LAB 15

STATE MANAGEMENT

What You Will Learn

- How to set and read cookies in PHP, Node, and JavaScript
- How to work with server-side session data in PHP and Node

Approximate Time

The walkthroughs in this lab should take approximately 20 minutes to complete.

Each project at the end of this lab should take between 30 and 60 minutes to complete.

Fundamentals of Web Development, 3rd Ed

Randy Connolly and Ricardo Hoar

Textbook by Pearson
http://www.funwebdev.com

Date Last Revised: March 21, 2023

Preparing Directories

The starting lab15 folder has been provided for you (within the zip folder downloaded from Gumroad). This lab requires PHP, Node, and client-side JavaScript coding. If you are only interested in, say, the PHP exercises, then you can skip the exercises involving Node (or vice-versa).

1 For the PHP exercises, this lab (like Lab 12) requires a functioning webserver. This lab assumes you are using XAMPP.

If you wish to do both the PHP and the Node exercises in this lab, you can do all your work within the htdocs folder provided by XAMPP.

Create a folder named lab15 in your PHP development location on your development machine. Copy the contents of the provided zip into the lab15 folder.

2 If you wish to only do the Node exercises, then create a folder named lab15 in your Node development location on your development machine. Copy the contents of the provided node subfolder into the lab15 folder.

3 For the Node exercises, you will need to prep the folder by issuing the following commands.

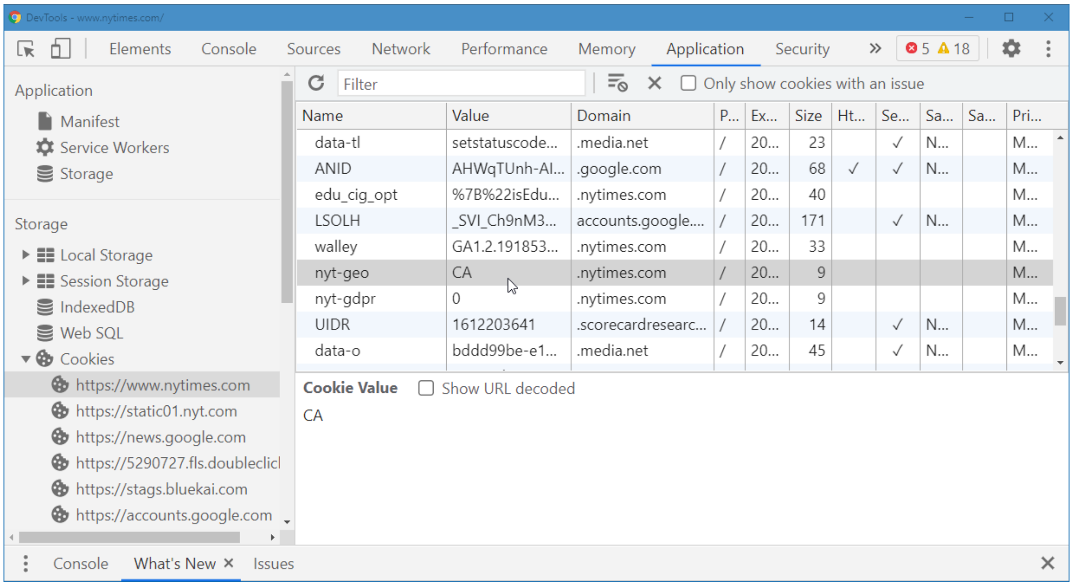
```
npm init
npm install express cookie-parser
```

Cookies

Exercise 15.1 — Examining Cookies in the Browser

1 In the Chrome browser, visit nytimes.com. Right-click and choose the Inspect option, then click on the Application tab. On the left-hand side of the window, open the Cookies option and examine. The result will look similar to that shown in Figure 15.1.

2 In the Firefox browser, visit nytimes.com. Right-click and choose the Inspect option, then click on the Storage tab. On the left-hand side of the window, open the Cookies option and examine.



Name	Value	Domain	P...	Ex...	Size	Ht...	Se...	Sa...	Sa...	Pri...
data-tl	setstatuscode...	.media.net	/	20...	23		✓	N...		M...
ANID	AHWqTUnh-Al...	.google.com	/	20...	68	✓	✓	N...		M...
edu_cig_opt	%7B%22isEdu...	.nytimes.com	/	20...	40					M...
LSOLH	_SVI_Ch9nM3...	accounts.google....	/	20...	171		✓	N...		M...
walley	GA1.2.191853...	.nytimes.com	/	20...	33					M...
nyt-geo	CA	.nytimes.com	/	20...	9					M...
nyt-gdpr	0	.nytimes.com	/	20...	9					M...
UIDR	1612203641	.scorecardresearc...	/	20...	14		✓	N...		M...
data-o	bddd99be-e1...	.media.net	/	20...	45		✓	N...		M...

Cookie Value □ Show URL decoded

CA

Figure 15.1 – Examining cookies within Chrome.

Exercise 15.2 — Cookies in PHP

1 Open and examine `lab15-ex02.php`.

Notice the form element's action attribute: the submit will re-request this same file.

2 Add the following to the top of the document.

```
<?php
/* if form has been submitted, save user's input in a cookie */
if ( isset($_POST["tracker"]) ) {
   setCookie("tracker", $_POST["tracker"]);
}
?>
```

3 In the second column, add the following code:

```
<div class="content">
   <?php
   if (isset($_POST["tracker"])) {
      echo "<em>Cookie set.</em><br>";
      echo "<a href='anotherpage.php'>";
      echo "Visit this page to see it</a>";
   } else {
      echo "Cookie not set yet";
   }
   ?>
</div>
```

Copyright © 2023 Randy Connolly

4 Add the following code to the top of `anotherpage.php`.

```
<?php
// add code for accessing cookie on the server side
if ( isset($_COOKIE["tracker"]) ) {
  $msg = "Cookie value is = " . $_COOKIE["tracker"];
} else {
  $msg = "Cookie not set yet";
}
?>
```

This page will read and display the cookie set by lab15-ex02.php.

5 In the markup, add the following.

```
<h2 class="is-size-4">Received from client and displayed on
server-side</h2>
<div class="content"><?= $msg ?></div>
```

6 Save both files and then view `lab15-ex02.php` in the browser.

Remember to view a PHP file it needs to be running on a live server such as XAMPP.

7 Enter some text in the input field and submit the form. Click on the link to the other file when it is displayed. You should see your input value as a cookie value.

8 Inspect your cookie value using the browser's Inspect feature (shown in the previous exercise).

9 Close the browser tab with this file, then create a new tab and re-request `lab15-ex02.php`. It should say that the cookie is not yet set. How can this be?

Because you did not specify an expiry on your cookie, your browser made it a session cookie, which means it only lasted as long as the browser session. When you closed the tab, the session was ended.

10 Edit `lab15-ex02.php` by modifying the code as follows.

```
if ( isset($_POST["tracker"]) ) {
   // add 1 day to the current time for expiry time
   $expiryTime = time()+60*60*24;
   setCookie("tracker", $_POST["tracker"], $expiryTime);
}
```

11 Test by setting a cookie, closing the tab, and then re-requesting `anotherpage.php` in a new tab.

The cookie value now should persist across sessions. You may want to check back in a day to see if the cookie is still valid!

The previous example illustrated how to set and read cookie values in PHP. While cookies will usually be initialized on the server using PHP, Node, ASP.NET, etc., it is not uncommon to manipulate them on the client-side just using JavaScript.

Exercise 15.3 —Cookies in Client-side JavaScript

1 Add the following code to the `<script>` of `anotherpage.php`.

```
<script>
  document.addEventListener("DOMContentLoaded", function() {

    // add code for accessing cookie on client-side,
    // i.e., a cookie sent from the server
    const div = document.querySelector("#cookieHolder");

    if (! document.cookie) {
      div.textContent = "No cookies set yet";
    } else {
      const cValue = document.cookie
        .split('; ')
        .find(row => row.startsWith('tracker='))
        .split('=')[1];
      div.textContent = "Value of cookie is = " + cValue;
    }

  });
</script>
```

2 Test by requesting `lab15-ex02.php`, setting a cookie, and then clicking the link to `anotherpage.php`

Exercise 15.4 —Cookies in Node

1 If you haven't already done so, install the necessary express cookie package in your node folder:

```
npm install cookie-parser
```

2 Add the following code to `lab15-ex04.js`.

```
// add the cookie parser middleware here
const cookieParser = require('cookie-parser');
app.use(cookieParser());

// now handle post request from form
app.post('/setCookie', (req,res) => {
      // set the cookie
     res.cookie('tracker',req.body.tracker);
     // then send the html that will display the cookie
     const filename = __dirname + "/public/anotherpage.html";
     res.sendFile(filename);
});
```

3 Modify `cookie-setter.html` in the `public` subfolder as follows:

```
<form action="/setCookie" method="post">
```

Notice that this is the same route you defined in the previous step.

Copyright © 2023 Randy Connolly

4 Examine `anotherpage.html` in the `public` subfolder.

Notice that this contains the same cookie-reading JavaScript you added in Exercise 15.3

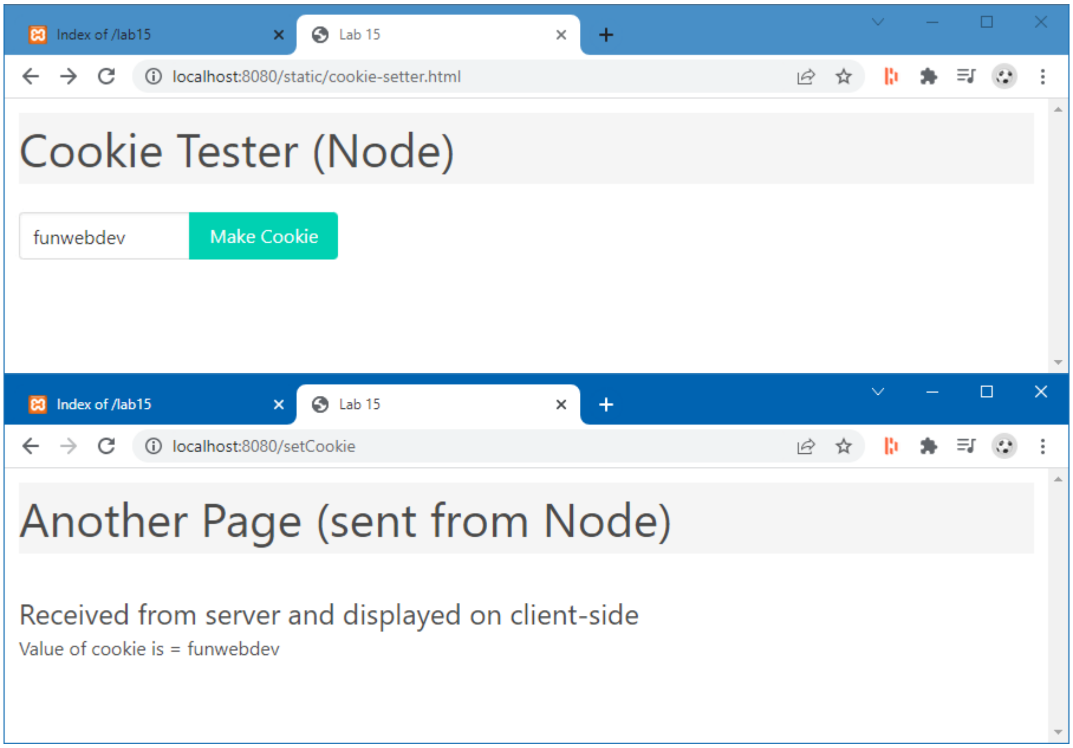
5 Test by using the following command:

```
node lab15-ex04
```

6 Make the following request in your browser:

```
http://localhost:8080/static/cookie-setter.html
```

When you click the Make Cookie button, it should display the cookie value you entered in the form (see Figure 15.2).



Cookie Tester (Node)

funwebdev

Another Page (sent from Node)

Received from server and displayed on client-side

Value of cookie is = funwebdev

Figure 15.2 – Setting cookies in Node

Exercise 15.5 —Sessions in PHP

1 Open and examine `lab15-ex04.php`.

This exercise will contain four files. This one provides a list of art items that can be added to the favorites list. Notice that this page displays items from an included data array.

2 Open and examine `includes/paintings.inc.php`.

Instead of pulling data from a database, this simplified exercise uses this small data array.

3 Modify the link in `lab15-ex04.php` as follows:

```
<a class="no-underline text-yellow-700 …"
   href="addToFavorites.php?id=<?= $key ?>" >
   Add to Favorites
</a>
```

You will be passing the key identifier for the painting to the page that will be adding it to the session-based favorites list.

4 Test to ensure the data is being displayed correctly and the links have the correct URL.

5 Edit `addToFavorites.php` and add the following code to it:

```
<?php
// ensure sessions works on this page
session_start();

// does session already exist?
if ( !isset($_SESSION["Favorites"]) ) {
   // initialize an empty array that will contain the favorites
   $_SESSION["Favorites"] = [];
}

// retrieve favorites array for this user session
$favorites = $_SESSION["Favorites"];

// now add passed painting id to our favorites array
$favorites[] = $_GET["id"];

// then resave modified array to session state
$_SESSION["Favorites"] = $favorites;

// finally redirect to favorites page
header("Location: favorites.php");

?>
```

6 Open and examine `favorites.php`.

This page will display all the paintings that have been added to the favorites list.

Copyright © 2023 Randy Connolly

7 At the top of this page, add the following:

```
<?php
include 'includes/paintings.inc.php';

// ensure sessions works on this page
session_start();
// if no favorites in session, initialize it to empty array
if ( !isset($_SESSION["Favorites"]) ) {
   $_SESSION["Favorites"] = [];
}

// retrieve favorites array for this user session
$favorites = $_SESSION["Favorites"];

?>
```

7 Loop through the favorites list and display each painting in it, by adding the following:

```
<ul class="flex flex-col divide divide-y">
<?php
foreach ($favorites as $fav) {
   // retrieve the painting for this id
   $painting = $data[$fav];
?>
   <li class="flex flex-row">
     <div class="select-none cursor-pointer flex flex-1 items-
center p-4">
         <div class="flex flex-col w-10 h-10 justify-center items-
center mr-4">
           <a href="#" class="block relative">
           <img src="images/<?= $painting[0] ?>.jpg"
             class="mx-auto object-cover rounded-full h-10 w-10 "/>
            </a>
         </div>
         <div class="flex-1 pl-1 mr-16">
            <div class="font-medium dark:text-white">
                <?= $painting[1] ?>
            </div>
         </div>
     </div>
   </li>
<?php
}
?>
</ul>
```

8 Add a link for emptying the favorites list by modifying this link's `href`:

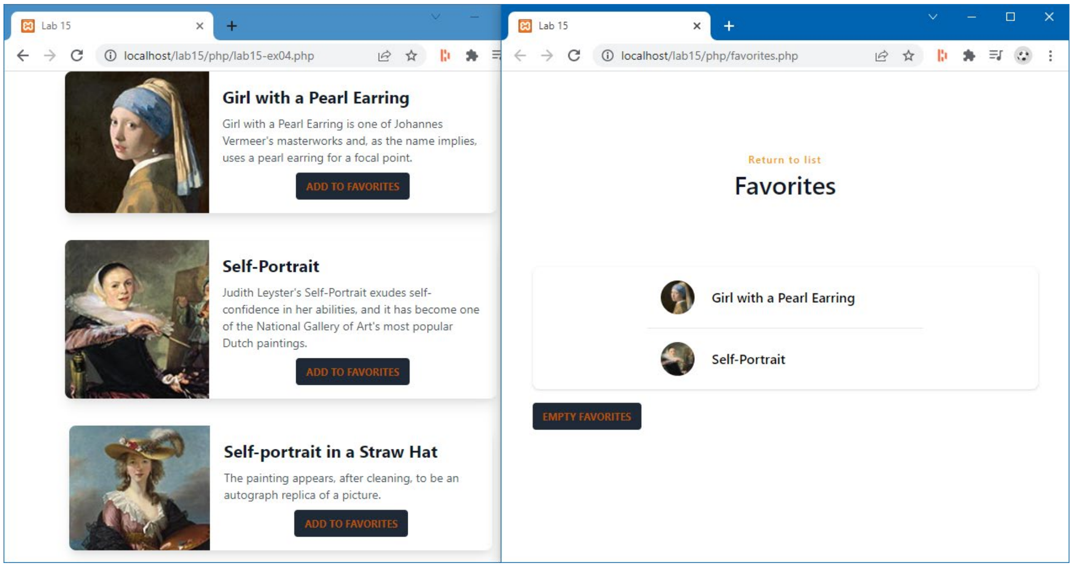
```
<a class="no-underline text-yellow-700 …"
      href="emptyFavorites.php">
         Empty Favorites
</a>
```

9 Create a new file named `emptyFavorites.php` and add the following code to it:

```
<?php
// ensure sessions works on this page
session_start();

// resave empty array to session state
$_SESSION["Favorites"] = [];

// redirect to favorites page
header("Location: favorites.php");

?>
```

10 Test. Begin by viewing `lab15-ex04.php` in the browser. Click on a painting's Add To Favorites link. Return to the list and add another painting. The favorites list should accurately display the paintings added to the list. Try the Empty Favorites link as well.

Note: this code doesn't check for duplicates. If you did want to prevent duplicate entries, which file would you change?



Girl with a Pearl Earring

Girl with a Pearl Earring is one of Johannes Vermeer's masterworks and, as the name implies, uses a pearl earring for a focal point.

Self-Portrait

Judith Leyster's Self-Portrait exudes self-confidence in her abilities, and it has become one of the National Gallery of Art's most popular Dutch paintings.

Self-portrait in a Straw Hat

The painting appears, after cleaning, to be an autograph replica of a picture.

Favorites

Girl with a Pearl Earring

Self-Portrait

Figure 15.3 – Using sessions in PHP

Copyright © 2023 Randy Connolly

Exercise 15.6 —Sessions in Node and Express

1 Begin by installing the necessary express cookie package in your node folder:

```
npm install express-session
```

2 Add the following code to `lab15-ex06.js`.

```
// add the session middleware here
const session = require('express-session');
app.use(session({
   secret: 'could be anything',
   resave: false,
   saveUninitialized: true
   // in a production https environment, you would also add
   // cookie: { secure: true, httpOnly: true }
}));

// now handle post request from form
app.post('/addToFavorites', (req,res) => {
   // check if we have a favorites array for this session
   if (req.session.favorites) {
      const favorites = req.session.favorites;
      favorites.push({id: req.body.id, title: req.body.title});
      req.session.favorites = favorites;
   } else {
      req.session.favorites = [
        {id: req.body.id, title: req.body.title} ];
   }
   res.json({
      message: "POST request was successful",
      count: req.session.favorites.length,
      received: {id: req.body.id, title: req.body.title}})
});
```

3 Examine `listing.js` in the `public` subfolder.

Notice that when the user clicks a Add to Favorites button, it posts the painting id and title as a json object using fetch.

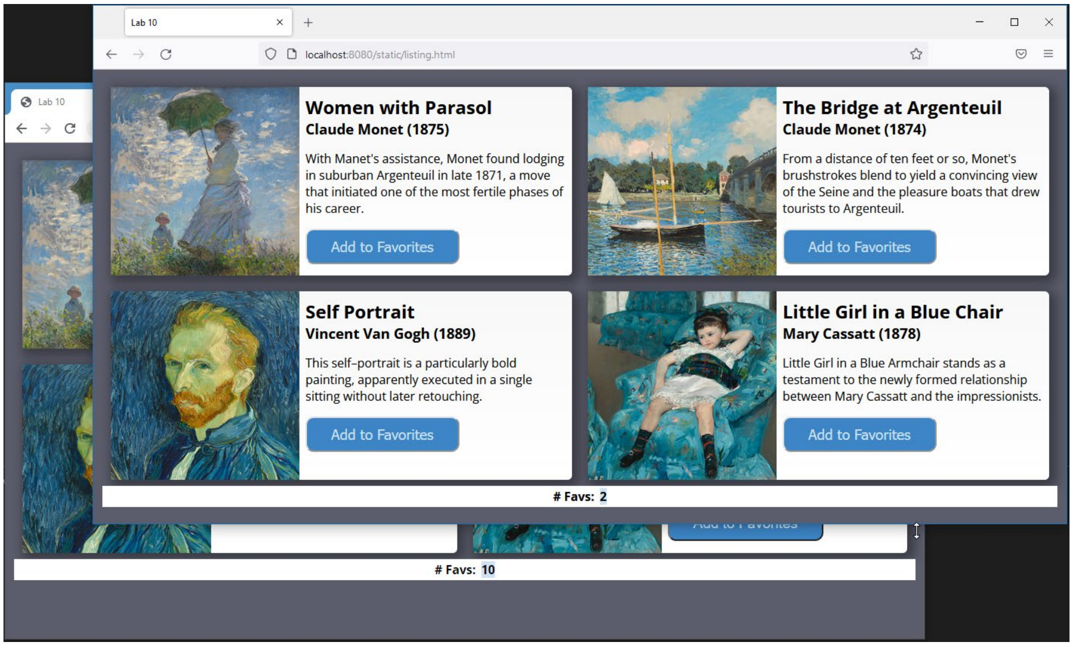
4 Test by using the following command:

```
node lab15-ex06
```

5 Make the following request in your browser:

`http://localhost:8080/static/listing.html`

When you click the Add to Favorites button, you should see a snack bar message when it receives a response from server; it should also update the # favorites count. Try to request the same page in different browsers to verify each browser session has its own session (see Figure 15.4).



Women with Parasol
Claude Monet (1875)

With Manet's assistance, Monet found lodging in suburban Argenteuil in late 1871, a move that initiated one of the most fertile phases of his career.

The Bridge at Argenteuil
Claude Monet (1874)

From a distance of ten feet or so, Monet's brushstrokes blend to yield a convincing view of the Seine and the pleasure boats that drew tourists to Argenteuil.

Self Portrait
Vincent Van Gogh (1889)

This self-portrait is a particularly bold painting, apparently executed in a single sitting without later retouching.

Little Girl in a Blue Chair
Mary Cassatt (1878)

Little Girl in a Blue Armchair stands as a testament to the newly formed relationship between Mary Cassatt and the impressionists.

Figure 15.4 – Each browser session has its own session

6 Add the following code to `lab15-ex06.js`.

```
// add view handler here
app.get("/listFavorites", (req, res) => {
  if (req.session.favorites) {
    const favorites = req.session.favorites;
    res.json(favorites);
  } else {
    res.json([]);
  }
});
```

7 Add the following markup to `listing.html`.

```
<div>
   # Favs: <span id="count">0</span>
   <a href="/listFavorites">View Favorites as JSON</a>
</div>
```

8 Test by adding a few favorites and then clicking the View Favorites link.

Copyright © 2023 Randy Connolly

Exercise 15.7 — Using memcache in PHP

1 In the future …

Exercise 15.8 — Caching Packages in Node

1 In the future …

Exercise 15.9 — Server-Side Rendering for React

1 In the future …