Drag and Drop Images

jQuery, Ajax, PHP & MySQL

Today we'll discuss the most useful topic for the web application. jQuery drag and drop are very popular and user-friendly feature. We will build the images reorder functionality using jquery drag and drop feature. Also, we will use Ajax, PHP and MySQL for saving the images order at the database.

This tutorial would be very helpful for your web project. In this tutorial, you will learn how to build drag and drop reorder images using jQuery, Ajax, PHP and MySQL. You would be able to provide a better user interface for your web project. We have been developed the simple scripts for dynamic drag and drop reorder. Using our scripts you can also implement the drag and drop reorder list, rows or sorting elements. You can use this functionality for managing images gallery, managing users list or any other useful place.

We have been used 4 technologies for build the whole system – **jQuery**, **Ajax**, **PHP** and **MySQL**. From the above**Demo** link you would be able to view the live demo of drag & drop reorder images, also able to download the full project from the above **Download** link.

This project has one folder for images, two jQuery files, one css file and three PHP files. The folders and files structure are given below.

- `index.php`
- `db.php`
- `order_update.php`
- `style.css`
- `jquery-1.8.3.min.js`

  - `jquery-ui.js`

- images/

Database table creation:

To store the images order, you need to create a database and a table. At first create a database like `codexworld`. After that create a table named `images` or you can just copy the below SQL query and run the SQL query into the database(codexworld).

```
CREATE TABLE `images` (

    `id` int(11) NOT NULL AUTO_INCREMENT,

    `image` varchar(255) COLLATE utf8_unicode_ci NOT NULL,

    `order` int(5) NOT NULL DEFAULT '0',

    `created` datetime NOT NULL,

    `modified` datetime NOT NULL,

    `status` tinyint(1) NOT NULL DEFAULT '1',

    PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Once the `images` table creation is completed, you can store the images at this table.

Db.php file:

This file is used for managing the data from the database. In this file we have been created a class named “DB” and used some functions for get the images rows and update the images order. Also you can declare any other database related functions as per your application needs.

db.php file:

`db.php` file have `DB` class with some functions. Into the `__construct()` function we have configured the database and connect with the database. You need only to change the `$dbServer`, `$dbUsername`, `$dbPassword` and `$dbName`variables value with your MySQL server details.

`getRows()` function is used to get image rows from the **images** table.

`updateOrder()` function is used to update images order into the **images** table.

```
<?php
class DB{
    function __construct(){
        //database configuration
        $dbServer = 'localhost'; //Define database server host
        $dbUsername = 'root'; //Define database username
        $dbPassword = ''; //Define database password
        $dbName = 'codexworld'; //Define database name

        //connect databse
        $con = mysqli_connect($dbServer,$dbUsername,$dbPassword,$dbName);
        if(mysqli_connect_errno()){
            die("Failed to connect with MySQL: ".mysqli_connect_error());
        }else{
            $this->connect = $con;
        }
    }

    function getRows(){
        $query = mysqli_query($this->connect,"SELECT * FROM `images` ORDER BY `orde
r` ASC") or die(mysql_error());
        while($row = mysqli_fetch_assoc($query))
        {
            $rows[] = $row;
        }
        return $rows;
    }

    function updateOrder($id_array){
        $count = 1;
        foreach ($id_array as $id){
            $update = mysqli_query($this->connect,"UPDATE `images` SET `order` = $count WHERE id = $id");
            $count ++;
        }
        return true;
    }
}
?>
```

`index.php` file:

In the `index.php` file we will include the `db.php` file for database access. Using of `DB` class we will get image rows from database and display here. please see the below step by step codes section.

- Include the `db.php` file and create an instance of DB class.

```
<?php
include_once("db.php");
```

```
$db = new DB();

?>
```

- Add jquery library `jquery-1.8.3.min.js` and jquery UI `jquery-ui.js` links at the page head section.

```
<script type="text/javascript" src="jquery-1.8.3.min.js"></script>

<script type="text/javascript" src="jquery-ui.js"></script>
```

- Following jQuery codes are used to enable the jQuery `sortable()` features and implement the drag & drop functionality. Ajax function is used for send the images order to `order_update.php` file and get the response from this file.

```
<script type="text/javascript">

$(document).ready(function(){

    $('.reorder_link').on('click',function(){

        $("ul.reorder-photos-list").sortable({ tolerance: 'pointer' });

        $('.reorder_link').html('save reordering');

        $('.reorder_link').attr("id","save_reorder");

        $('#reorder-helper').slideDown('slow');

        $('.image_link').attr("href","javascript:void(0);");

        $('.image_link').css("cursor","move");

        $("#save_reorder").click(function( e ){

            if( !$("#save_reorder i").length )

            {

                $(this).html('').prepend('<img src="images/refresh-animated.gif"/>');
                $("ul.reorder-photos-list").sortable('destroy');
                $("#reorder-helper").html( "Reordering Photos - This could
take a moment. Please don't navigate away from this
page." ).removeClass('light_box').addClass('notice notice_error');
                var h = [];
                $("ul.reorder-photos-list li").each(function() {  h.push($
(this).attr('id').substr(9));  });
                $.ajax({
                    type: "POST",
                    url: "order_update.php",
                    data: {ids: " " + h + ""},
                    success: function(html)
                    {
                        window.location.reload();
                    }
                });
                return false;
            }
            e.preventDefault();
        });
    });
});
</script>
```

- Get the images from the database and display the all images using PHP.

```
<a href="javascript:void(0);" class="btn outlined mleft_no reorder_link"
id="save_reorder">reorder photos</a>
```

```
<div id="reorder-helper" class="light_box" style="display:none;">1. Drag photos to reorder.<br>2. Click 'Save Reordering' when finished.</div>

<div class="gallery">

    <ul class="reorder_ul reorder-photos-list">

    <?php

        //Fetch all images from database

        $rows = $db->get_rows();

        foreach($rows as $row): ?>

        <li id="image_li_<?php echo $row['id']; ?>" class="ui-sortable-handle"><a href="javascript:void(0);" style="float:none;" class="image_link"><img src="images/<?php echo $row['image']; ?>" alt=""></a></li>

    <?php endforeach; ?>

    </ul>

</div>
```

Images reorder (order_update.php):

order_update.php file receive the current images order from the index.php through Ajax and update the images reorder at the database.

- Include the db.php file and create an instance of DB class.

```
include_once('db.php');

$db = new DB();
```

- Breaks the images ids string into an array.

```
$idArray = explode(",",$_POST['ids']);
```

- Call the `updateOrder()` function and pass the ids array for update the images order at the database table.

```
$db->updateOrder($idArray);
```

- The complete codes of the `order_update.php` file is given below.

```
<?php

include_once('db.php');

$db = new DB();

$idArray     = explode(",",$_POST['ids']);

$db->updateOrder($idArray);

?>
```

Courtesy:

http://www.codexworld.com/drag-drop-images-reorder-using-jquery-ajax-php-mysql/