UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Centro de e-Learning
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Curso:

Professional Webmaster

UTN.BA | Centro de e-Learning
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 4:

Aplicaciones web con Node.js

Unidad 4:

Sesiones y middleware

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

En esta unidad vemos qué son y para qué sirven las variables de sesión, que nos van a permitir aprender el concepto de middleware.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Objetivos

Que los participantes logren...

- Comprender los conceptos y usos más comunes de las variables de sesión.

- Utilizar funciones de middleware para acceder a las peticiones.

- Integrar variables de sesión y middleware para lograr funcionalidades avanzadas..

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Sesiones.

2. Middleware.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

1. Sesiones

Las sesiones son un método para hacer que algunas variables estén disponibles en múltiples controladores sin tener que pasarlas como parámetro. A diferencia de las cookies, las variables de sesión se almacenan en el servidor y tienen un tiempo limitado de existencia.

Para identificar al usuario que generó las variables de sesión, el servidor genera una clave única que es enviada al navegador y almacenada en una cookie. Luego, cada vez que el navegador solicita otra página al mismo sitio, envía esta cookie (clave única) con la cual el servidor identifica de qué navegador proviene la petición y puede rescatar de un archivo de texto las variables de sesión que se han creado. Después de un tiempo (configurable) sin peticiones por parte de un cliente (navegador) las variables de sesión son eliminadas automáticamente.

Una variable de sesión es más segura que una cookie ya que se almacena en el servidor. Otra ventaja es que no tiene que estar enviándose continuamente como sucede con las cookies.

Uno de los usos más comunes de las variables de sesión en las aplicaciones web es el proceso de autenticación y autorización de las distintas peticiones que el usuario realiza.

En primer lugar, debemos de conocer brevemente los términos de autenticación y autorización.

Autenticación es el proceso de verificar si el usuario es el mismo que él declara ser.

Autorización es el proceso de determinar si un usuario tiene los privilegios para acceder a un cierto recurso al que ha solicitado acceder.

Para poder utilizar variables de sesion debemos instalar el modulo express-session en nuestro proyecto e inicializarlo correctamente.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Para instalarlo bastará con escribir npm i express-session en nuestra consola. Para configurarlo primero debemos importar el módulo e inicializarlo.

A continuación vemos un pequeño fragmento de código Node.js en el que se ilustra de una manera muy simple el proceso de autenticación y autorización mediante sesiones de express.js. Como era de esperar, hay un punto de inicio de sesión, un punto de cierre de sesión. Para ver la página que está posteada debemos autenticarnos previamente, de esta forma nuestra identidad será verificada y guardada durante la sesión. Cuando cerremos la sesión lo que se producirá internamente es un borrado de nuestra identidad en dicha sesión.

Para poner en funcionamiento debemos instalar el módulo de express-session via npm.

```
npm i express-session
```

Vamos al archivo de configuración (**app.js**)

```
const session = require('express-session');

app.use(session({
    secret: 'inserte clave aqui',
    resave: false,
    saveUninitialized: true
}));
```

En este caso inicializamos el middleware de sesiones pasando como parámetro un objeto con las propiedades secret, resave y saveUninitialized. El único parámetro requerido de estos es secret, que será la cadena de texto o clave que se utilizará para generar el id de sesión que se envía en la cookie al usuario y luego verificar que el mismo no haya sido adulterado por el usuario. Este valor no debe ser sencillo de adivinar, por lo que se recomienda usar cadenas de texto aleatorias de por lo menos 20 caracteres.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

A partir de este momento el objeto res que reciben todos nuestros controladores contará con una propiedad llamada session la cual podemos tanto leer como escribir.

En el siguiente ejemplo hacemos una verificación sencilla de la existencia de la variable de sesion nombre. En caso de existir saludamos al usuario usando este valor. En caso contrario consideramos que el usuario es desconocido.

```
app.get('/ejemplo', function(req, res) {
    if (req.session.nombre) {
        res.send('Hola ' + req.session.nombre) ;
    } else {
        res.send('Hola usuario desconocido.') ;
    }
});
```

A continuación vemos un ejemplo más completo donde verificamos también la existencia de una variable de sesión llamada nombre y en base a su existencia mostraremos o no al usuario un formulario que le permita ingresar su nombre y que el mismo quede grabado en la sesión.

Así mismo le daremos al usuario un link que llevará al usuario a una ruta que se encarga de destruir la sesión, permitiendo iniciar el ciclo nuevamente.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
app.get('/', function(req, res) {
  var conocido = Boolean(req.session.nombre);

  res.render('index', {
    title: 'Sesiones en Express.js',
    conocido: conocido,
    nombre: req.session.nombre
  });

});


app.post('/ingresar', function(req, res) {
  if (req.body.nombre) {
    req.session.nombre = req.body.nombre
  }
  res.redirect('/');
});

app.get('/salir', function (req, res) {
  req.session.destroy();
  res.redirect('/');
});
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
<h1>{{title}}</h1>
{{#if conocido}}
    <p>Hola {{nombre}} <a href="/salir">Salir</a></p>
{{/if}}

{{#unless conocido}}
    <p>No te conozco</p>
    <form action="/ingresar" method="post">
        <input type="text" name="nombre"><button>ingresar</button>
    </form>
{{/unless}}
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

2. Middleware

Un middleware es una función que se puede ejecutar antes o después del manejo de una ruta. Esta función tiene acceso al objeto Request, Response y la función next().

Las funciones middleware suelen ser utilizadas como mecanismo para verificar niveles de acceso antes de entrar en una ruta, manejo de errores, validación de datos, etc.

En resumen las funciones de middleware pueden realizar las siguientes tareas:

- Realizar cambios en la solicitud y los objetos de respuesta.
- Finalizar el ciclo de solicitud/respuestas.
- Invocar el siguiente middleware en la pila.

Si la función de middleware actual no finaliza el ciclo de solicitud/respuestas, debe invocar next() para pasar el control a la siguiente función de middleware. De lo contrario, la solicitud quedará colgada.

El siguiente ejemplo creamos un middleware sencillo para contar las vistas del usuario a una ruta determinada y almacenaremos dicho valor en una variable de

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
app.use(function(req, res, next) {
  // si no existe la variable de sesion vistas la
  // creamos como un objeto vacio
  if (!req.session.vistas) {
    req.session.vistas = {};
  }

  /**
   * buscamos una clave dentro session.vistas que
   * coincida con la url actual. Si no existe, la
   * inicializamos en 1. Si existe sumamos 1 al contador
   * de esa ruta
   */
  if (!req.session.vistas[req.originalUrl]) {
    req.session.vistas[req.originalUrl] = 1;
  } else {
    req.session.vistas[req.originalUrl]++;
  }

  next();
});


app.get('/pagina1', function(req, res) {
  /**
   * pasamos al template la variable vistas con el
   * numero devuelto por la clave que pedimos
   */
  res.render('pagina', {
    nombre: 'pagina1',
    vistas: req.session.vistas[req.originalUrl]
  });
});
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bibliografía utilizada y sugerida

Artículos de revista en formato electrónico:

Express. Disponible desde la URL: https://expressjs.com/es/

Handlebars. Disponible desde la URL:https://handlebarsjs.com/