Chapter 8

Node JS Timer

Node.js Timer

Node.js Timer functions are global functions. You don't need to use require() function in order to use timer functions. Let's see the list of timer functions.

setImmediate(): It is used to execute setImmediate.

setInterval(): It is used to define a time interval.

setTimeout(): ()- It is used to execute a one-time callback after delay milliseconds.

//...

Clear timer functions:

clearImmediate(immediateObject): It is used to stop an immediateObject, as created by setImmediate

clearInterval(intervalObject): It is used to stop an intervalObject, as created by setInterval

clearTimeout(timeoutObject): It prevents a timeoutObject, as created by setTimeout

//...

Node.js Timer setInterval() Example

This example will set a time interval of 1000 millisecond and the specified comment will be displayed after every 1000 millisecond until you terminate.

File: timer1.js

```
setInterval(function() {
 console.log("setInterval: Hey! 1 millisecond completed!..");
}, 1000);
```

Open Node.js command prompt and run the following code:

```
node timer1.js
```

//...

File: timer5.js

```
var i =0;
console.log(i);
setInterval(function(){
i++;
console.log(i);
}, 1000);
```

Open Node.js command prompt and run the following code:

```
node timer5.js
```

//...

Node.js Timer setTimeout() Example

File: timer1.js

```
setTimeout(function() {
console.log("setTimeout: Hey! 1000 millisecond completed!..");
}, 1000);
```

Open Node.js command prompt and run the following code:

```
node timer1.js
```

This example shows time out after every 1000 millisecond without setting a time interval. This example uses the recursion property of a function.

File: timer2.js

```
var recursive = function () {
   console.log("Hey! 1000 millisecond completed!..");
```

```
    setTimeout(recursive,1000);
}
recursive();
```

Open Node.js command prompt and run the following code:

```
node timer2.js
```

//..

Node.js setInterval(), setTimeout() and clearTimeout()

Let's see an example to use clearTimeout() function.

File: timer3.js

```
function welcome () {
  console.log("Welcome to JavaTpoint!");
}
var id1 = setTimeout(welcome,1000);
var id2 = setInterval(welcome,1000);
clearTimeout(id1);
//clearInterval(id2);
```

Open Node.js command prompt and run the following code:

```
node timer3.js
```

You can see that the above example is recursive in nature. It will terminate after one step if you use ClearInterval.

//..

Node.js setInterval(), setTimeout() and clearInterval()

Let's see an example to use clearInterval() function.

File: timer3.js

```
function welcome () {
  console.log("Welcome to JavaTpoint!");
}
var id1 = setTimeout(welcome,1000);
var id2 = setInterval(welcome,1000);
//clearTimeout(id1);
clearInterval(id2);
```

Open Node.js command prompt and run the following code:

```
node timer3.js
```