

ExpressJS - Sessions

 tutorialspoint.com/expressjs/expressjs_sessions.htm

HTTP is stateless; in order to associate a request to any other request, you need a way to store user data between HTTP requests. Cookies and URL parameters are both suitable ways to transport data between the client and the server. But they are both readable and on the client side. Sessions solve exactly this problem. You assign the client an ID and it makes all further requests using that ID. Information associated with the client is stored on the server linked to this ID.

We will need the *Express-session*, so install it using the following code.

```
npm install --save express-session
```

We will put the **session** and **cookie-parser** middleware in place. In this example, we will use the default store for storing sessions, i.e., MemoryStore. Never use this in production environments. The session middleware handles all things for us, i.e., creating the session, setting the session cookie and creating the session object in **req** object.

Whenever we make a request from the same client again, we will have their session information stored with us (given that the server was not restarted). We can add more properties to the session object. In the following example, we will create a view counter for a client.

```
var express = require('express');
var cookieParser = require('cookie-parser');
var session = require('express-session');

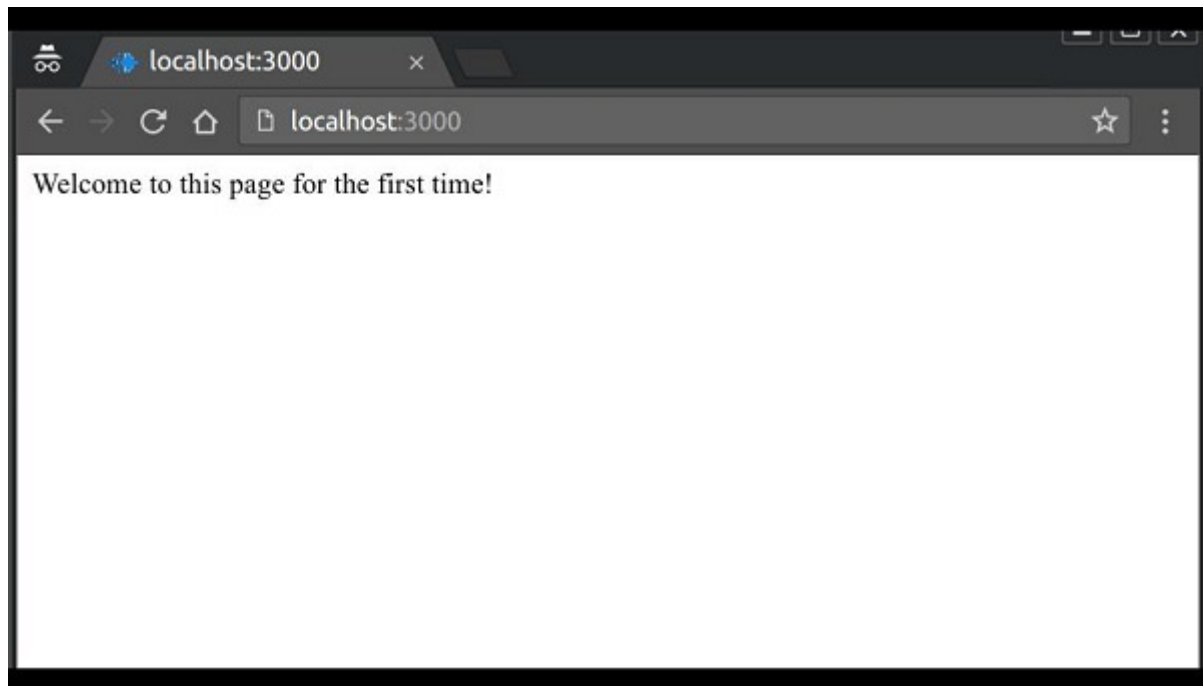
var app = express();

app.use(cookieParser());
app.use(session({secret: "Shh, its a secret!"}));

app.get('/', function(req, res){
  if(req.session.page_views){
    req.session.page_views++;
    res.send("You visited this page " + req.session.page_views + " times");
  } else {
    req.session.page_views = 1;
    res.send("Welcome to this page for the first time!");
  }
});
app.listen(3000);
```

What the above code does is, when a user visits the site, it creates a new session for the user and assigns them a cookie. Next time the user comes, the cookie is checked and the **page_view** session variable is updated accordingly.

Now if you run the app and go to **localhost:3000**, the following output will be displayed.



If you revisit the page, the page counter will increase. The page in the following screenshot was refreshed 42 times.

