

Create your WEB Application with MVC paradigm and OPP – PAP support

Tutorial 18

In this tutorial, you will create a state function. Imagine you want to show on the front end if a product or category is enabled or disabled. You have already started to implement this feature, but you need to create a function to change the state.

First, you need to change the view. In the last tutorial, you started to make this change. You should already have a marker that is disabled if the database number is 1 and enabled otherwise.

Now, let's introduce an onclick function to automatically make that change when the user clicks on the marker.

Ok, adapt your code as shown in the example:

```

<tbody>
<?php if (!empty($data['categories'])): ?>
  <?php foreach ($data['categories'] as $index => $category): ?>
    <?php $status = $category['disabled'] ? "Disabled" : "Enabled"; ?>
    <tr>
      <td><?php echo $index + 1; ?></td>
      <td><?php echo htmlspecialchars($category['category']); ?></td> <!-- Access as array -->
      <td>
        <?php if ($category['disabled']): ?>
          <span class="label label-warning label-mini" style="cursor:pointer"
            onclick="disabled_row(<?php echo $category['id']; ?>, <?php echo $category['disabled']; ?>)">Disabled</span>
        <?php else: ?>
          <span class="label label-success label-mini" style="cursor:pointer"
            onclick="disabled_row(<?php echo $category['id']; ?>, <?php echo $category['disabled']; ?>)">Enabled</span>
        <?php endif; ?>
      </td>
    </tr>
  </foreach>
</if>

```

This code will give you the option to call a function `disabled_row()` when the user clicks. In this function, you send the ID, and if the category is enabled, it will be turned to disabled, and vice versa.

Now add this function.

```

// Function to change the state
function disabled_row(id,state)
{
    //console.log(state);
    send_data(data = {
        data_type:"disabled_row",
        id:id,
        current_state:state
    })
}

```

In this function, you will get the change from the user, and send the state to the controller.

In the controller next the delete, from the other tutorial, update your code to receive this ajax call:

```
// Handle deleting a category
elseif ($data->data_type == 'delete_row') {

    $check = $category->delete($data->id);
    if ($check) {
        $arr['message'] = "A sua categoria foi removida com sucesso!";
        $arr['message_type'] = "info";
    } else {
        $arr['message'] = "Erro ao remover a categoria!";
        $arr['message_type'] = "error";
    }
    $_SESSION['error'] = "";
    $arr['data'] = "";
    $arr['data_type'] = "delete_row";

    echo json_encode($arr);
}

// Handle change state of a category
elseif ($data->data_type == 'disabled_row') {
    $id = $data->id;
    $disabled = $data->current_state ? 0 : 1;

    $query = "UPDATE categories SET disabled = :disabled WHERE id = :id LIMIT 1";
    $params = array(':disabled' => $disabled, ':id' => $id);
    $DB = Database::getInstance();
    $DB->write($query, $params);

    $arr['message'] = "";
    $_SESSION['error'] = "";
    $arr['message_type'] = "info";
    $arr['data'] = "";
    $arr['data_type'] = "disabled_row";

    echo json_encode($arr);
}
```

This code handles changing the state of a category, specifically enabling or disabling the category, and sends a JSON response back to the client to confirm the action.

First, the code checks if the received action type (`data_type`) is `'disabled_row'`. If it is, it proceeds with the corresponding block of code. Next, it extracts the ID of the category to be modified from the `$data` object.

Then, it determines the new state of the category based on the current state (`current_state`). If `current_state` is true (1), the new state (`disabled`) will be 0, which means the category will be enabled. If `current_state` is false (0), the new state (`disabled`) will be 1, disabling the category.

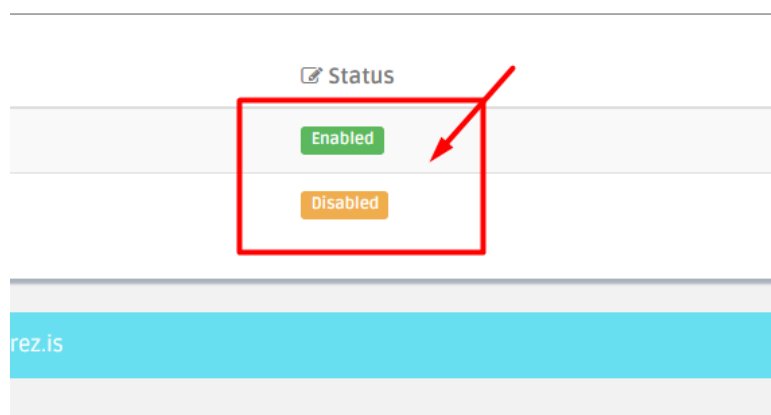
Next, it prepares an SQL query to update the `disabled` field in the `categories` table for the specific ID. It creates an associative array `$params` that maps the placeholders `:disabled` and `:id` to their respective values.

It then obtains a single instance of the Database class (likely implemented as a singleton) and calls the write method of that instance, passing the SQL query and the parameters. This method executes the update query in the database.

After executing the query, it prepares an associative array \$arr to be sent back as a JSON response. It sets message to an empty string, resets \$_SESSION['error'] to an empty string, sets message_type to "info", data to an empty string, and data_type to "disabled_row".

Finally, it converts the \$arr array to JSON and sends it as a response to the AJAX request, confirming that the category state has been successfully changed.

Now, please test your result:



ASSIGNMENT: now you should update on method, to lead with the status. Imagine the status is enabled, and the user want to delete the category. You should provide a error message, and change the function in controller and in the view and also in the model, to solve this.

TIP: look to your delete functions.

Remember, always comment your code, to facilitate in the future to read them.

END Tutorial 18

Reference: **PHP Ecommerce website development | Change state of the category | MVC OOP - Quick programming**

Template: NICEADMIN, design by BOOTSTRAP

Modal: <https://getbootstrap.com/docs/4.0/components/modal/>