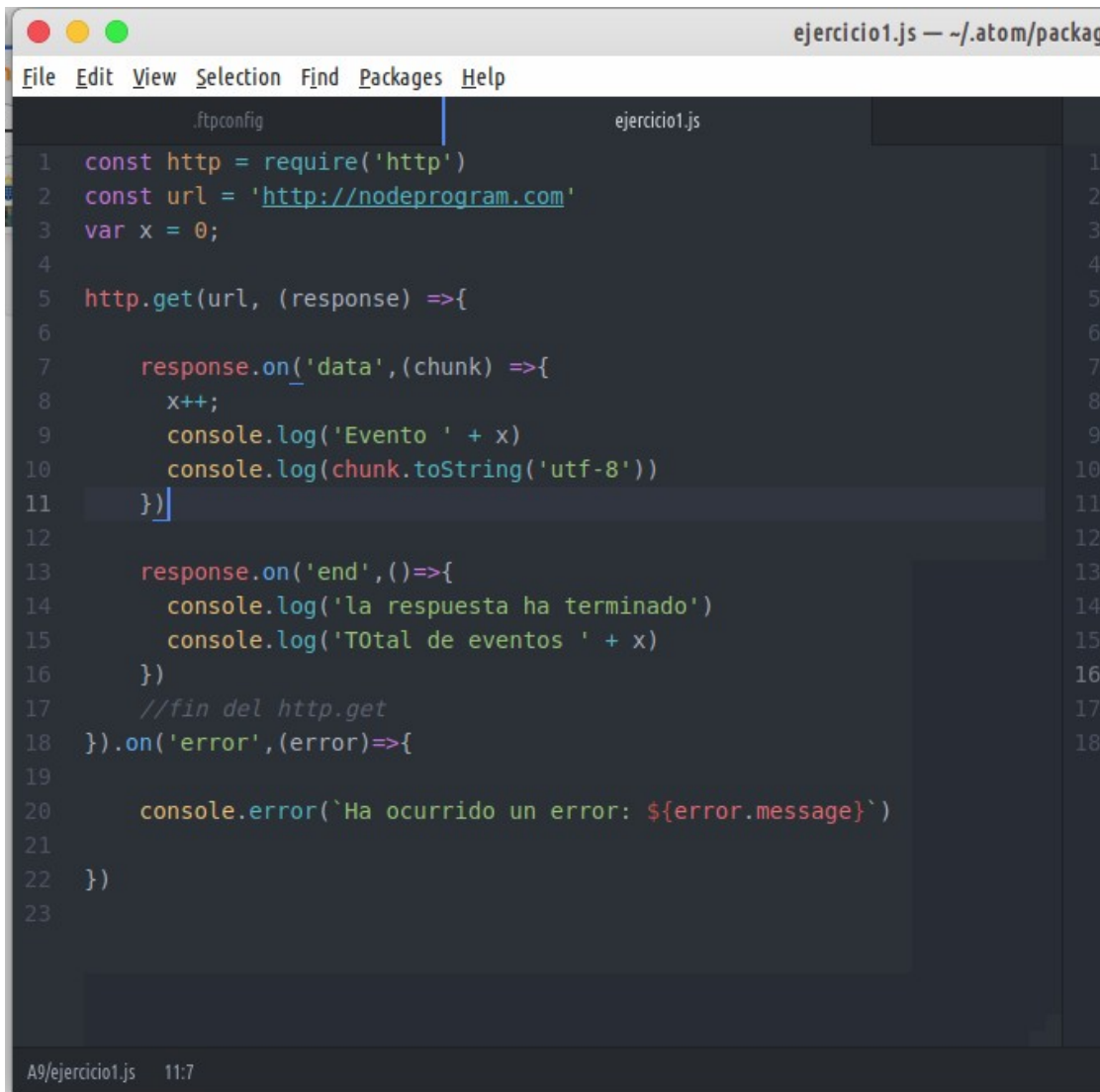


A9. Cliente HTTP

Ejercicio 1

Escribe un programa que reciba como argumento una URL y realice una petición HTTP GET a la misma. Luego, deberá imprimir por consola el contenido de cada evento "data" de la petición. Antes de imprimir cada chunk debe incluir el texto "Evento X", donde X es el número de evento.



```
1  const http = require('http')
2  const url = 'http://nodeprogram.com'
3  var x = 0;
4
5  http.get(url, (response) =>{
6
7      response.on('data', (chunk) =>{
8          x++;
9          console.log('Evento ' + x)
10         console.log(chunk.toString('utf-8'))
11     })
12
13     response.on('end', ()=>{
14         console.log('la respuesta ha terminado')
15         console.log('Total de eventos ' + x)
16     })
17     //fin del http.get
18 }).on('error', (error) =>{
19
20     console.error(`Ha ocurrido un error: ${error.message}`)
21
22 })
23
```

Observamos como nos devuelve el número de eventos.

```
root@debian-ivan:/home/ivan/node/A9# node ejercicio1.js
Evento 1
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- META DATA -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimum-scale=1.0, maximum-scale=1.0">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">

  <meta name="description" content="San Francisco, Los Angeles, and New York City courses on React.js, Node.js, MongoDB, Express.js and JavaS
  <title>React.js, Node.js, MongoDB, Express.js and JavaScript Courses</title>

  <link rel="apple-touch-icon-precomposed" sizes="144x144" href="assets/images/ico/apple-touch-icon-144-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="114x114" href="assets/images/ico/apple-touch-icon-114-p
Evento 2
recomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/images/ico/apple-touch-icon-72-precomposed.png">
  <link rel="apple-touch-icon-precomposed" href="assets/images/ico/apple-touch-icon-57-precomposed.png">
  <link rel="shortcut icon" href="assets/images/ico/favicon.ico">

  <!-- STYLESHEETS -->
  <link rel="stylesheet" href="assets/style/bootstrap.css" type="text/css" />
  <link rel="stylesheet" href="assets/style/main.css" type="text/css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flipclock/0.7.8/flipclock.min.css">

```

Evento 3

```

      <input type="email" class="span12" placeholder="Email" required="required">
      <textarea rows="10" class="span12" required="required"></textarea>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form> -->
  </div>
Evento 3
  <!-- End Contact form -->
  <!-- Contact information -->
  <div class="span5 offset1">
    <h5>Contact Us</h5>
    <!-- <div class="media">
      <div class="social-icons sicon-white pull-left">
        <a href="#" class="sicon-phone"><i>Phone</i></a>
      </div>
      <div class="media-body smallFontBy08">
        (+1) 502 964-0393 | (+1) 502 964-0393 (FAX)
      </div>
    </div> -->

    <div class="media">
      <div class="social-icons sicon-white pull-left">

```

Total de eventos

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/flipclock/0.7.8/flipclock.min.js"></script>
<script>
  var dest = new Date([2017, 09, 09]);
  var diff = dest.getTime() / 1000 - new Date().getTime() / 1000;
  window.clockz = $('#clock').FlipClock(diff, {
    clockFace: 'DailyCounter',
    countdown: true
  });
</script>
</body>

</html>
la respuesta ha terminado
Total de eventos 3
root@debian-ivan:/home/ivan/node/A9# █
```

Ejercicio 2

Escribe un programa que realice una solicitud HTTP GET a una URL proporcionada como el primer argumento de línea de comando. Recoge todos los datos del servidor y luego escribe dos líneas en la consola (utiliza `stdout` y no `console.log`):

- La primera línea debe ser un número entero que muestre el número de caracteres recibidos del servidor.
- La segunda línea debe contener la cadena completa de caracteres enviados por el servidor.

```
File Edit View Selection Find Packages Help
.eftpconfig ejercicio1.js ejercicio2.js
1 const http = require('http')
2 const url = 'http://nodeprogram.com'
3 var x = 0;
4
5 http.get(url, (response) =>{
6   let rawData = ''
7   response.on('data', (chunk) =>{
8     rawData += chunk
9     x++;
10
11   })
12
13   response.on('end', ()=>{
14
15     process.stdout.write("Caracteres totales: " + rawData.length + "\n")
16     process.stdout.write(rawData)
17     //console.log('Total de eventos ' + x)
18   })
19   //fin del http.get
20 }).on('error', (error)=>{
21
22   console.error(`Ha ocurrido un error: ${error.message}`)
23
24 })
25
```

```
Actividades  PuTTY SSH Client  mié oct 10, 18:52:50
ivan@debian-ivan: ~

root@debian-ivan:/home/ivan/node/A9# node ejercicio2.js
Caracteres totales: 52952
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- META DATA -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimum-scale=1.0, maximum-scale=1.0">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">

  <meta name="description" content="San Francisco, Los Angeles, and New York City courses on React.js, Node.js, MongoDB, Express.js and JavaScript Courses">
  <title>React.js, Node.js, MongoDB, Express.js and JavaScript Courses</title>

  <link rel="apple-touch-icon-precomposed" sizes="144x144" href="assets/images/ico/apple-touch-icon-144-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="114x114" href="assets/images/ico/apple-touch-icon-114-precomposed.png">
  <link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/images/ico/apple-touch-icon-72-precomposed.png">
  <link rel="apple-touch-icon-precomposed" href="assets/images/ico/apple-touch-icon-57-precomposed.png">
  <link rel="shortcut icon" href="assets/images/ico/favicon.ico">
```

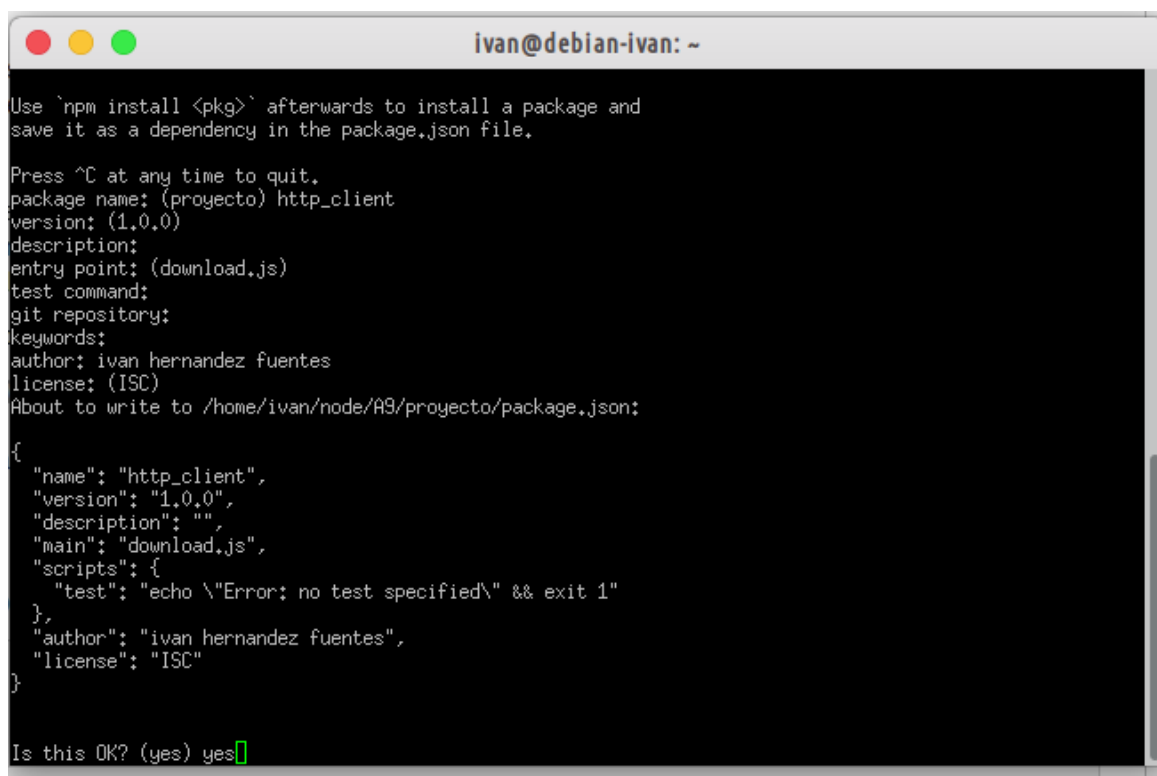
Ejercicio 3

Este último ejercicio es similar a los anteriores. Hay que escribir un programa que realice una solicitud HTTP GET a una URL proporcionada por línea de comando pero en esta ocasión escribiremos la página obtenida en un fichero que se llamará file.html, dentro de una carpeta cuyo nombre generaremos aleatoriamente con el módulo UUID instalable mediante npm.

Antes de empezar a programar crea un proyecto específico para este ejercicio siguiendo los siguientes pasos:

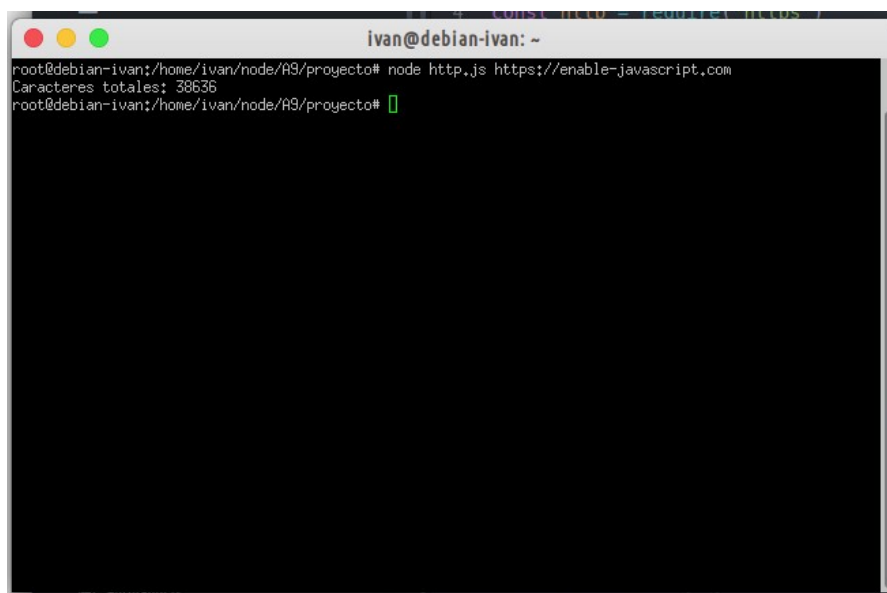
1. Crea la carpeta del proyecto y crea un archivo download.js en ella.
2. Crea un archivo package.json usando npm init.
3. Instala el módulo uuid para generar timestamps aleatorios que usaremos como nombres de carpeta distintos donde almacenaremos la página a descargar.

Creamos nuestro package.json



```
ivan@debian-ivan: ~  
Use `npm install <pkg>` afterwards to install a package and  
save it as a dependency in the package.json file.  
  
Press ^C at any time to quit.  
package name: (proyecto) http_client  
version: (1.0.0)  
description:  
entry point: (download.js)  
test command:  
git repository:  
keywords:  
author: ivan hernandez fuentes  
license: (ISC)  
About to write to /home/ivan/node/A9/proyecto/package.json:  
  
{  
  "name": "http_client",  
  "version": "1.0.0",  
  "description": "",  
  "main": "download.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "ivan hernandez fuentes",  
  "license": "ISC"  
}  
  
Is this OK? (yes) yes
```

Ejecutamos el script y agregamos el parámetro url



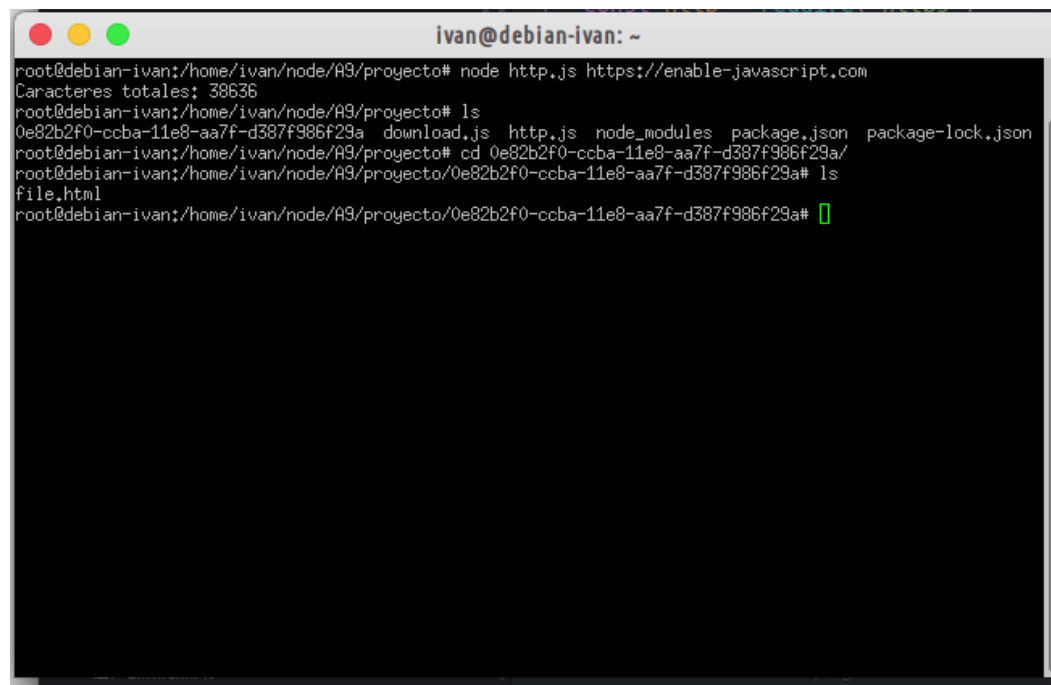
```
ivan@debian-ivan: ~  
root@debian-ivan:/home/ivan/node/A9/proyecto# node http.js https://enable-javascript.com  
Caracteres totales: 38636  
root@debian-ivan:/home/ivan/node/A9/proyecto#
```

Mostramos los archivos que existen en el proyecto y vemos como se ha creado la carpeta aleatoria.



```
ivan@debian-ivan: ~  
root@debian-ivan:/home/ivan/node/A9/proyecto# node http.js https://enable-javascript.com  
Caracteres totales: 38636  
root@debian-ivan:/home/ivan/node/A9/proyecto# ls  
0e82b2f0-ccba-11e8-aa7f-d387f986f29a  download.js  http.js  node_modules  package.json  package-lock.json  
root@debian-ivan:/home/ivan/node/A9/proyecto#
```

Accedemos a la carpeta y vemos el archivo “file.html”



```
ivan@debian-ivan: ~  
root@debian-ivan:/home/ivan/node/A9/proyecto# node http.js https://enable-javascript.com  
Caracteres totales: 38636  
root@debian-ivan:/home/ivan/node/A9/proyecto# ls  
0e82b2f0-ccba-11e8-aa7f-d387f986f29a  download.js  http.js  node_modules  package.json  package-lock.json  
root@debian-ivan:/home/ivan/node/A9/proyecto# cd 0e82b2f0-ccba-11e8-aa7f-d387f986f29a/  
root@debian-ivan:/home/ivan/node/A9/proyecto/0e82b2f0-ccba-11e8-aa7f-d387f986f29a# ls  
file.html  
root@debian-ivan:/home/ivan/node/A9/proyecto/0e82b2f0-ccba-11e8-aa7f-d387f986f29a#
```

Mostramos la información del archivo

```
Actividades PuTTY SSH Client mié o
ivan@

Share</h4>
<a id="ShareTwitter" class="btn-soc btn-twitter" rel="nofollow" href="https://twitter.com/intent/tweet?text=How to enable JavaScript in your browser and why&url=http://enable-javascript.com/&hashtags=JavaScript" target="_blank">Twitter</a>
<a id="ShareFacebook" class="btn-soc btn-facebook" rel="nofollow" href="https://facebook.com/sharer.php?u=enable-javascript.com/" target="_blank">Facebook</a>
<a id="ShareGooglePlus" class="btn-soc btn-googleplus" rel="nofollow" href="https://plus.google.com/share?url=enable-javascript.com/" target="_blank">Google+</a>
<a id="ShareEmail" class="btn-soc btn-email" rel="nofollow" href="mailto:?subject=How to enable JavaScript in your browser and why&body=enable-javascript.com/" target="_blank">Email</a>
>

<p class="font-small">
  How to enable JavaScript in your browser and why<br />http://enable-javascript.com/
</p>
</div>
</div>
</div>
</div>
<div class="footer">
  &copy; <a href="http://toni.podmanicki.com/en/" title="Web design">Toni Podmanicki </a>2009-20
  &#124;
  Contact
  enablejs@gmail.com
</div>
</form>
<script type="text/javascript">
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-6516330-2']);
  _gaq.push(['_trackPageview']);
  (function () {
    var ga = document.createElement('script');
    ga.type = 'text/javascript';
    ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(ga, s);
  })();
</script>
<script type="text/javascript">
  $(document).ready(function () {
    $(".close").click(function () {
      $(".cookies").hide()
    });
  });
</script>
</body>
</html>
root@debian-ivan:/home/ivan/node/A9/proyecto/0e82b2f0-ccba-11e8-aa7f-d387f986f29a#
```

Código

```
mié oct 10, 19:42:15
http.js - ~/.atom/packages/remote-ftp - Atom

p
ftpconfig ejercicio1.js ejercicio2.js http.js
1 var fs = require('fs')
2 const uuidv1 = require('uuid/v1')
3 const folderName = uuidv1();
4 const http = require('https')
5 const url = process.argv[2]
6 var x = 0;
7
8 http.get(url, (response) =>{
9   let rawData = ''
10   response.on('data', (chunk) =>{
11     rawData += chunk
12     x++;
13   })
14
15   response.on('end', ()=>{
16
17     process.stdout.write("Caracteres totales: " + rawData.length + "\n")
18     //process.stdout.write(rawData)
19     fs.mkdirSync(folderName)
20     fs.writeFileSync(folderName + '/file.html', rawData)
21     //console.log('Total de eventos ' + x)
22   })
23   //fin del http.get
24 }).on('error', (error) =>{
25
26   console.error(`Ha ocurrido un error: ${error.message}`)
27
28 })
29
LF UTF-8 JavaScript 0 files
Estilo predeterminado 0,68 / 0,64 15,64 x 9
```