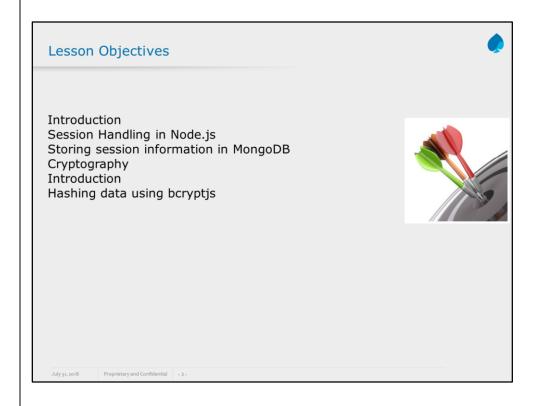
Instructor Notes:



Instructor Notes:



Instructor Notes:

Add instructor notes here.

4.1: Session Handling Introduction



>HTTP is a stateless protocol, server cannot differentiate between different connections of different users.

➤One of the most common requirements for any web application are sessions.

>Using sessions to keep track of users as they browse through the site, becomes an important aspect for web application.

>Typically, the process of managing the state of a web-based client is through the use of session IDs. Session IDs are used by the application to uniquely identify a client browser, while background (server-side) processes are used to associate the session ID with a level of access.

There are three methods available to allocate and receive session ID information.

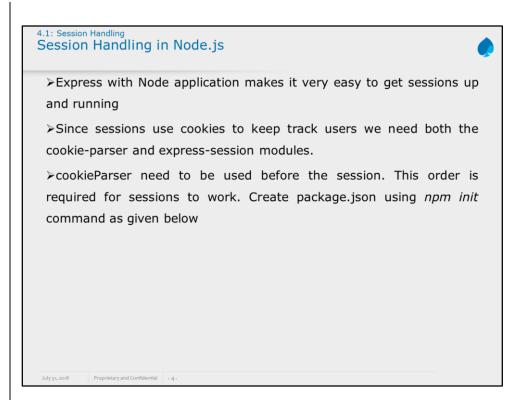
- URL Based Session ID's
- · Hidden Post Fields
- Cookies

July 31, 2018 Proprietary and Confidential - 3 -

Session cookies are temporary cookie files, which are erased when browser is closed. When browser restarted a new session cookie will be generated, which will store browsing information and will be active until the browser is closed.

Each time a client web browser accesses content from a particular domain or URL, if a cookie exists, the client browser is expected to submit any relevant cookie information as part of the HTTP request. Thus cookies can be used to preserve knowledge of the client browser across many pages of the site.

Instructor Notes:

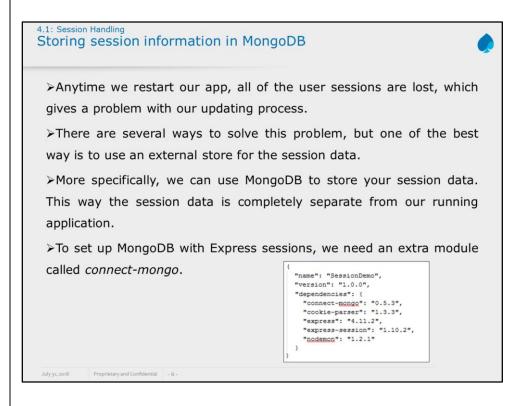


Instructor Notes:

```
4.1: Session Handling
Session Handling in Node.js
     var express = require('express');
     var cookieParser = require('cookie-parser');
     var session = require('express-session');
     var app = express();
     app.use(cookieParser());
     app.use(session({secret:'IGATEKarthik', resave: true, saveUninitialized: true}));
     app.get('/', function(req, res) {
        console.log('home : '+req.sessionID);
         if(req.session.lastPage) {
           res.write('Last page was: ' + req.session.lastPage + '.\n');
        req.session.lastPage = 'Home';
        res.end('Home Page');
     app.get('/about', function(req, res) {
        console.log('about : '+req.sessionID);
        if(req.session.lastPage) {
            res.write('Last page was: ' + req.session.lastPage + '.\n');
        req.session.lastPage = 'AboutUs';
        res.end('About us');
     });
     app.listen(process.env.PORT || 3000);
     console.log('Server listening on the port 3000');
          Proprietary and Confidential - 5 -
```

```
/*Deny Directory Listing*/
app.use('/', express.static(__dirname + '/'));
app.get('*', function(request, response, next) {
  response.sendfile(__dirname + '/index.html');
});
```

Instructor Notes:



Instructor Notes:

Add instructor notes here.

```
4.1: Session Handling
Storing session information in MongoDB
        var express = require('express');
        var cookieParser = require('cookie-parser');
       var session = require('express-session');
       var MongoStore = require('connect-mongo') (session);
       var app = express();
        app.use(cookieParser());
        app.use(session({
           store : new MongoStore({"db": "SessionDB", "host": "localhost", "port": "27017", "collection": "sessions"}),
            cookie: { maxAge: 50 * 1000 }, /* 5 min */
           secret: 'IGATEKarthik'.
           saveUninitialized: true
       }));
       app.get('/', function(req, res) {
   console.log('home : '+req.sessionID);
            if(req.session.lastPage) {
               res.write('Last page was: ' + req.session.lastPage + '.\n');
            req.session.lastPage = 'Home';
           res.end('Home Page');
       app.get('/logout', function(req, res) {
    req.session.destroy();
            res.redirect("/");
        app.listen(process.env.PORT || 3000);
        console.log('Server listening on the port 3000');
              Proprietary and Confidential - 7 -
```

Failure to set "saveUninitialized" and "resave" will generate two warnings:

This is simply saying the default values will change so they want to ensure that by setting the values explicitly now. By doing this you won't run into unexpected behavior if the defaults do change in the near future.

Instructor Notes:



Instructor Notes:

Add instructor notes here.

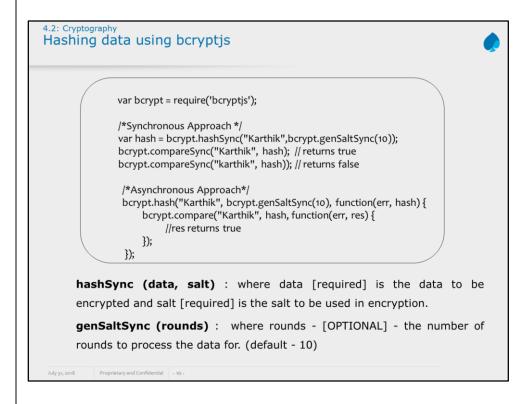
4.2: Cryptography Introduction



- >Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.
- >As a best practice, any private information (like password) stored on the server should be encrypted.
- ➤ Hashing is the greatest way for protecting passwords and considered to be pretty safe for ensuring the integrity of data or password.
- ➤Bcrypt hashing algorithm is a good option for secured password hashing. It is a one way hash i.e. It cannot be decrypted.
- >bcryptjs is the popular node module for hashing private information.

July 31, 2018 Proprietary and Confidential - 9 -

Instructor Notes:



Instructor Notes:

