

Bachelor Thesis Presentation



Besenbäck Simon, 11801501



- JavaScript
 - Most used programming language
 - Mainly for front-end development
 - Framework Node.js enables Back-end development
- Alternative Approach:
 - Jakarta Server Pages + Tag Library (BSF4ooRex) + Nashorn

Title of the Thesis

- English: “Developing Jakarta Server Pages as a Front-End Developer - An Alternative Approach to Developing Server-Side with Node.js”
- German: “Entwicklung von Jakarta-Serverseiten als Frontend-Entwickler - Ein alternativer Ansatz zur serverseitigen Entwicklung mit Node.js”

- Theoretical Background Information
- Necessary software components
- Presentation of Developed Applications
 - Basic Nutshell Examples
 - E-Commerce Example
 - Advanced Examples
- Summary

Theoretical Background

What is Web Development?

- Creating and maintaining websites
 - Design
 - Layout
 - Content
 - Functionality
- Subcategories:
 - Front-end development
 - Back-end development
 - Full-stack development

What are Jakarta Server Pages?

- Creating dynamic web pages for Java web applications
- Connect HTML page to the back-end Java Code
- “Normal Web Development”:
 - Static HTML page sent to the client
 - Content made dynamic using JavaScript
- “JSP Web Development”:
 - **HTML page generated dynamically on the server-side**

- Set of custom tags
 - Enables to invoke custom actions in a JSP file
- Simplifies complex actions
 - Provides Java functions without the need to be capable of Java
- Dynamically generate page content

Why develop JSPs with Nashorn?

- What is Nashorn?
 - JavaScript engine
 - Enables using JavaScript for the development of JSPs
- Conclusion:

JavaScript instead of Java + JSP's back-end capabilities

=

Develop front-end and back-end by using JavaScript

JSR-223 taglib - script-jsr223.tld

- Drawback using Nashorn:
 - No support for XMLHttpRequests
- Solution:
 - Script-jsr223.tld developed by Dr Rony G. Flatscher
 - Enables using any Java scripting language for back-end development

Necessary Software

Necessary Software



Apache Tomcat



OpenJDK

Developed Applications – Basic Nutshell Examples

Basic structure of an JSP file

```
<%@ page session="false" pageEncoding="UTF-8" contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="WEB-INF/script-jsp223.tld" prefix="s" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hello, World</title>
</head>
<body>
<s:script type="javascript">
    var greeting = "Hello, world! (Sent from Nashorn)";
    print("<div>" + greeting + "</div>")
</s:script>
</body>
</html>
```

Page
Directives

Script
Accessing the
Tag Library

/helloworld/helloworld.jsp

Setting Cookies

```
12
13 <s:script type="javascript" throwException="true">
14
15 //request
16 var lastVisit;
17 var allCookies = request.getCookies();
18
19 //response to add the current time
20 var today = new Date();
21 var time = today.getHours() + ":" + today.getMinutes() + ":" + today.getSeconds();
22 timeString = String(time);
23 var Cookie = Java.type("jakarta.servlet.http.Cookie");
24 var newCookie = new Cookie("lastVisit", timeString);
25 newCookie.setMaxAge(60 * 60 * 24);
26 newCookie.setPath("/");
27 response.addCookie(newCookie);
28
29 </s:script>
```

/helloworld/lastvisit.jsp

Print Time of
Last Visit

Create and
Send Cookie

```
32 <s:script type="javascript">
33 //Print the last visit or if it is the first visit
34 if (allCookies != null) {
35     for (var i = 0; i < allCookies.length; i++) {
36         var c = allCookies[i];
37         if (c.getName() == 'lastVisit') {
38             lastVisit = c.getValue();
39         }
40     }
41 }
42
43 if (lastVisit == null) {
44     print("This is your first Visit!");
45 } else {
46     print("Your last visit was at " + lastVisit);
47 }
48 </s:script>
```

/helloworld/lastvisit.jsp

Problem with setting Cookies and print()

- `print()` executed before `response.addCookie()`
- Solutions:
 - a) Execute `print()` after response
 - b) Use **`out.print()`** or **`out.println()`**

Requesting User Input

```
28 out.println('<form>' +  
29         '<label for="username">Username:</label>' +  
30         '<input type="text" name="username" required>' +  
31         '<input type="submit" value="Ok">' +  
32         '</form>');
```

/helloworld/greeting.jsp

Form Enabling
User Input

Setting New
Cookie &
Reloading Page

```
39 if (request.getParameter('username') != null) {  
40     var uname = request.getParameter('username');  
41     var Cookie = Java.type("jakarta.servlet.http.Cookie");  
42     var newCookie = new Cookie("username", uname);  
43     newCookie.setMaxAge(60 * 60 * 24);  
44     newCookie.setPath("/");  
45     response.addCookie( newCookie );  
46     response.sendRedirect(request.getRequestURI())  
47 }
```

/helloworld/greeting.jsp

Deleting Cookies

Accessing
External
Script

```
42 <s:script type="javascript" src="code/logout.js" cacheSrc="false" />
```

/helloworld/greeting_ext.jsp

Deleting Cookie
by Replacing it

```
11 if (request.getParameter('logoutButton') != null) {  
12     var Cookie = Java.type('jakarta.servlet.http.Cookie');  
13     var removerCookie = new Cookie('username', '');  
14     removerCookie.setPath('/');  
15     removerCookie.setMaxAge(0);  
16     response.addCookie(removerCookie);  
17     response.sendRedirect(request.getRequestURI());  
18 }
```

/helloworld/code/logout.js

Developed Applications – E-Commerce Example

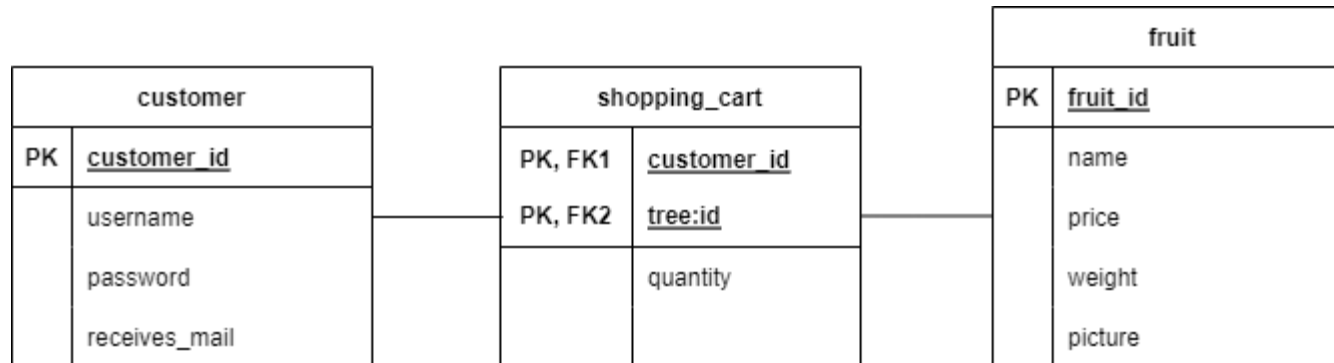
Developing an Online Shop, where...

- offered products are displayed in the main page
- users can sign in and log in
- users can add products to the shopping cart and checkout

- SQLite
 - Open-source library
 - Relational database system
 - No additional server software necessary
 - Storage sparing
- Connection:

```
19 // Import the DriverManager class from the java.sql package
20 var DriverManager = Java.type('java.sql.DriverManager');
21
22 // Create a connection to the SQLite database
23 var conn = DriverManager.getConnection("jdbc:sqlite:C://sqlite/shop.db");
```

Database Structure



Reading Data from the Server

```
18<s:script type="javascript">
19 // Import the DriverManager class from the java.sql package
20 var DriverManager = Java.type('java.sql.DriverManager');
21
22 // Create a connection to the SQLite database
23 var conn = DriverManager.getConnection("jdbc:sqlite:C://sqlite/shop.db");
24
25 // Execute a SELECT query to retrieve all data from table fruit
26 var stmt = conn.createStatement();
27 var rs = stmt.executeQuery('SELECT * FROM fruit;');
28
29 out.println('<ul>');
30 while (rs.next()) { //iterate through all rows in the table
31     out.println('<li>' + rs.getString('name') + ': ' + rs.getString('price') +
32         '€</li>');
33 };
34 out.println('</ul>')
35 </s:script>
```

/fruitshop/productlist.jsp

Main Page

Welcome to our fruitshop!

Sign Up

Login

Productlist

Items in Cart: 0

View Cart

Ananas

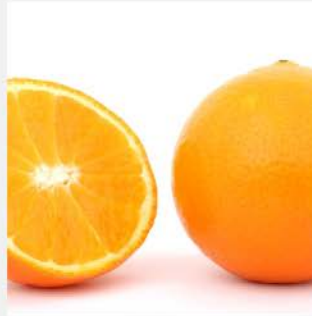


Price: 80 Euro

Weight: 20Kg

1 ▾ Buy

Orange



Price: 50 Euro

Weight: 15Kg

1 ▾ Buy

Cherry



Price: 70 Euro

Weight: 10Kg

1 ▾ Buy

Main Page – Displaying Products

Template for
Displaying the
Products

```
105 /// HTML TEMPLATE ///
```

```
106 function templateProductItem(name, price, weight, picture, fruit_id) {
```

```
107     out.println('<div class="product-container">' +
```

```
108         '<h2>' + name + '</h2>' +
```

```
        '' +
```

```
        '<p>Price: ' + price + ' Euro</p>' +
```

```
        '<p>Weight: ' + weight + ' Kg</p>' +
```

```
        '<form name="selection" method="post">' +
```

```
        '<input type="hidden" name="selection" value="' + fruit_id + '">' +
```

```
114         '<select name="quantity">' +
```

```
115             '<option value="1">1</option>' +
```

```
116             '<option value="2">2</option>' +
```

```
117             '<option value="3">3</option>' +
```

```
118             '<option value="4">4</option>' +
```

```
119             '<option value="5">5</option>' +
```

```
120             '</select>' +
```

```
121             '<input type="submit" style="cursor: pointer;" value="Buy">' +
```

```
122             '</form>' +
```

```
123             '</div>');
```

```
124 }
```

/fruitshop/code/mainpage.js

Excurs: Cookies vs Sessions

- Both store data for users not logged in/registered
- Benefits of Session:
 - Data is stored on the server
 - Store a lot more data
 - Higher level of security

Establishing Secure Registration

- Hashing algorithm Bcrypt

```
9 if (username !== null && pwd1 !== null && pwd1 === pwd2) {  
10     insertIntoDB();  
11 }
```

/fruitshop/code/create_user.js

Check Entered
Credentials

```
24     while (rs.next()) {  
25         if (rs.getBoolean(1)) {  
26             out.print('<p class="warning">User already exists!</p>');  
27             out.println(templateLoginButton());  
28             rs.close();  
29             prepstmt.close();  
30             conn.close();  
31             return; // stop the program  
32         }  
33     }
```

/fruitshop/code/create_user.js

Exit Program if
User Already
Exists

Excurs: Bcrypt

```
37 var bcrypt = Java.type("org.mindrot.jbcrypt.BCrypt");
38
39 // Generate a new salt, cost of the later hashpw will be 12
40 var salt = bcrypt.gensalt(12);
41
42 // Hash the password with the salt
43 var hashedPassword = bcrypt.hashpw(pwd1, salt);
```

Password Hash
creation

/fruitshop/code/create_user.js

```
sqlite> select password from customer;
$2a$12$CpYF.x.XRMGRls28dYBEoua9XUzVHbLrsSfyi1Ags0yYwwpIJPvmG
$2a$12$PeogW1ip6FW25A0tT86/f0jGy4umMvpEdU81yz94mR/xDYjcZYD0C
$2a$12$dUZMUU1c6CNbTitfGqC.VeePgeoYB7yQ69ueSconLjxt.nKRKiMh.
```

Example Hash

Login

```
28     var bcrypt = Java.type("org.mindrot.jbcrypt.BCrypt");
29     //check if password is correct
30     if (bcrypt.checkpw(pwd, hashedPassword)) {
31         //store the login status in the session
32         session.setAttribute('logged', id)
```

/fruitshop/code/login.js

Credential Check
& Login

```
42     //iterate through the shopping_cart that was stored in the session
43     for (var selection in shopping_cart) {
44         quantity = shopping_cart[selection];
45         quantity = parseInt(quantity);
46         item = selection;
```

/fruitshop/code/login.js

Accessing Items
Stored in Session

Creation of a Shopping Cart

Your Shopping Cart!

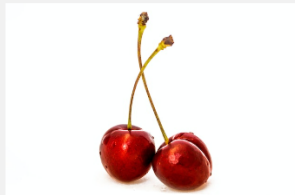
[Back to Main Page](#)

Ananas



Weight: 20 Kg
Price per item: 80 Euro
Quantity ordered: 4
Subtotal: 320 Euro

Cherry



Weight: 10 Kg
Price per item: 70 Euro
Quantity ordered: 4
Subtotal: 280 Euro

Your total is: 600 Euro

[Checkout](#)

Creation of a Shopping Cart - Continued

```
116● if (request.getParameter("actn") == "-") {  
117     var fruit_id = request.getParameter("fruit_id");  
118     var quantity = request.getParameter("quantity") - 1;  
119     // delete product from cart if quantity goes below 1  
120●   if (quantity <= 0) {  
121       var qry = "DELETE FROM shopping_cart WHERE customer_id = ? AND fruit_id = ?";
```

/fruitshop/code/shopping_cart.js

```
159● function getBackgroundColor(i) {  
160●   if ((i % 2) == 0) {  
161       var backgroundColor = '#F9F9F9';  
162●   } else {  
163       var backgroundColor = '#F1F1F1';  
164   }  
165   return backgroundColor;  
166 }
```

/fruitshop/code/shopping_cart.js

Functionality to
Reduce the
Quantity

Changing
Background
Color

Developed Applications – Advanced Examples

File Upload - Configuration

```
<servlet>
  <servlet-name>uploader</servlet-name>
  <jsp-file>/admin/code/uploader.jsp</jsp-file>
  <multipart-config>
    <location>C:\Users\Simon\Documents\apache-tomcat-10.0.23\files\</location>
    <max-file-size>10000000</max-file-size>
    <max-request-size>10000000</max-request-size>
  </multipart-config>
</servlet>

<servlet-mapping>
  <servlet-name>uploader</servlet-name>
  <url-pattern>/admin/uploader</url-pattern>
</servlet-mapping>
/fruitshop/WEB-INF/web.xml
```

File Upload

```
17 <form action="uploader" style="margin: 10px" enctype="multipart/form-data" method="post">
```

/fruitshop/admin/addproducts.html

```
if (checkIfProductExists()) {  
    var filename = name + ".jpg";  
    var location = "/files/" + filename;  
    request.getPart("file").write(filename);  
}
```

/fruitshop/admin/code/uploader.jsp

Enable
Uploading Files

Write File to the
Database

Newsletter – Prerequisites

- Jakarta Mail & Jakarta Activation
- MailHog
 - Purpose: E-mail developing and testing
 - SMTP server = port 1025
 - HTTP server = port 8025

From newsletter@treeshop.com
Subject **Here Are the Latest Products from treeshop!**
To mustermann@mail.com

HTML

Plain text

Source

Vist fruitshop

[Click Here to Unsubscribe](#)

Ananas



Price: 80 Euro

Weight: 20 KG

Cherry



Price: 70 Euro

Weight: 10 KG

Watermelon



Price: 40 Euro

Weight: 20 KG

Create a Newsletter

Admin Starting Page

- Ananas ☐
- Orange ☐
- Cherry ☐
- Banana ☐
- Guava ☐
- Watermelon ☐
- Coconut ☐
- Coconut2 ☐

Send Newsletter to 1 Receivers

Newsletter – Filter Chosen Products

```
5 var choices = request.getParameterValues("choice");
```

/fruitshop/admin/code/mailer.jsp

```
14 var choice = '';  
15 for (var i = 0; i < choices.length; i++) {  
16     // append all product names to a string  
17     choice += "" + choices[i] + ",";  
18 }
```

/fruitshop/admin/code/mailer.jsp

```
45 var qry2 = "SELECT * FROM fruit WHERE name IN (" + choice + ")";
```

/fruitshop/admin/code/mailer.jsp

Save Chosen
Products in a String &
Filter Select
Statement by this
String

Newsletter – E-mail Creation

```
while (customers.next()) {  
    var props = new (Java.type("java.util.Properties"))();  
    var session = Java.type("jakarta.mail.Session").getInstance(props);  
    var msg = new (Java.type("jakarta.mail.internet.MimeMessage"))(session);
```

/fruitshop/admin/code/mailer.jsp

Enables HTML Format

```
50     var productHTML = [];  
51  
52     while (products.next()) {  
53         var line1 = '<div style="float: left; margin-right: 10px;">';  
54         var line2 = '<h2>' + products.getString("name") + '</h2>';  
55         var line3 = '';  
56         var line4 = '<p>Price: ' + products.getString("price") + ' Euro</p>';  
57         var line5 = '<p>Weight: ' + products.getString("weight") + ' KG</p>';  
58         var line6 = '</div>';  
59  
60         productHTML.push(line1 + line2 + line3 + line4 + line5 + line6);  
61         i++;  
62     }
```

/fruitshop/admin/code/mailer.jsp

Iteration Over
Chosen
Products

Newsletter – Sending Email

```
70     for (var count = 0; count < i; count++) {  
71         //i is one more than the index in the array  
72         text += productHTML[count];  
73     }
```

/fruitshop/admin/code/mailer.jsp

Complete HTML
Code Saved as a
String

```
77     msg.setContent(text, "text/html");  
78  
79     var transport = session.getTransport("smtp");  
80     transport.connect("localhost", 1025, "username", "pw");  
81     transport.sendMessage(msg, msg.getRecipients(type));
```

/fruitshop/admin/code/mailer.jsp

Email Sent to
Port 1025
(MailHog)

- Presenting an alternative less popular approach for web development
- Projects of increasing complexity
 - Working with cookies
 - Developing an online shop
 - File upload
 - Sending e-mails
- Guidance for beginners