EmailTwitterFacebookLinkedInGoogle+

In our node js application, sometimes we have to execute functions either synchronously or asynchronously. To achieve so , we have a node js module **“async”** that helps us to push down our functions in an array, so that we can execute them as per the need of the project.

**Installation Part:**

Run this command to install the “async” module through **npm** (Node Package Manager)

$npm install async

Now this will install the async module in your Project.

After the installation, you can use the predefined functions of this modules by simply passing an array of functions to its methods.For that you need to require the async module in your application.

var async =require('async');

Now all you have to do is simply create an array of functions you want to call either synchronously or asynchronously. And pass this array to a desired method of async module.

Task[] = [<function 1>,<function 2>........<function n>];

Some of the methods of async module that we usually use to control the flow of the tasks are

* series
* parallel
* forever
* waterfall

Lets see a demonstration of how it works

**Code :**

var async =require('async');

var task=[];

task.push(function(cb){

setTimeout(function(){console.log("first");

cb(null,"first");

},3000)

});

task.push(function(cb){

setTimeout(function(){console.log("second");

cb(null,"second");}

,1000)

});

task.push(function(cb){

setTimeout(function(){console.log("final");

cb(null,"final");}

,6000)

});

async.series(task,function(err,data){console.log(data)});

async.parallel(task,function(err,data){console.log(data)});

The series method calls each function of the array task, one by one in a form of series.This makes the execution flow synchronous in nature.

##Output of series :

first

second

final

[ 'first', 'second', 'final' ]

The parallel method calls each function of the array task, simultaneously in a form of parallel execution.This makes the execution flow asynchronous in nature.

##Output of parallel :

second

first

final

[ 'first', 'second', 'final' ]