Exercise: DVD Library

Overview

Project source code: **Wireframes.pdf (https://lms.thesoftwareguild.com/courses/368/files/72000/download?wrap=1)** **(https://lms.thesoftwareguild.com/courses/368/files/72000/download?wrap=1)**

Project source code: **DVDLibraryRESTService.zip (https://lms.thesoftwareguild.com/courses/368/files/72156/download?wrap=1)**

The purpose of this exercise is to show your proficiency in using HTML, CSS, JavaScript, and jQuery to create a browser-based user interface for an existing web service. You will show your ability to use these skills by building a page that allows a user to create, read, update, and delete DVD information via the supplied DVD Library REST API.

In order to complete this lab successfully, you will have to analyze and understand the DVD Library API. The details of the API are outlined later in this document.

> It is highly recommended that you use Postman to explore the API before you begin coding!

Requirements

Your assignment is to create an HTML page (and the associated CSS and JavaScript files) that matches the functionality outlined in the requirements and wireframes below.

- Your HTML file must be called **home.html**.
- Your JavaScript file must be called **home.js**.
- Use Postman to explore the API before you begin coding.
- The DVD Search Category drop-down menu choices are: Title, Release Year, Director Name, Rating.
- The Search drop-down must default to “Search Category”, which must be non-selectable.
- Search Category and Search Term are both required fields.
- The DVD Title field must be required for both the Create and Edit forms.
- You must restrict the Release Year entry field to digits only for both the Create and Edit forms.
- You must require that the Release Year entry field is exactly 4 digits long for both the Create and Edit forms.
- You must use HTML5 validation and JavaScript code to validate the form input and display the error messages. Follow the pattern shown in the lessons.

Starting the Dvd Library Web Service

You were supplied with an executable JAR file containing the Dvd Library REST Web Service. This JAR file is similar to the one that has been used throughout the code-alongs for this section. As was the case for the code-alongs, you must start the web service before you can access it. To start the web service, type the following command into your shell:

```
java -jar DvdLibraryRESTWebService.jar
```

Verifying Dvd Library Web Service Startup

To verify that your web service started successfully, open Postman and send a GET request to **localhost:8080/dvds**. If the service is started, you should see a list of sample DVD JSON objects returned.

Wireframes

Please reference the wireframes downloaded above to supplement this section.

Initial State

Upon initial load, your page should display in the initial state.

- The page is separated horizontally by a rule into a top row and a content area.
- The top row contains columns containing the following:
  - A Create DVD link is present on the left.
  - A search category select element and a search term input element are placed to the right along with a search button.
- A horizontal rule separates the top row from the bottom portion
- The bottom portion is the details area which contains a table of DVD records. The columns from left to right are Title, Release Date, Director, Rating, and an unlabeled column whose rows contain edit and delete elements that are styled to look like links. The DVD titles are links that load a details view.

Search Category/Term Validation Error

Both the search category and the search term are required to perform the search via the search button. If either is not provided a bordered div appears above the DVD records table that has a light red background color and contains an error message: Both Search Category and Search Term are required.

Delete Confirmation

If a user clicks the delete link in a row of the DVD list a confirmation dialogue appears with the message "Are you sure you want to delete this DVD from your collection?".

Edit Dvd Display

This is an HTML form that appears when a user clicks the edit link on a DVD. The form contains labels and elements for:

- DVD Title - Text input.
- Release Year - Text input.
- Director - Text input.
- Rating - Select element, defaults to G.
- Notes - TextArea element.
- Cancel button - Shows the DVD list view.
- Save Changes button - Submits the data to the REST Service for updating.

Validation errors appear above the form in a bordered div above the edit form similar to the DVD list page that display validation errors for the title and release year. The title is required and the release year must have 4-digits. The error messages are 'Please enter a title for the DVD' and 'Please enter a 4-digit year' respectively.

Create Dvd Display

This is an HTML form that appears when a user clicks the Create DVD link. The form contains labels and elements for:

- DVD Title - Text input.
- Release Year - Text input.
- Director - Text input.
- Rating - Select element, defaults to G.
- Notes - TextArea element.
- Cancel button - Shows the DVD list view.
- Save Changes button - Submits the data to the REST Service for creation.

Validation errors appear above the form in a bordered div above the edit form similar to the DVD list page that display validation errors for the title and release year. The title is required and the release year must have 4-digits. The error messages are 'Please enter a title for the DVD' and 'Please enter a 4-digit year' respectively.

Dvd Details Display

This displays HTML containing additional details about the DVD. It is displayed when a user clicks the DVD's title on the DVD list. The DVD title is a header on the page with a horizontal rule underneath follow by labels and values for the DVD record:

- DVD Title - Text input.

- Release Year - Text input.
- Director - Text input.
- Rating - Select element, defaults to G.
- Notes - TextArea element.

A back button appears beneath the data that returns the user to the DVD list view.

DVD Web Service API

The following are descriptions of the DVD web service API calls:

Retrieving a DVD by Id	
HTTP Method	GET
URL	/dvd/{id}
Path Variable	{id} – Id of the requested DVD
Request Body:	None

Response Body	**JSON representation of a DVD**
	{
	  “dvdId”: 0,
	  “title”: “A Great Tale”,
	  “realeaseYear”: 2015,
	  “director”: “Sam Jones”,
	  “rating”: “PG”,
	  “notes”: “This really is a great tale!”
	}

Retrieving all Dvds	
HTTP Method	GET
URL	/dvds
Path Variable	None
Request Body	None

Response Body	**JSON array of Dvds**
	[
	  {
	    "dvdId": 0,
	    "title": "A Great Tale",
	    "realeaseYear": 2015,

Retrieving all Dvds

```
    "director": "Sam Jones",
    "rating": "PG",
    "notes": "This really is a great
tale!"
  },
  {
    "dvdId": 1,
    "title": "A Good Tale",
    "realeaseYear": 2012,
    "director": "Joe Smith",
    "rating": "PG-13",
    "notes": "This is a good tale!"
  }
]
```

Retrieving Dvds by Title

HTTP Method:	GET
URL	/dvds/title/{title}
Path Variable	{title} – Title search term
Request Body	None

Response Body

JSON array of Dvds that match the search criteria:

```
[
  {
    "dvdId": 0,
    "title": "A Super Tale",
    "realeaseYear": 2015,
    "director": "Sam Jones",
    "rating": "PG",
    "notes": "A great remake!"
  },
  {
    "dvdId": 1,
    "title": "A Super Tale",
    "realeaseYear": 1985,
    "director": "Joe Smith",
    "rating": "PG",
    "notes": "The original!"
  }
]
```

Retrieving Dvds by Release Year

HTTP Method	GET
URL	/dvds/year/{releaseYear}
Path Variable	{releaseYear} – Release year search term

Retrieving Dvds by Release Year

Request Body None

Response Body

JSON array of Dvds that match the search criteria:

```
[
  {
    "dvdId": 0,
    "title": "A Great Tale",
    "realeaseYear": 2015,
    "director": "Sam Jones",
    "rating": "PG",
    "notes": "This really is a great tale!"
  },
  {
    "dvdId": 1,
    "title": "A Wonderful Tale",
    "realeaseYear": 2015,
    "director": "Joe Smith",
    "rating": "PG-13",
    "notes": "Wonderful, just wonderful!"
  }
]
```

Retrieving Dvds by Director Name

HTTP Method	GET
URL	/dvds/director/{directorName}
Path Variable	{directorName} – Director name search term
Request Body	None

Response Body

JSON array of Dvds that match the search criteria:

```
[
  {
    "dvdId": 0,
    "title": "A Great Tale",
    "realeaseYear": 2015,
    "director": "Sam Jones",
    "rating": "PG",
    "notes": "This really is a great tale!"
  },
  {
    "dvdId": 1,
    "title": "A Wonderful Tale",
    "realeaseYear": 2015,
    "director": "Joe Smith",
    "rating": "PG-13",
    "notes": "Wonderful, just wonderful!"
  }
]
```

Retrieving Dvds by Rating	
HTTP Method	GET
URL	/dvds/rating/{rating}
Path Variable	{rating} – Rating search term
Request Body	None

Response Body:

JSON array of Dvds that match the search criteria:

```
[
  {
    "dvdId": 0,
    "title": "A Great Tale",
    "realeaseYear": 2015,
    "director": "Sam Jones",
    "rating": "PG",
    "notes": "This really is a great tale!"
  },
  {
    "dvdId": 1,
    "title": "Just A Tale",
    "realeaseYear": 2015,
    "director": "Joe Baker",
    "rating": "PG",
    "notes": "It is a tale!"
  }
]
```

Creating a new DVD	
HTTP Method	POST
URL	/dvd
Path Variable	None

Request Body

JSON representation of DVD to be created (NOTE – this does not include a dvdId):

```
{
    "title": "A New Tale",
    "realeaseYear": 2016,
    "director": "Jack Jameson",
    "rating": "PG-13",
    "notes": "Brand spanking new!"
}
```

Response Body:

JSON representation of the newly created DVD (NOTE – this does include the system assigned dvdId):

```
{
    "dvdId": 42,
    "title": "A New Tale",
    "realeaseYear": 2016,
    "director": "Jack Jameson",
```

Creating a new DVD	

	`"rating": "PG-13",`
	`"notes": "Brand spanking new!"`
	`}`

Updating an Existing DVD	
HTTP Method	PUT
URL	/dvd/{id}
Path Variable	{id} – Id of DVD to be updated

Request Body	**JSON representation of DVD to be created (NOTE – this does include a dvdId)**:
	`{`
	`"dvdId": 42,`
	`"title": "A New Tale",`
	`"realeaseYear": 2016,`
	`"director": "Jack Jameson",`
	`"rating": "PG-13",`
	`"notes": "I'm just changing my note..."`
	`}`

Response Body	None

Deleting a DVD	
HTTP Method	DELETE
URL	/dvd/{id}
Path Variable	{id} – Id of DVD to be deleted
Request Body	None
Response Body	None

This is a practice exercise and is not part of your grade. If you need help or feedback on this exercise, please visit open office hours.