

CMPS 356 Enterprise Application Development - Spring 2019

Lab 8 – Client-side JavaScript

Objective

The objective of this lab is to practice implementing Web UI and calling Web API using Client-side JavaScript. Particularly DOM manipulation, events handling and using Fetch API.

Overview

This lab has two parts:

- **Part A:** Extend the Banking App to make the app functionality accessible via Web UI using Client-side JavaScript (1.5h).
- **Part B:** Extend the **BookStore** App to make the app functionality accessible via Web UI using Client-side JavaScript (1.5h).

Part A - Extend the Banking App to make the app functionality accessible via Web UI

1. Sync cmeps356-content repo to get the Lab files.
2. Copy **Lab8-ClientJS** folder from cmeps356-content repo to your repository.
3. Open **Lab8- WebApi \BankingApp** in WebStorm. Run **npm install** to install the packages.
4. Develop a Web UI to make the App functionality accessible to users:

Web Page	Functionally
index.html	<p>First, the index page should have a main navigation menu providing 3 links:</p> <ul style="list-style-type: none">• <u>Home</u> link is the home page that allows getting accounts.• <u>Add Account</u> link allows adding an account.• <u>Add Transaction</u> link allows adding a transaction. <p>- All the app pages should be accessible from the index.html page.</p> <p>- This page allows getting accounts by type. It provides a dropdown to select the account type: <i>Saving</i>, Current or All.</p> <p>When the page load or when a different account type is selected then the page should fetch the accounts from /api/accounts?acctType=</p> <p>- For each account with the balance=0, a <i>Delete</i> button is provided to enable deleting the account. Then this button is clicked the account should be deleted using /api/accounts/:id Web API. Also, the account row should be deleted from the html accounts table.</p>
acct-form.html	Allows creating a new account by posting the account data to /api/accounts
acct-trans.html	<p>Allows selecting a particular account from the accounts dropdown then submitting a deposit or withdrawal transaction.</p> <p>The new transaction should be posted to /api/accounts/:id/trans. The page should display any error returned from the Web API (e.g., insufficient balance).</p>

5. Test the Banking App Web UI and provide screenshots as evidence in the testing document.

Part B - Extend the Book Store to make the app functionality accessible via Web UI.

Deadline – Next week 1 hour before the Lab

1. Open **Lab8- WebApi \BookStore** in WebStorm. Run **npm install** to install the packages.
2. Design and implement a Web UI to make the App functionality accessible to users:

Important Note: The baseline solution will be posted on Wednesday at 1pm as the Lab B52 students will be submitting their assignment during that time. So, until the baseline solution is posted, you can use your own implementation.

Web Page	Functionality	
index.html	<p>First, the index page should have a main navigation menu providing three links:</p> <ul style="list-style-type: none"> - <u>Books</u> link is the home page that allows searching for books. - <u>Add Book</u> link allows adding a book. - Authors Book Summary link allows adding a book. <p>The index page should allow the user to enter the parameters to search for books by <i>category, name, isbn, author</i> or <i>pageCount</i>.</p> <p>The Book category and the Author parameter should be a dropdown filled with the data returned from <code>/api/categories</code> and <code>/api/authors</code> respectively.</p> <p>Upon submission of the search parameters, the app should fetch the books then display them in a card. Based on the selected search option the app should make use of the Web API developed in Lab 7 to fetch the books. You should use the Web API implementation in the provided base solution.</p> <p>Besides each book displayed in the cards, the app should offer the following links:</p> <ul style="list-style-type: none"> - details: this link/button should display the book details page. - update: this link/button should allow updating the book. - delete: this link/button should allow deleting the book. After the user confirmation, the app should delete the book using the Web API @ <code>/api/books/:isbn</code> 	
	Reminder of books API from Lab 7:	
	Get	<div><code>/api/books/:isbn</code></div> <div>Gets a book by isbn.</div>

	Get	/api/books?category=	Returns all the books for a particular category. If the category is not provided then you get all the books otherwise you should get the books that match that category. E.g. <i>category = Programming</i> should get all the programming books.								
	Get	/api/books?name=	Get the books where the name matches the name parameter.								
	Get	/api/books?pageCount=	Get the books with pages >= the pageCount parameter. E.g. if the pageCount=200 you should get all the books with pages >= 200.								
	Get	/api/books?author=	Display all the books authored by that specific author.								
book.html	This page displays the details of the selected book. The book details should be fetched using <i>/api/books/:isbn</i>										
book-editor.html	Provides a form to allow adding a book to the book-catalog. The entered book should be posted to the Web API @ <i>/api/books/</i> The form can also be used for updating an existing book. The updated book should be sent to the Web API @ <i>/api/books/</i>										
books-summary.html	Display a table that contains the author name and the number of books they have authored. The data for this page should be fetched using <i>Get /api/books/summary</i> <table><tr><th>Author Name</th><th>Book Count</th></tr><tr><td>James</td><td>2</td></tr><tr><td>Ali</td><td>4</td></tr><tr><td>Omar</td><td>7</td></tr></table>			Author Name	Book Count	James	2	Ali	4	Omar	7
Author Name	Book Count										
James	2										
Ali	4										
Omar	7										

Make sure the app main menu is accessible in all the app pages.

3. Test the **BookStore** Web UI and provide screenshots as evidence in the testing document.

After you complete the lab, fill in the **Lab8-TestingDoc-Grading-Sheet.docx** and save it in **Lab8-ClientJS** folder. Sync your repository to push your work to Github.

Sample screen shots of the app is provided below

Select Search Criteria

☐ Category ☐ Book Name ☐ ISBN ☐ Page Count

Search By Categories

Search



UNLOCKING ANDROID

Unlocking Android: A Developer's Guide provides concise, hands-on instruction for the Android operating system and development tools. This book teaches important architectural concepts in a straightforward writing style and builds on this with practical and useful examples throughout.

DETAILS

UPDATE

DELETE



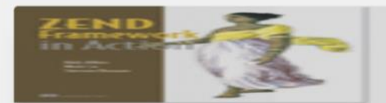
ANDROID IN ACTION, SECOND EDITION

Android in Action, Second Edition is a comprehensive tutorial for Android developers. Taking you far beyond "Hello Android," this fast-paced book puts you in the driver's seat as you learn important architectural concepts and implementation strategies. You'll master the SDK, build WebKit apps using HTML 5, and even learn to extend or replace Android's built-in features by building useful and intriguing examples.

DETAILS

UPDATE

DELETE



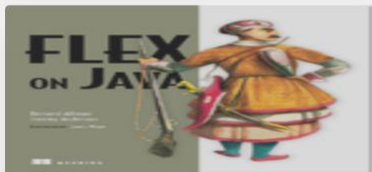
ZEND FRAMEWORK IN ACTION

Zend Framework in Action is a comprehensive tutorial that shows how to use the Zend Framework to create web-based applications and web services. This book takes you on an over-the-shoulder tour of the components of the Zend Framework as you build a high quality, real-world web application.

DETAILS

UPDATE

DELETE



FLEX ON JAVA

A beautifully written book that is a must have for every Java Developer. Ashish Kulkarni, Technical Director, E-Business Software Solutions Ltd.



GRIFFON IN ACTION

Griffon in Action is a comprehensive tutorial written for Java developers who want a more productive approach to UI development. In this book, you'll immediately dive into Griffon. After a Griffon orientation and a quick Groovy tutorial, you'll start building examples that explore Griffon's high productivity approach to Swing development. One of the troublesome parts of Swing development is the amount of Java code that is required to get a simple application off



OSGi IN DEPTH

Enterprise OSGi shows a Java developer how to develop to the OSGi Service Platform Enterprise specification, an emerging Java-based technology for developing modular enterprise applications. Enterprise OSGi addresses several shortcomings of existing enterprise platforms, such as allowing the creation of better maintainable and extensible applications, and provide a simpler, easier-to-use, lightweight solution to enterprise software development.

Book Store

BooksAdd BookAuthor Book Summary

Select Search Criteria

☐ Category
☐ Book Name
☒ Author Name
☐ ISBN
☐ Page Count

alexandre de castro alves
christian bauer
gavin king
clinton begin
brandon goodin
larry meadors
emmanuel bernard
john griffin
bear bibeault
yehuda katz
david a. black

OSGI IN DEPTH

Enterprise OSGi shows a Java developer how to develop to the OSGi Service Platform specification, an emerging Java technology for developing modular enterprise applications. Enterprise OSGi addresses several existing enterprise platforms, such as JBoss, and provides a simple, light-weight solution to enterprise development.

IBATIS IN ACTION

Gets new users going and gives experienced users in-depth coverage of advanced features. Jeff Cunningham, The Weather Channel Interactive

Book Store

BooksAdd BookAuthor Book Summary

Select Search Criteria

☒ Category
☐ Book Name
☐ Author Name
☐ ISBN
☐ Page Count

java
web development
programming
internet
microsoft .net
python
business
open source
powerbuilder
software engineering
mobile technology
object-technology programming
computer graphics
client-server

OSGI IN DEPTH

Enterprise OSGi shows a Java developer how to develop to the OSGi Service Platform specification, an emerging Java technology for developing modular enterprise applications. Enterprise OSGi addresses several existing enterprise platforms, such as JBoss, and provides a simple, light-weight solution to enterprise development.

IBATIS IN ACTION

Gets new users going and gives experienced users in-depth coverage of advanced features. Jeff Cunningham, The Weather Channel Interactive

Edit Book

Title	<input type="text" value="iBatis in Action"/>
ISBN	<input type="text" value="1932394826"/>
Page Count	<input type="text" value="384"/>
Published Date	<input type="text" value="mm / dd / yyyy"/>
thumbnailUrl	<input type="text" value="https://s3.amazonaws.com/AKIAJC5RLADLUMVRPFDQ.book-thumb-images/begin.jpg"/>
Short Description	<input type="text" value="Gets new users going and gives experienced users in-depth coverage of advanced features. Jeff Cunningham, The Weather Channel Interactive"/>
Long Description	<input type="text" value="Unlike some complex and invasive persistence solutions, iBatis keeps ORM clean and simple. It is an elegant persistence framework that maps classes to SQL statements and keeps the learning curve flat."/>
Book Status	<input type="text" value="PUBLISH"/>
Authors	<input type="text" value="Clinton Begin ; Brandon Goodin ; Larry Meadors"/>
Categories	<input type="text" value="Web Development"/>
<input type="button" value="SUBMIT"/>	

Add Book

Title	<input type="text"/>
ISBN	<input type="text" value="isbn"/>
Page Count	<input type="text" value="page count"/>
Published Date	<input type="text" value="mm / dd / yyyy"/>
thumbnailUrl	<input type="text"/>
Short Description	<input type="text"/>
Long Description	<input type="text"/>
Book Status	<input type="text" value="REVIEW"/>
Authors	<input type="text" value="Separate them by Comma"/>
Categories	<input type="text" value="Separate them by Comma"/>
<input type="button" value="SUBMIT"/>	

Book Store		Books	Add Book	Author Book Summary
Author Name	No Of Books			
Alexandre de Castro Alves	1			
Christian Bauer	1			
Gavin King	1			
Clinton Begin	1			
Brandon Goodin	1			
Larry Meadors	1			
Emmanuel Bernard	1			
John Griffin	1			
Bear Bibeault	1			
Yehuda Katz	2			
David A. Black	1			
Paul A. Boag	1			