```

```

<body ng-controller='objectBrowserController'>
  <div id="header">
    <div class="container">
      <div class="row">
        <div class="col-md-12">Entity browser</div>
      </div>
    </div>
  </div>
  <div id="content">
    <div class="container" >
      <div class="row query">
        <div class="col-md-12 queryFullHeight" >
          <textarea style="width:100%;height:100%;border: 1px solid gray" ng-model='query' ng-
keydown='queryKeyDownEvent()'></textarea>
        </div>
      </div>
      <div class="row result">
        <div class="col-md-12 resultFullHeight">
          <div class='col-md-12' id='resultDivision' style='width:100%; height:100%;border: 1px solid
gray; overflow: scroll'>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div id="footer">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <span ng-show='statusBarContent.length>0'>{{statusBarContent}}</span>
        </div>
      </div>
    </div>
  </div>
</body>
</html>

```

Model.js

```

var application=angular.module("TMApplication",[]);
var developmentMode=true;

```

service.js

```

application.service("databaseService",function($http){
this.sendRequest=function(requestJSON)
{
requestJSON.xcjsu23jxcjsd=new Date();
return $http({
method: 'POST',

```

```

url:'http://localhost:3000',
data: JSON.stringify(requestJSON),
dataType: 'json'
});
}
});
function errorHandler(responseError){
    switch (responseError.status) {

        default:
            var form=document.getElementById("siteFallbackForm");
            form.submit();
    }
}
}

```

ObjectBrowserController.js

```

application.controller("objectBrowserController",function($scope,databaseService){
$scope.query="";
$scope.statusBarContent="";
$scope.clear=function()
{
$scope.query="";
$scope.statusBarContent="";
document.getElementById("resultDivision").innerHTML="";
}
$scope.queryKeyDownEvent=function()
{
var keyCode=event.keyCode;
if((keyCode==10 || keyCode==13) && event.ctrlKey)
{
$scope.fire();
}
if((keyCode==10 || keyCode==13) && event.altKey)
{
$scope.clear();
}
}
$scope.fire=function()
{
$scope.statusBarContent="";
if($scope.query.trim().length==0) return;
var startTime=new Date();
var requestJSON;
try
{
requestJSON=JSON.parse($scope.query);

```

```
}catch(e)
{
var endTime=new Date();
var difference=(endTime-startTime);
$scope.statusBarContent="Request processed in "+difference+" milliseconds.";
document.getElementById("resultDivision").innerHTML=e;
return;
}
var operation=databaseService.sendRequest(requestJSON);
operation.then(function(response){
var endTime=new Date();
var difference=(endTime-startTime);
//alert(JSON.stringify(response.data));
$scope.statusBarContent="Request processed in "+difference+" milliseconds.";
if(response.data.success)
{
if(response.data.responseType=="json")
{
document.getElementById("resultDivision").innerHTML=JSON.stringify(response.data[response.data.result]);
}
if(response.data.responseType=="html")
{
document.getElementById("resultDivision").innerHTML=(response.data[response.data.result]);
}
} // success ends
else
{
document.getElementById("resultDivision").innerHTML=JSON.stringify(response.data);
}
},function(){
alert('Unable to connect to server, contact Administrator');
});
});
```

http5.js

```
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var pathUtilities=require("path");
//var qs = require('querystring');
var httpServer=httpServerFactory.createServer();
httpServer.on('request',function(request,response){
var requestURI=urlUtilities.parse(request.url).pathname;
console.log(requestURI);
var currentWorkingFolder=process.cwd();
```

```
console.log(currentWorkingFolder);
var fullPath=pathUtilities.join(currentWorkingFolder,requestURI);
console.log(fullPath);
// In this example we will not be dealing with the fullPath, l
// We will use it later on.
console.log("Method : "+request.method);
// the following code will work in case of GET type request
var queryStringParameters = urlUtilities.parse(request.url,true).query;
console.log(queryStringParameters["nm"]);
console.log(queryStringParameters["ct"]);
});
httpServer.listen(3000);
console.log("http Server is listening on port 3000");
```

```
http6
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var pathUtilities=require("path");
var qs = require('querystring');
var httpServer=httpServerFactory.createServer();
httpServer.on('request',function(request,response){
var requestURI=urlUtilities.parse(request.url).pathname;
console.log(requestURI);
var currentWorkingFolder=process.cwd();
console.log(currentWorkingFolder);
var fullPath=pathUtilities.join(currentWorkingFolder,requestURI);
console.log(fullPath);
console.log("Method : "+request.method);
if(requestURI=="/form")
{
response.writeHead(200,{"content-type":"text/html"});
fileSystemUtilities.createReadStream('abcd.html').pipe(response);
return;
}
// the following code will work in case of POST type request only
if(requestURI=="/saveData")
{
var buffer="";
request.on('data', function (data) {
buffer +=data;
});
request.on('end',function() {
var queryStringParameters = qs.parse(buffer);
console.log(queryStringParameters["nm"]);
console.log(queryStringParameters["ct"]);
console.log(queryStringParameters["sx"]);
```

```
        console.log(queryStringParameters["ag"]);
    });
}
});
httpServer.listen(3000);
console.log("http Server is listening on port 3000");
```

eg134.js

// Assignment completed by Shubham Nagota

```
var data="Whatever is cool <?serverJS coolStuff ?>. we like it <?serverJS whateverStuff ?>. It is really cool";
var array=[];
var p=/<?serverJS(.*?)\?>/g;
var k=p.exec(data);
var x=0;
while(k!=null)
{
array.push(k[1].trim());
k=p.exec(data);
x++;
}
for(x=0;x<array.length;x++) console.log(array[x]);
```

eg135.js

// Assignment completed by Sanidhya Saraswat

```
var data="Whatever is cool <?serverJS var x=10; client.print(x); ?>. we like it <?serverJS whateverStuff ?>. It is really cool";
var array=[];
var p=/<\?serverJS[\s\S]*?\?>/g;
var match;
var startIndex,endIndex;
var initialIndex=0;
while(match=p.exec(data))
{
startIndex=p.lastIndex-match[0].length;
if(startIndex!=initialIndex)
{
array.push({"text":data.substring(initialIndex,startIndex),"type": "html"});
}
endIndex=p.lastIndex-1;
initialIndex=endIndex+1;
array.push({"text":data.substring(startIndex+10,endIndex-1).trim(),"type": "serverJS"});
}
if(data.length!=endIndex+1)
{
array.push({"text":data.substring(initialIndex,data.length),"type":"html"});
}
for(x=0;x<array.length;x++) console.log(array[x]);
```

```
http7.js
var webApplicationFactory=require("./http7webapplication.js");
var webApplication=webApplicationFactory.createWebApplication();
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var pathUtilities=require("path");
var qs = require('querystring');
var httpServer=httpServerFactory.createServer();
function processGetRequest(request,response)
{
console.log("Get Request : "+request.requestURI);
webApplication.process(request,response);
}
function processPostRequest(request,response)
{
console.log("Post Request : "+request.requestURI);
webApplication.process(request,response);
}
function requestHandler(request,response)
{
request.requestURI=urlUtilities.parse(request.url).pathname;
if(request.method.toUpperCase()=="GET")
{
var queryStringParameters=urlUtilities.parse(request.url,true).query;
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
processGetRequest(request,response);
}
if(request.method.toUpperCase()=="POST")
{
var buffer="";
request.on('data', function (data) {
buffer +=data;
});
request.on('end',function() {
var queryStringParameters = qs.parse(buffer);
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
processPostRequest(request,response);
});
}
}
httpServer.on('request',requestHandler);
httpServer.listen(3000);
console.log("http Server is listening on port 3000");
```

```
utils.js
module.exports={
responseCodes: {
ok: 200,
badRequest: 400,
notAuthorized: 401,
forbidden: 403,
notFound: 404,
internalServerError:500
}
}
}
}
}

http7webapplication.js
var mime=require("mime");
var fileSystemUtilities = require('fs');
var pathUtilities=require("path");
var utils=require('./utils.js');
var responseCodes=utils.responseCodes;
function WebApplication()
{
var publicPath="http7\\public\\";
var resources=[
{uri : "/", file: "default.html", mimeType: "text/html"},
{uri : "/favicon.ico", file: "images\\favicon.png"}
];
var resourceCache=[];
function getResource(uri)
{
var resource;
resource=resourceCache.find(function(e){return e.uri==uri;});
if(resource) return resource;
resource=resources.find(function(e){return e.uri==uri;});
if(resource)
{
resource.file=publicPath+resource.file;
resource.responseCode=responseCodes.ok;
if(!resource.mimeType)
{
resource.mimeType=mime.lookup(resource.file);
}
}
}
else
{
var relativePath=pathUtilities.join(publicPath,uri.substring(1));
var absolutePath=pathUtilities.join(process.cwd(),pathUtilities.join(publicPath,uri.substring(1)));
console.log(absolutePath);
try
```

```
{
console.log("Looking for : "+absolutePath);
stats=fileSystemUtilities.lstatSync(absolutePath);
if(stats.isFile())
{
resource={};
resource.uri=uri;
resource.file=publicPath+uri.substring(1);
resource.responseCode=responseCodes.ok;
resource.mimeType=mime.lookup(absolutePath);
resources.push(resource);
}
else
{
resource={};
resource.responseCode=responseCodes.notFound;
}
}catch (e)
{
resource={};
resource.responseCode=responseCodes.notFound;
}
}
resourceCache.push(resource);
return resource;
}
this.process=function(request,response)
{
var resource=getResource(request.requestURI);
console.log(resource);
if(resource.responseCode===responseCodes.ok)
{
response.writeHead(200,{"content-type": resource.mimeType });
console.log('resource file : '+resource.file);
fileSystemUtilities.createReadStream(resource.file).pipe(response);
}
if(resource.responseCode===responseCodes.notFound)
{
response.writeHead(responseCodes.notFound,{"content-type":"text/plain"});
response.end();
}
}
}
}
module.exports={
createWebApplication:function()
{
return new WebApplication();
}
```

```
}  
};  
default.html (location http7\public)  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>one.com</title>  
</head>  
<body>  
<img src='images/logo.jpg'/><br/>  
<a href='contactForm.html'>Contact Form</a>  
</body>  
</html>
```

2 more files are required to be kept in http7\public\images folder
favicon.png and logo.png

```
eg1.c  
#include<stdio.h>  
int main(int n,char *p[])  
{  
printf("Cool 1 : %s\n",p[1]);  
printf("Cool 2 : %s\n",p[2]);  
fputs(" God is great",stderr);  
return 0;  
}
```

compile the above code and eg1.exe should get created

```
eg138.js  
var execFile=require('child_process').execFile;  
var child=execFile('eg1',[10,20],function(error,stdout,stderr){  
console.log(typeof(stdout));  
if(error)  
{  
console.log(error);  
}  
else  
{  
console.log("Content of stdout : "+stdoutString);  
console.log("Content of stderr : "+stderrString);  
}  
});
```

```
sample1.technology  
<html>  
<head>  
<?serverJS  
client.print("<title>Great</title>");  
>
```

```
</head>
<body>
"Cool stuff starts here"
<?serverJS
var i=0;
while(i<=10)
{
client.print(i);
i++;
}
?>
</body>
</html>
```

eg137.js

// Assignment completed by Burhanuddin Pithawala

```
var fs=require("fs");
var sourceFile="sample1.technology";
var fileToBeGenerated="sample1.js";
var data=fs.readFileSync(sourceFile).toString();
console.log(data);
console.log("-----");
var array=[];
var p=/<?\?serverJS[\s\S]*?\?>/g;
var match;
var startIndex,endIndex;
var initialIndex=0;
while(match=p.exec(data))
{
startIndex=p.lastIndex-match[0].length;
if(startIndex!=initialIndex)
{
array.push({"text":data.substring(initialIndex,startIndex),"type": "html"});
}
endIndex=p.lastIndex-1;
initialIndex=endIndex+1;
array.push({"text":data.substring(startIndex+10,endIndex-1).trim(),"type": "serverJS"});
}
if(data.length!=endIndex+1)
{
array.push({"text":data.substring(initialIndex,data.length),"type":"html"});
}
var ws=fs.createWriteStream(fileToBeGenerated);
ws.write("var client={\r\n");
ws.write("print: function(data)\r\n");
ws.write("{\r\n");
ws.write("console.log(data);\r\n");
```

```

ws.write("}\r\n");
ws.write("};\r\n");
array.forEach(function(e,i,a){
if(e.type=="html")
{
console.log("Writing html : "+e.text);
ws.write("client.print(\""+e.text.replace(/\r\n/g, "\\n").replace(/\"/g, "\\\"")+"\");"+e.text);
}
else
{
console.log("Writing js "+e.text);
ws.write(e.text+"\r\n");
}
});
ws.end();
var execFile=require('child_process').execFile;
var child=execFile('node',[fileToBeGenerated],function(error,stdout,stderr){
console.log(typeof(stdout));
if(error)
{
console.log(error);
}
else
{
console.log(stdout);
}
});

```

The problem with above code is that it runs the generated script in another node instance whereas it should run in the same process in which the eg137.js is running

eg139.js

```

var splitPattern=/<?\serverJS[\s\S]*?\?>/g;
var data="Whatever is cool <?serverJS var x=10; client.print(x); ?>. we like it <?serverJS
whateverStuff ?>. It is really cool";
var array=data.split(splitPattern);
for(var i=0;i<array.length;i++)
{
console.log(i, ' --- ',array[i]);
}

```

eg141.js

```

module.exports=function(client)
{
client.print("God is great");
}

```

eg142.js

```

var whatever=require('./eg141.js');
function Client()

```

```
{
this.print= console.log;
}
whatever(new Client());

```

```
eg136.js
var newLineChar=require("os").EOL; // I have used this
var newLineCharPattern=/[\r\n]+/; // I have used this also
var path=require("path");
var fileSystemUtilities=require("fs");
var sourceFile="sample1.technology";
var fileToBeGenerated="sample1.js";
var data=fileSystemUtilities.readFileSync(sourceFile).toString();
var splitPattern=/(<?serverJS[\s\S]*?>|{\{[\s\S]*?\})/g;
var searchPattern=^"/g;
var array=data.split(splitPattern);
array.forEach(function(e,i){
var object={};
if(e.startsWith("<?serverJS"))
{
object.data=e.substring(10,e.length-2);
object.type="serverJS";
}
else if(e.startsWith("{") && e.endsWith("}")){
object.data=e.substring(2,e.length-2);
object.type="regularExpression";
} else
{
object.data=e.split(newLineCharPattern);
object.type="text";
object.data.forEach(function(e,i,a){
a[i]=e.replace(searchPattern,"\\");
});
}
array[i]=object;
});
var fileDescriptor; // what we call file handle/file pointer
fileDescriptor=fileSystemUtilities.openSync(path.join(process.cwd(),fileToBeGenerated),"w");
fileSystemUtilities.writeSync(fileDescriptor,"module.exports=function(client)+newLineChar);
fileSystemUtilities.writeSync(fileDescriptor,"{"+newLineChar);
array.forEach(function(element){
if(element.type==="serverJS")
{
if(element.data.trim().length>0)
{
fileSystemUtilities.writeSync(fileDescriptor,element.data+newLineChar);
}
}
}
```

```
}
else if(element.type=="regularExpression")
{
if(element.data.trim().length>0)
{
fileSystemUtilities.writeSync(fileDescriptor,"client.print("+element.data+");"+newLineChar);
}
}
else
{
element.data.forEach(function(line){
if(line.trim().length>0)
{
fileSystemUtilities.writeSync(fileDescriptor,"client.print(\""+line+"\");"+newLineChar);
}
});
});
fileSystemUtilities.writeSync(fileDescriptor,"}"+newLineChar);
fileSystemUtilities.closeSync(fileDescriptor);
function client()
{
this.print=console.log;
}
var requiredFile="./"+fileToBeGenerated;
var generatedFunction=require(requiredFile);
generatedFunction(new client());


---


var fs=require("fs");
var folderName="Rakesh";
var exists=fs.existsSync(folderName);
if(exists)
{
var stats=fs.lstatSync(folderName);
if(stats.isFile())
{
fs.unlinkSync(folderName);
exists=false;
}
}
if(exists===false)
{
fs.mkdirSync(folderName);
}
console.log("cool");


---


eg144.js
var fs=require("fs");
```

```
function createFolderSync(folderName)
{
var exists=fs.existsSync(folderName);
if(exists)
{
var stats=fs.lstatSync(folderName);
if(stats.isFile())
{
fs.unlinkSync(folderName);
exists=false;
}
}
if(exists===false)
{
fs.mkdirSync(folderName);
}
}
function createFoldersSync(folders)
{
var array=folders.split("\\");
var folder="";
array.forEach(function(e){
folder+=e;
createFolderSync(folder);
folder+="\\";
});
}
var folders="Rakesh\\Suresh\\Mahesh\\Gopal\\Lalit\\Pulkit";
createFoldersSync(folders);
```

Date 22/8/2016

eg145.js

```
var fs = require('fs'),
path = require('path');
```

```
function getDirectories(srcpath) {
return fs.readdirSync(srcpath).filter(function(file) {
return fs.statSync(path.join(srcpath, file)).isDirectory();
});
}
```

```
console.log(getDirectories("."));
```

```
console.log(Array.isArray(getDirectories(".")));
```

eg146.js

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



```
function isDirectory(folderName)
{
if(fs.existsSync(folderName))
{
if(fs.lstatSync(folderName).isDirectory()) return true;
else return false;
} else return false;
}
```

```
console.log(isDirectory("xyz"));
```

eg147.js

```
var fs = require('fs');
function isFile(fileName)
{
if(fs.existsSync(fileName))
{
if(fs.lstatSync(fileName).isFile()) return true;
else return false;
} else return false;
}
```

```
console.log(isFile("xyz"));
```

eg148.js

```
var fs=require("fs");

function getJSONFromFile(fileName)
{
var r={};
try
{
r.json=JSON.parse(fs.readFileSync(fileName,"utf-8"));
r.success=true;
} catch(e)
{
r.error=e;
r.success=false;
}
return r;
}
var k=getJSONFromFile("web.json");
console.log(k.success);
if(k.success)
{
var object=k.json;
console.log(object.contextName);
```

```
}  
else  
{  
console.log("Some error : ",k.error);  
}
```

```
eg149.js  
var requestURI="/Gopal/";  
var contextName=requestURI.split("/")[1];  
console.log(contextName);
```

Create a folder named as TMJSWebServer
in it create the following js files

```
utils.js  
var fs=require("fs");  
var path=require("path");  
module.exports={  
responseCodes: {  
ok: 200,  
badRequest: 400,  
notAuthorized: 401,  
forbidden: 403,  
notFound: 404,  
internalServerError:500  
},  
createFolderSync:function(folderName)  
{  
var exists=fs.existsSync(folderName);  
if(exists)  
{  
var stats=fs.lstatSync(folderName);  
if(stats.isFile())  
{  
fs.unlinkSync(folderName);  
exists=false;  
}  
}  
if(exists===false)  
{  
fs.mkdirSync(folderName);  
}  
},  
createFoldersSync: function(folders)  
{  
var array=folders.split(/[\\]/);
```

```
var folder="";
var object=this;
array.forEach(function(e){
  folder+=e;
  object.createFolderSync(folder);
  folder+="\\";
});
},
getDirectories:function(srcpath)
{
  return fs.readdirSync(srcpath).filter(function(file) {
    return fs.statSync(path.join(srcpath, file)).isDirectory();
  });
},
isDirectory:function(folderName)
{
  if(fs.existsSync(folderName))
  {
    if(fs.lstatSync(folderName).isDirectory()) return true;
    else return false;
  } else return false;
},
isFile:function(fileName)
{
  if(fs.existsSync(fileName))
  {
    if(fs.lstatSync(fileName).isFile()) return true;
    else return false;
  } else return false;
},
getJSONFromFile:function(fileName)
{
  var r={};
  try
  {
    r.json=JSON.parse(fs.readFileSync(fileName,"utf-8"));
    r.success=true;
  }catch(e)
  {
    r.error=e;
    r.success=false;
  }
  return r;
}
}
```

```
var mime=require("mime");
var fileSystemUtilities = require('fs');
var pathUtilities=require("path");
var utils=require('./utils.js');
var responseCodes=utils.responseCodes;
function WebApplication(contextName,folderName)
{
this.contextName=contextName;
this.folderName=folderName;
var publicPath=this.folderName+"\\public\\";
var privatePath=this.folderName+"\\private\\";
/*var resources=[
{uri : "/", file: "default.html", mimeType: "text/html"},
{uri : "/favicon.ico", file: "images\\favicon.png"}
];*/
var resourceCache=[];
function getResource(uri)
{
var resource;
resource=resourceCache.find(function(e){return e.uri===uri;});
if(resource) return resource;
resource=resources.find(function(e){return e.uri===uri;});
if(resource)
{
resource.file=publicPath+resource.file;
resource.responseCode=responseCodes.ok;
if(!resource.mimeType)
{
resource.mimeType=mime.lookup(resource.file);
}
}
else
{
var relativePath=pathUtilities.join(publicPath,uri.substring(1));
var absolutePath=pathUtilities.join(process.cwd(),pathUtilities.join(publicPath,uri.substring(1)));
console.log(absolutePath);
try
{
console.log("Looking for : "+absolutePath);
stats=fileSystemUtilities.lstatSync(absolutePath);
if(stats.isFile())
{
resource={};
resource.uri=uri;
resource.file=publicPath+uri.substring(1);
resource.responseCode=responseCodes.ok;
resource.mimeType=mime.lookup(absolutePath);
```

```
resources.push(resource);
}
else
{
resource={};
resource.responseCode=responseCodes.notFound;
}
} catch (e)
{
resource={};
resource.responseCode=responseCodes.notFound;
}
}
resourceCache.push(resource);
return resource;
}
this.process=function(request,response)
{
var resource=getResource(request.requestURI);
console.log(resource);
if(resource.responseCode===responseCodes.ok)
{
response.writeHead(200, {"content-type": resource.mimeType });
console.log('resource file : '+resource.file);
fileSystemUtilities.createReadStream(resource.file).pipe(response);
}
if(resource.responseCode===responseCodes.notFound)
{
response.writeHead(responseCodes.notFound, {"content-type":"text/plain"});
response.end();
}
}
this.equals=function(contextName)
{
return this.contextName===contextName;
}
}
module.exports={
createWebApplication:function(contextName,folderName)
{
return new WebApplication(contextName,folderName);
}
};
}
TMJSWebServer.js
console.log("Starting server");
var serverSideTechnologyExtension="technology";
```

```
var webApplicationFactory=require("./WebApplicationProcessor.js");
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var defaultWebServerResource="index.html";
var applicationConfigurationFile="web.json";
var pathUtilities=require("path");
var qs = require('querystring');
var utils=require("./utils.js");
var webApplications=[];
// folder scanning starts
var foldersToAnalyze=utils.getDirectories(".");

foldersToAnalyze.forEach(function(folderName){
var privateFolder=folderName+"\private";
var publicFolder=folderName+"\public";
var privateFolderExists=utils.isDirectory(privateFolder);
var publicFolderExists=utils.isDirectory(publicFolder);
if(privateFolderExists && publicFolderExists)
{
var configurationJSON={};
var configurationFile=privateFolder+"\ "+applicationConfigurationFile;
var configurationFileExists=utils.isFile(configurationFile);
if(configurationFileExists)
{
var configurationJSONWrapper=utils.getJSONFromFile(configurationFile);
if(configurationJSONWrapper.success)
{
configurationJSON=configurationJSONWrapper.json;
}
}
else
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : "+configurationJSONWrapper.error);
console.log("----- END -----");
return;
}
}
var contextName=folderName;
if(configurationJSON.contextName)
{
contextName=configurationJSON.contextName;
}
else
{
configurationJSON.contextName=contextName;
}
}
```

```
var webApplication=webApplicationFactory.createWebApplication(contextName,folderName);
webApplication.configuration=configurationJSON;
var dynamicJSFolder="generatedJS\\"+webApplication.folderName;
utils.createFoldersSync(dynamicJSFolder);
webApplications.push(webApplication);
}
});
// display list of sites
console.log("List of web applications");
console.log("-----");
webApplications.forEach(function(e){console.log(e.contextName);});
console.log("-----");
```

```
var httpServer=httpServerFactory.createServer();
function processGetRequest(request,response)
{
console.log("Get Request : "+request.requestURI);
webApplication.process(request,response);
}
function processPostRequest(request,response)
{
console.log("Post Request : "+request.requestURI);
webApplication.process(request,response);
}
function requestHandler(request,response)
{
request.requestURI=urlUtilities.parse(request.url).pathname;
if(request.requestURI=='/')
{
fileSystemUtilities.createReadStream(defaultWebServerResource).pipe(response);
return;
}
request.contextName=request.requestURI.split("/")[1];
var webApplication=webApplications.find(function(e){
if(e.equals(request.contextName)) return true; else return false;
});
if(!webApplication)
{
return;
}

if(request.method.toUpperCase()=="GET")
{
var queryStringParameters=urlUtilities.parse(request.url,true).query;
```

```
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
processGetRequest(request,response);
}
if(request.method.toUpperCase()=="POST")
{
    var buffer="";
    request.on('data', function (data) {
        buffer +=data;
    });
    request.on('end',function(){
        var queryStringParameters = qs.parse(buffer);
        if(!queryStringParameters) queryStringParameters={};
        request.queryStringParameters=queryStringParameters;
        processPostRequest(request,response);
    });
}
}
}
httpServer.on('request',requestHandler);
httpServer.listen(3000);
console.log("http Server is listening on port 3000");
```

move the http7 folder in the newly created folder by the name of jsone.com
in jsone.com\private create the following web.json file
web.json

```
{
  "resources":[
    {"uri" : "/", "file": "default.html", "mimeType": "text/html"},
    {"uri" : "/favicon.ico", "file": "images\\favicon.png"}
  ],
  "contextName": "cooljsone.com"
}
```

Date : 25/8/2016

```
console.log("Starting server");
var rootSite="ROOT";
var serverSideTechnologyExtension="technology";
var webApplicationFactory=require("./WebApplicationProcessor.js");
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var applicationConfigurationFile="web.json";
var pathUtilities=require("path");
var qs = require('querystring');
var utils=require("./utils.js");
var webApplications=[];
var rootExists=false;
// folder scanning starts
```



```
var foldersToAnalyze=utils.getDirectories(".");
foldersToAnalyze.forEach(function(folderName){
var privateFolder=folderName+"\\private";
var publicFolder=folderName+"\\public";
var privateFolderExists=utils.isDirectory(privateFolder);
var publicFolderExists=utils.isDirectory(publicFolder);
if(privateFolderExists && publicFolderExists)
{
var configurationJSON={};
var configurationFile=privateFolder+"\\"+applicationConfigurationFile;
var configurationFileExists=utils.isFile(configurationFile);
if(configurationFileExists)
{
var configurationJSONWrapper=utils.getJSONFromFile(configurationFile);
if(configurationJSONWrapper.success)
{
configurationJSON=configurationJSONWrapper.json;
}
}
else
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : "+configurationJSONWrapper.error);
console.log("----- END -----");
return;
}
}
if(configurationJSON.contextName &&
utils.isValidContextName(configurationJSON.contextName)==false)
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : Invalid context name");
console.log("----- END -----");
return;
}

var contextName="/";
if(configurationJSON.contextName)
{
contextName=configurationJSON.contextName;
} else if(folderName!=rootSite)
{
contextName="/" + folderName;
}
var webApplication=webApplicationFactory.createWebApplication(contextName, folderName);
webApplication.configuration=configurationJSON;
var dynamicJSFolder="generatedJS\\" + webApplication.folderName;
```

```
utils.createFoldersSync(dynamicJSFolder);
webApplications.push(webApplication);
if(folderName==rootSite) rootExists=true;
}
});
// if root does not exist
if(!rootExists)
{
var webApplication=webApplicationFactory.createRootWebApplication();
webApplications.push(webApplication);
}
// display list of sites
console.log("List of web applications");
console.log("-----");
webApplications.forEach(function(e){console.log(e.contextName);});
console.log("-----");
var httpServer=httpServerFactory.createServer();
function processGetRequest(request,response)
{
console.log("Get Request : "+request.requestURI);
webApplication.process(request,response);
}
function processPostRequest(request,response)
{
console.log("Post Request : "+request.requestURI);
webApplication.process(request,response);
}
function requestHandler(request,response)
{
request.requestURI=urlUtilities.parse(request.url).pathname;
if(request.requestURI=='/')
{
fileSystemUtilities.createReadStream(defaultWebServerResource).pipe(response);
return;
}
request.contextName=request.requestURI.split("/")[1];
var webApplication=webApplications.find(function(e){
if(e.equals(request.contextName)) return true; else return false;
});
if(!webApplication)
{

return;
}

if(request.method.toUpperCase()=="GET")
```

```

{
var queryStringParameters=urlUtilities.parse(request.url,true).query;
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
processGetRequest(request,response);
}
}
if(request.method.toUpperCase()=="POST")
{
    var buffer="";
    request.on('data', function (data) {
        buffer +=data;
    });
    request.on('end',function(){
        var queryStringParameters = qs.parse(buffer);
        if(!queryStringParameters) queryStringParameters={};
        request.queryStringParameters=queryStringParameters;
        processPostRequest(request,response);
    });
}
}
}
}
httpServer.on('request',requestHandler);
httpServer.listen(3000);
console.log("http Server is listening on port 3000");

```

WebApplicationProcess.js (Minor change)

```

module.exports={
createWebApplication:function(contextName,folderName)
{
return new WebApplication(contextName,folderName);
},
createRootWebApplication:function()
{
return new WebApplication();
}
};

```

Date : 01/9/2016

eg150.js

```

var fs=require("fs");
console.log(process.cwd());
var p=fs.realpathSync(process.cwd());
console.log(fs.realpathSync("http"));
console.log(p);

```

eg151.js

```

var k={};
// /golu /tony /golu /bholu /kalu /tony /sam /doggy /tony
var c=[["/golu","GOLU"],["/tony","TONY"],["/golu","GOLU"],["/bholu","BHOLU"],,
["/sam","SAM"],["/doggy","DOGGY"],["/tony","TONY"]];

```

```
c.forEach(function(e){
  if(k[e[0]])
  {
    k[e[0]].push(e[1]);
  }
  else
  {
    k[e[0]]=[e[1]];
  }
});
for(var a in k)
{
  if(k[a].length==1)
  {
    console.log(k[a]);
  }
}
```

eg152.js

```
var k={};
//golu /tony /golu /bholu /kalu /tony /sam /doggy /tony
var c=["/golu","/tony","/golu","/bholu","/sam","/doggy","/tony"];
var m=[];
c.forEach(function(e){
  if(k[e])
  {
    k[e]=function(){};
  }
  else
  {
    k[e]=function(){
      m.push(e);
      console.log(e);
    };
  }
});
for(var a in k)
{
  k[a]();
}
console.log("-----");
console.log(m);
```

eg153.js

```
var k;
k="/";
var firstWord=k.split("/")[1];
console.log("Cool [",firstWord,]",",firstWord.length,"/"+firstWord);
```

```
k="/Good/Food";
firstWord=k.split("/")[1];
console.log("Cool [" ,firstWord, "]",firstWord.length);
k="/Whatever";
firstWord=k.split("/")[1];
console.log("Cool [" ,firstWord, "]",firstWord.length);
```

```
TMJSWebServer.js
console.log("Starting server");
var rootFolder="ROOT";
var rootFolderExists=false;
var rootConfigurationJSON;
var serverSideTechnologyExtension="technology";
var webApplicationFactory=require("./WebApplicationProcessor.js");
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var applicationConfigurationFile="web.json";
var pathUtilities=require("path");
var qs = require('querystring');
var utils=require("./utils.js");
var webApplications={};
var sitesJSON={};
// folder scanning starts
var foldersToAnalyze=utils.getDirectories(".");
foldersToAnalyze.forEach(function(folderName){
var privateFolder=folderName+"\\private";
var publicFolder=folderName+"\\public";
var privateFolderExists=utils.isDirectory(privateFolder);
var publicFolderExists=utils.isDirectory(publicFolder);
if(privateFolderExists && publicFolderExists)
{
var configurationJSON={};
var configurationFile=privateFolder+"\\"+applicationConfigurationFile;
var configurationFileExists=utils.isFile(configurationFile);
if(configurationFileExists)
{
var configurationJSONWrapper=utils.getJSONFromFile(configurationFile);
if(configurationJSONWrapper.success)
{
configurationJSON=configurationJSONWrapper.json;
}
else
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : "+configurationJSONWrapper.error);
console.log("-----END -----");
```

```

return;
}
}
if(configurationJSON.contextName &&
utils.isValidContextName(configurationJSON.contextName)==false)
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : Invalid context name");
console.log("----- END -----");
return;
}
var contextName="/";
if(configurationJSON.contextName)
{
if(!isValidContextName(configurationJSON.contextName))
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration - Invalid context name :
"+configurationJSON.contextName);
console.log("----- END -----");
return;
}
contextName=configurationJSON.contextName;
} else if(folderName!=rootFolder)
{
contextName="/" + folderName;
}

if(sitesJSON[contextName])
{
sitesJSON[contextName]=function(){};
}
else
{
sitesJSON[contextName]=function(){
var webApplication=webApplicationFactory.createWebApplication(contextName, folderName);
webApplication.configuration=configurationJSON;
var dynamicJSFolder="generatedJS\\" + webApplication.folderName;
utils.createFoldersSync(dynamicJSFolder);
webApplications[contextName]=webApplication;
};
}
if(folderName==rootFolder) rootFolderExists=true;
});
for(var s in sitesJSON)
{

```

```
sitesJSON[s]());
}
// if root does not exist
if(!rootFolderExists)
{
if(!(sitesJSON["/"]))
{
var webApplication=webApplicationFactory.createRootWebApplication();
webApplication["/"]=webApplication;
}
}
// display list of sites
console.log("List of web applications");
console.log("-----");
for(var wa in webApplications) console.log(webApplications[wa].contextName);
console.log("-----");
// removing all WebApplication instances whose context names are not unique
var httpServer=httpServerFactory.createServer();
function requestHandler(request,response)
{
request.requestURI=urlUtilities.parse(request.url).pathname;
var firstWord=request.requestURI.split("/")[1];
var contextName;
var webApplication;
if(firstWord.length==0)
{
contextName="/";
}
else
{
if((webApplications["/"+firstWord]))
{
contextName="/"+firstWord;
}
else
{
contextName="/";
}
}
webApplication=webApplications[contextName];
if(!(webApplication))
{
response.writeHead(404,{"content-type":"text/html"});
response.write("<html><head><title>TM Node Web Server/1.0 - Error report</title><STYLE><!--
H1 {font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY {font-
family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B {color : white;backgroun-
d-color : #0086b2;} HR {color : #0086b2;} --></STYLE> </head><body><h1>TM Node Web Server/1.0
```

```
- HTTP Status 404 - "+request.requestURI+"</h1><HR size='1' noshade><p><b>type</b> Status
report</p><p><b>message</b> <u>"+request.requestURI+"</u></p><p><b>description</b> <u>The
requested resource ("<u>"+request.requestURI+"</u>") is not available.</u></p><HR size='1'
noshade</body></html>");
response.end();
return;
}
request.contextName=contextName;
if(request.method.toUpperCase()=="GET")
{
var queryStringParameters=urlUtilities.parse(request.url,true).query;
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
webApplication.process(request,response);
}
if(request.method.toUpperCase()=="POST")
{
    var buffer="";
    request.on('data', function (data) {
        buffer +=data;
    });
    request.on('end',function() {
        var queryStringParameters = qs.parse(buffer);
        if(!queryStringParameters) queryStringParameters={};
        request.queryStringParameters=queryStringParameters;
        webApplication.process(request,response);
    });
}
}
}
httpServer.on('request',requestHandler);
httpServer.listen(3000);
console.log("http Server is listening on port 3000");
WebApplicationProcessor.js
var mime=require("mime");
var fileSystemUtilities = require('fs');
var pathUtilities=require("path");
var utils=require('./utils.js');
var responseCodes=utils.responseCodes;
function WebApplication(contextName,folderName)
{
if(contextName)
{
this.contextName=contextName;
this.folderName=folderName;
}
else
```



```
{
this.contextName="/";
this.folderName=null;
}
var publicPath=this.folderName+"\\public\\";
var privatePath=this.folderName+"\\private\\";
/*var resources=[
{uri : "/", file: "default.html", mimeType: "text/html"},
{uri : "/favicon.ico", file: "images\\favicon.png"}
];*/
var resourceCache=[];
function getResource(uri)
{
var resource;
resource=resourceCache.find(function(e){return e.uri==uri;});
if(resource) return resource;
resource=resources.find(function(e){return e.uri==uri;});
if(resource)
{
resource.file=publicPath+resource.file;
resource.responseCode=responseCodes.ok;
if(!resource.mimeType)
{
resource.mimeType=mime.lookup(resource.file);
}
}
else
{
var relativePath=pathUtilities.join(publicPath,uri.substring(1));
var absolutePath=pathUtilities.join(process.cwd(),pathUtilities.join(publicPath,uri.substring(1)));
console.log(absolutePath);
try
{
console.log("Looking for : "+absolutePath);
stats=fileSystemUtilities.lstatSync(absolutePath);
if(stats.isFile())
{
resource={};
resource.uri=uri;
resource.file=publicPath+uri.substring(1);
resource.responseCode=responseCodes.ok;
resource.mimeType=mime.lookup(absolutePath);
resources.push(resource);
}
else
{
resource={};
}
```

```
resource.responseCode=responseCodes.notFound;
}
} catch (e)
{
resource={};
resource.responseCode=responseCodes.notFound;
}
}
resourceCache.push(resource);
return resource;
}

this.process=function(request,response)
{
if(!folderName)
{
response.writeHead(500, {"content-type":"text/html"});
response.write("<html><head><title>TM Node Web Server Error Report</title><br><br><h1>HTTP
Status 500 - No Context configured to process this request</h1></body></html>");
response.end();
return;
}
request.resource=request.requestURI.substring(this.contextName.length);
var resourceJSON=getResource(request.resource);
if(resourceJSON.responseCode==responseCodes.ok)
{
/*response.writeHead(200, {"content-type": resource.mimeType });
console.log('resource file : '+resource.file);
fileSystemUtilities.createReadStream(resource.file).pipe(response);*/
}
if(resourceJSON.responseCode==responseCodes.notFound)
{
response.writeHead(responseCodes.notFound, {"content-type":"text/html"});
response.write("<html><head><title>TM Node Web Server/1.0 - Error report</title><STYLE><!--
H1 {font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY {font-
family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B {color : white;background-
color : #0086b2;} HR {color : #0086b2;} --></STYLE> </head><body><h1>TM Node Web Server/1.0
- HTTP Status 404 - "+request.resource+"</h1><HR size='1' noshade><p><b>type</b> Status
report</p><p><b>message</b> <u>"+request.resource+"</u></p><p><b>description</b> <u>The
requested resource (" +request.resource+" ) is not available.</u></p><HR size='1'
noshade></body></html>");
response.end();
return;
}
}
}
}
module.exports={
```

```
createWebApplication:function(contextName,folderName)
{
return new WebApplication(contextName,folderName);
},
createRootWebApplication:function()
{
return new WebApplication();
}
};
```

Date : 2/9/2016

TMJSWebServer.js

```
console.log("Starting server");
var rootFolder="ROOT";
var rootFolderExists=false;
var rootConfigurationJSON;
var serverSideTechnologyExtension="technology";
var webApplicationFactory=require("./WebApplicationProcessor.js");
var httpServerFactory=require("http");
var fileSystemUtilities=require("fs");
var urlUtilities=require("url");
var applicationConfigurationFile="web.json";
var pathUtilities=require("path");
var qs = require('querystring');
var utils=require("./utils.js");
var webApplications={};
var sitesJSON={};
// folder scanning starts
var foldersToAnalyze=utils.getDirectories(".");
foldersToAnalyze.forEach(function(folderName){
var privateFolder=folderName+"\\private";
var publicFolder=folderName+"\\public";
var privateFolderExists=utils.isDirectory(privateFolder);
var publicFolderExists=utils.isDirectory(publicFolder);
if(privateFolderExists && publicFolderExists)
{
var configurationJSON={};
var configurationFile=privateFolder+"\\"+applicationConfigurationFile;
var configurationFileExists=utils.isFile(configurationFile);
if(configurationFileExists)
{
var configurationJSONWrapper=utils.getJSONFromFile(configurationFile);
if(configurationJSONWrapper.success)
{
configurationJSON=configurationJSONWrapper.json;
}
}
else
```

```

{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : "+configurationJSONWrapper.error);
console.log("----- END -----");
return;
}
}
if(!(configurationJSON.resources)) configurationJSON.resources=[];
if(configurationJSON.contextName &&
utils.isValidContextName(configurationJSON.contextName)==false)
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration : Invalid context name");
console.log("----- END -----");
return;
}
var contextName="/";
if(configurationJSON.contextName)
{
if(!utils.isValidContextName(configurationJSON.contextName))
{
console.log("-----START-----");
console.log("Problems in "+folderName+" configuration - Invalid context name :
"+configurationJSON.contextName);
console.log("----- END -----");
return;
}
contextName=configurationJSON.contextName;
} else if(folderName!=rootFolder)
{
contextName="/" + folderName;
}

if(sitesJSON[contextName])
{
sitesJSON[contextName]=function(){};
}
else
{
sitesJSON[contextName]=function(){
var webApplication=webApplicationFactory.createWebApplication(contextName, folderName);
webApplication.configuration=configurationJSON;
console.log(webApplication.contextName, webApplication.configuration);
var dynamicJSFolder="generatedJS\\" + webApplication.folderName;
utils.createFoldersSync(dynamicJSFolder);
webApplications[contextName]=webApplication;
};
}

```

```
}
if(folderName==rootFolder) rootFolderExists=true;
}
});
for(var s in sitesJSON)
{
sitesJSON[s]();
}
// if root does not exist
if(!rootFolderExists)
{
if(!(sitesJSON["/"]))
{
var webApplication=webApplicationFactory.createRootWebApplication();
webApplication["/"]=webApplication;
}
}
// display list of sites
console.log("List of web applications");
console.log("-----");
for(var wa in webApplications) console.log(webApplications[wa].contextName);
console.log("-----");
// removing all WebApplication instances whose context names are not unique
var httpServer=httpServerFactory.createServer();
function requestHandler(request,response)
{
request.requestURI=urlUtilities.parse(request.url).pathname;
var firstWord=request.requestURI.split("/")[1];
var contextName;
var webApplication;
if(firstWord.length==0)
{
contextName="/";
}
else
{
if((webApplications["/"+firstWord]))
{
contextName="/"+firstWord;
}
else
{
contextName="/";
}
}
webApplication=webApplications[contextName];
if(!(webApplication))
```

```

{
response.writeHead(404, {"content-type": "text/html"});
response.write("<html><head><title>TM Node Web Server/1.0 - Error report</title><STYLE><!--
H1 {font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY {font-
family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B {color : white;background-
color : #0086b2;} HR {color : #0086b2;} --></STYLE> </head><body><h1>TM Node Web Server/1.0
- HTTP Status 404 - "+request.requestURI+"</h1><HR size='1' noshade><p><b>type</b> Status
report</p><p><b>message</b> <u>"+request.requestURI+"</u></p><p><b>description</b> <u>The
requested resource (" +request.requestURI+") is not available.</u></p><HR size='1'
noshade</body></html>");
response.end();
return;
}
request.contextName=contextName;
if(request.method.toUpperCase()=="GET")
{
var queryStringParameters=urlUtilities.parse(request.url,true).query;
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
webApplication.process(request,response);
}
if(request.method.toUpperCase()=="POST")
{
var buffer="";
request.on('data', function (data) {
buffer +=data;
});
request.on('end',function() {
var queryStringParameters = qs.parse(buffer);
if(!queryStringParameters) queryStringParameters={};
request.queryStringParameters=queryStringParameters;
webApplication.process(request,response);
});
}
}
}
httpServer.on('request',requestHandler);
httpServer.listen(3000);
console.log("http Server is listening on port 3000");

```

```

WebApplicationProcessor.js
var mime=require("mime");
var fileSystemUtilities = require('fs');
var pathUtilities=require("path");
var utils=require('./utils.js');
var responseCodes=utils.responseCodes;
var defaultResources=["index.html","index.htm","index.js","index.technology"];
function WebApplication(contextName,folderName)

```

```
{
if(contextName)
{
this.contextName=contextName;
this.folderName=folderName;
}
else
{
this.contextName="/";
this.folderName=null;
}
var publicPath=this.folderName+"\\public\\";
var privatePath=this.folderName+"\\private\\";
/*var resources=[
{uri : "/", file: "default.html", mimeType: "text/html"},
{uri : "/favicon.ico", file: "images\\favicon.png"}
];*/
var resourceCache=[];
var object=this;
function getResource(uri)
{
if(uri.length==0) uri="/";
if(uri=="/")
{
if(object.configuration.homePage)
{
return {
responseCode: responseCodes.redirect,
redirectTo: object.configuration.homePage
};
}
var defaultResource=object.configuration.resources.find(function(e){if(e.uri=="/") return true;});
if(defaultResource)
{
return {
responseCode: responseCodes.redirect,
redirectTo: defaultResource.file
};
}
var defaultResource=defaultResources.find(function(e){
return utils.isFile(publicPath+e);
});
if(defaultResource)
{
object.configuration.homePage=defaultResource;
return {
responseCode: responseCodes.redirect,
```

```
redirectTo: defaultResource
};
}
return {
responseCode: responseCodes.notFound,
};
}
console.log("Context name : "+object.contextName);
console.log("URI : ",uri);
console.log("Configuration : ",object.configuration);
var resource;
resource=resourceCache.find(function(e){return e.uri==uri;});
if(resource) return resource;
resource=object.configuration.resources.find(function(e){return e.uri==uri;});
if(resource)
{
resource.file=publicPath+resource.file;
resource.responseCode=responseCodes.ok;
if(!resource.mimeType)
{
resource.mimeType=mime.lookup(resource.file);
}
}
else
{
var relativePath=pathUtilities.join(publicPath,uri.substring(1));
console.log(relativePath);
var absolutePath=pathUtilities.join(process.cwd(),pathUtilities.join(publicPath,uri.substring(1)));
console.log(absolutePath);
try
{
console.log("Looking for : "+absolutePath);
stats=fileSystemUtilities.lstatSync(absolutePath);
if(stats.isFile())
{
resource={};
resource.uri=uri;
resource.file=publicPath+uri.substring(1);
resource.responseCode=responseCodes.ok;
resource.mimeType=mime.lookup(absolutePath);
object.configuration.resources.push(resource);
console.log("if main phasa");
}
else
{
resource={};
resource.responseCode=responseCodes.notFound;
```



```

console.log("else main phasa");
}
} catch (e)
{
console.log("catch main phasa ",e.message);
resource={};
resource.statusCode=responseCodes.notFound;
}
}
resourceCache.push(resource);
return resource;
}
this.process=function(request,response)
{
if(!folderName)
{
response.writeHead(500, {"content-type":"text/html"});
response.write("<html><head><title>TM Node Web Server Error Report</title><br><br><h1>HTTP
Status 500 - No Context configured to process this request</h1></body></html>");
response.end();
return;
}
request.resource=request.requestURI.substring(this.contextName.length);
var resourceJSON=getResource(request.resource);
if(resourceJSON.statusCode==responseCodes.redirect)
{
response.writeHead(302, {Location: this.contextName+"/"+resourceJSON.redirectTo});
response.end();
return;
}

if(resourceJSON.statusCode==responseCodes.ok)
{
response.writeHead(200, {"content-type": resourceJSON.mimeType });
console.log('resource file : '+resourceJSON.file);
fileSystemUtilities.createReadStream(resourceJSON.file).pipe(response);
}

if(resourceJSON.statusCode==responseCodes.notFound)
{
response.writeHead(responseCodes.notFound, {"content-type":"text/html"});
response.write("<html><head><title>cool foolTM Node Web Server/1.0 - Error
report</title><STYLE><!--H1 {font-family : sans-serif,Arial,Tahoma;color : white;background-color :
#0086b2;} BODY {font-family : sans-serif,Arial,Tahoma;color : black;background-color : white;}
B{color : white;background-color : #0086b2;} HR {color : #0086b2;} --></STYLE>
</head><body><h1>TM Node Web Server/1.0 - HTTP Status 404 - "+request.resource+"</h1><HR
size='1' noshade><p><b>type</b> Status report</p><p><b>message</b>

```

```
<u>"+request.resource+"</u></p><p><b>description</b> <u>The requested resource
("+request.resource+") is not available.</u></p><hr size='1' noshade></body></html>");
response.end();
return;
}
}
}
module.exports={
createWebApplication:function(contextName,folderName)
{
return new WebApplication(contextName,folderName);
},
createRootWebApplication:function()
{
return new WebApplication();
}
};
```

Lot of more can be done, it is just a start.