Date: 06-Feb-2020

Proof of Concept

on

JavaScript CRUD POC

A basic CRUD operation requires data. CRUD stands for Create, Read, Update and Delete and these are four basic functions to manipulate data in a database. It can be any database. For my CRUD application, I am using data in a JSON object.

*1)* I have data stored in a JSON object (in my application). I’ll extract the data and display it in a [dynamically created HTML table](https://www.encodedna.com/javascript/dynamically-add-remove-rows-to-html-table-using-javascript-and-save-data.htm).

*2)* Each row has few more [dynamically created HTML elements](https://www.encodedna.com/javascript/dynamically-create-html-elements-using-createElement-method.htm) like *buttons, textboxes* and *Dropdown Lists*, to perform functions like, update, delete, save, create new and cancel.

*3)* Button’s and textboxes will have events attached, dynamically.

**What does this application do?**

The application is a *Books inventory*. It will show a list of books, in a distinct category with a price. See the above image. This Application allows users to View the list, create or add new books to the existing list (database), update or edit a rows data and *delete* the data (the entire row).

Every transaction will affect an Array of JSON objects, a temporary database.

**Functions for CRUD operation**

 I have *five* different functions to perform different operations. These functions are

*1)* this.Cancel () – This will cancel the update procedure. Every row of the table has an *Update* button. Clicking this button will show two more buttons, to cancel and to save. Clicking the cancel button will call the function *this.cancel()* that takes a parameter as the calling element.

Using the object’s reference, you can get the active row, its elements and values.

*2)* this.Update () – This functions is called when you click the *Update* button in any row. It only shows the input elements like textbox and a dropdown list with values. So now, you can edit the values in each cell of the selected row (except the first column).

*3)* this.Delete() – This function uses [JavaScript splice() method](https://www.encodedna.com/angularjs/tutorial/how-to-push-new-elements-in-between-existing-ng-repeat-array.htm) to remove the data from JSON array.

The *splice()* usually takes three parameters, explained [here](https://www.encodedna.com/angularjs/tutorial/how-to-push-new-elements-in-between-existing-ng-repeat-array.htm#splice). I have provided two. The first parameter is the *location* in the array (that’s why I am using activeRow - 1), which I want to remove. The second parameter is the *number of items* (I’ve *1*) that I want to remove from the array. Add the value *2* and see what happens.

*4)* this.Save() - This function will update or save the rows data in the Array. It is associated with *Update* operations. The *Save* option is activated when the user click’s the *Update* button in any row.

*5)* this.Create() – The last row has blank boxes and a button named *Create* at the last column. Clicking the *Create* button will call this function and it will add a new set of data in the exiting list inside the *myBooks* array.