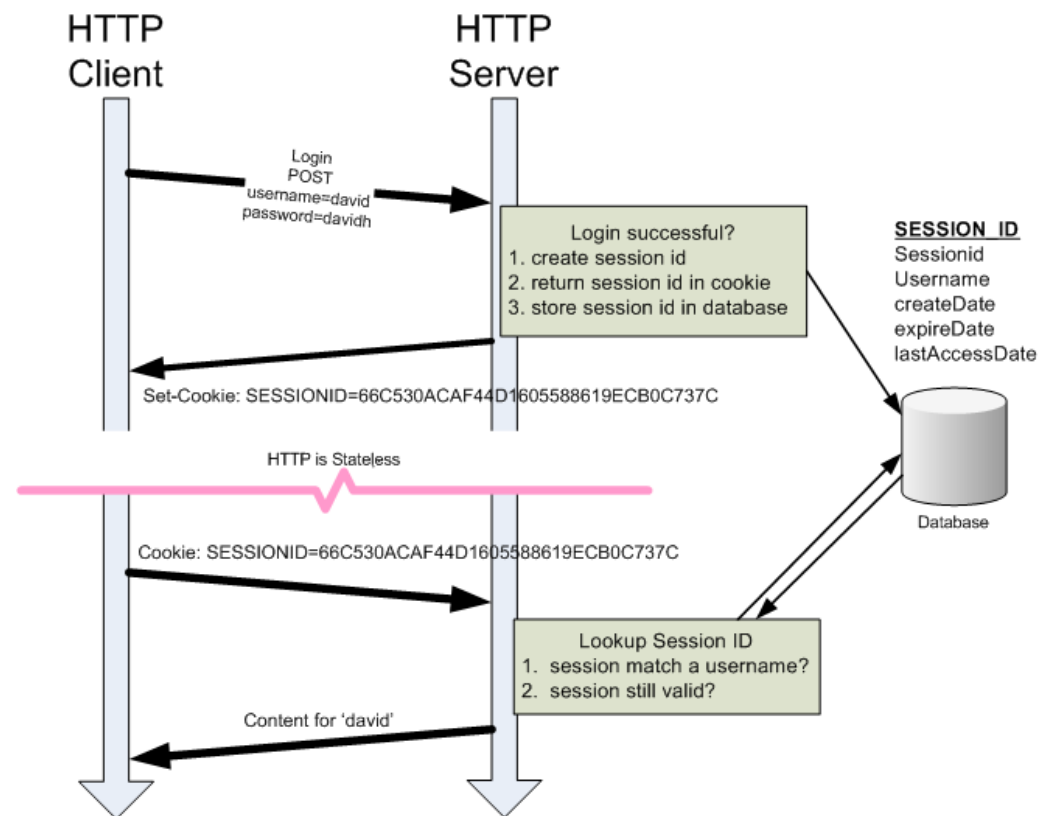


Session

- ▶ Provides a way to identify a user across more than one page request or visit to the web site and to store information about that user
- ▶ Session allows the application to store state
 - ▶ Based on what action a user took on Page A, we can show a different Page B

How Does Session Work?

- ▶ A **session ID or token** is a unique number which is used to identify a user that has logged into a website



express-session

- ▶ The package express-session can be used as a simple session middleware for Express

```
C:\NodeJS\session-demo>npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN session-demo@1.0.0 No description
npm WARN session-demo@1.0.0 No repository field.

+ express@4.16.3
added 50 packages in 3.236s

C:\NodeJS\session-demo>npm install cookie-parser
npm WARN session-demo@1.0.0 No description
npm WARN session-demo@1.0.0 No repository field.

+ cookie-parser@1.4.3
added 1 package in 1.045s

C:\NodeJS\session-demo>npm install express-session
npm WARN session-demo@1.0.0 No description
npm WARN session-demo@1.0.0 No repository field.

+ express-session@1.15.6
added 5 packages in 1.755s
```

Setting a Session

- ▶ Here's how you can set up a simple session in Express:

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

const app = express();

app.use(cookieParser());
app.use(session({ secret: 'keyboard cat' }));
```

- ▶ The secret is used to sign the session ID cookie
- ▶ To store or access session data, simply use the request property `req.session`

Session Storage

- ▶ The default server-side session storage is **MemoryStore**
- ▶ MemoryStore uses the application RAM for storing session data
- ▶ It is not recommended for use in a production environment because:
 - ▶ Memory consumption will keep growing with each new session
 - ▶ In case the app is restarted for any reason all session data will be lost
 - ▶ Session data cannot be shared by other instances of the app in a cluster
- ▶ Other ways to store session data:
 - ▶ In a cookie (use the session-cookie package)
 - ▶ In a memory cache server (use connect-memcached package)
 - ▶ In a database (e.g., for working with MongoDB use connect-mongo)

Session Usage Example

- ▶ To store or access session data, simply use the request property **req.session**

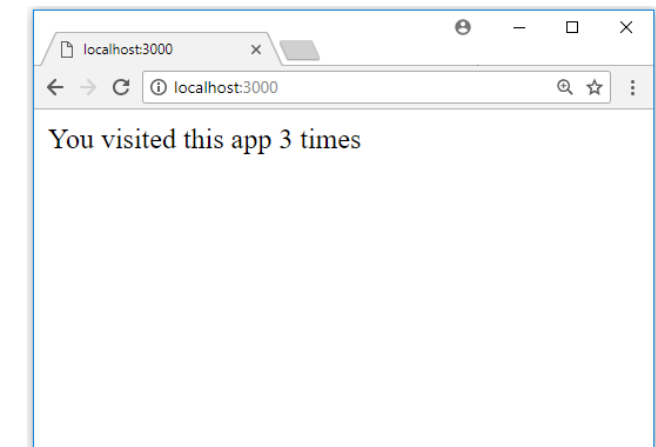
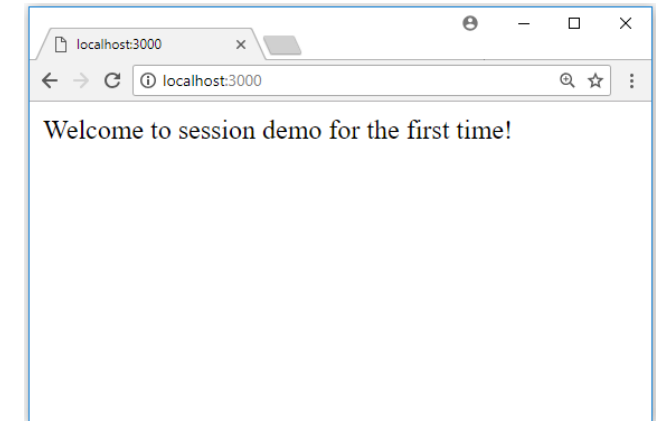
```
const express = require('express');
const cookieParser = require('cookie-parser');
const session = require('express-session');

const app = express();

app.use(cookieParser());
app.use(session({ secret: 'keyboard cat' }));

app.get('/', (req, res) => {
  if (req.session.pageViews) {
    req.session.pageViews++;
    res.send('You visited this app ' + req.session.pageViews + ' times');
  } else {
    req.session.pageViews = 1;
    res.send('Welcome to session demo for the first time!');
  }
});

app.listen(3000);
```



Session Timeout

- ▶ **cookie.maxAge** specifies the expiration time of the session relative to the current server time (in milliseconds)
- ▶ By default, no maximum age is set

```
app.use(session({ secret: 'keyboard cat', cookie: { maxAge: 600000 } })))  
// session expires after 10 minute
```

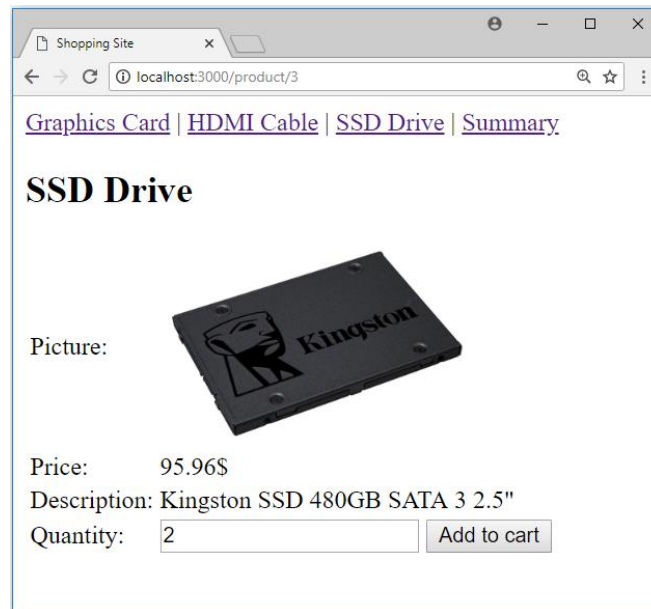
Deleting a Session

- ▶ The method **req.session.destroy()** is used to logout from the current session
- ▶ It destroys the session, deletes the session cookie, and unsets req.session
- ▶ The destroy() method accepts an optional callback function to be executed after the session is cleared from the store

```
req.session.destroy(function() {  
    res.send('Session deleted');  
});
```


Exercise (10)

- ▶ Create a shopping site that will let the user choose one of 3 products
- ▶ Save the products the user has picked and their amounts in a shopping cart
 - ▶ Note: the user can choose the same product more than once
- ▶ The summary page should display the shopping cart and its total price



Name	Price	Quantity	Total
Graphics Card	299\$	2	598\$
HDMI Cable	8.99\$	1	8.99\$
SSD Drive	95.96\$	2	191.92\$

Total: 798.91\$