Crear Un Cargador De Archivos Con jQuery

La carga de archivos es algo que, como desarrolladores, hemos tenido que hacer durante años y conseguir un cargador de archivos ajax es siempre una mejor opción, en lugar de redirigir al usuario a una página de carga o incluso volver a cargar la página; mantener al usuario en la misma página mejorará la usabilidad de la aplicación. Es por esto que hoy vamos a crear un sencillo **cargador de archivos ajax** con soporte para drag and drop.

El Plugin

En este ejemplo voy a utilizar el plugin jQuery Upload File, porque es fácil de usar y robusto. Cuando se trata de la creación de **plugins,** es tan simple como dice la página web, lo primero que necesitamos incluir es jQuery, el archivo JS y también el CSS, para este ejemplo vamos a descargar esta copia del archivo JS para dar cabida a las imágenes previas, de la siguiente manera:

```
[html]
<link href="css/uploadfile.css" rel="stylesheet">
<script src="js/jquery.uploadfile.min.js"></script>
[/html]
```

Una vez hecho esto tenemos que configurar el HTML y el uso de este **plugin** todo lo que realmente necesitamos hacer es crear un **div** vacío y el plugin se encargará del resto. Nosotros no tenemos que preocuparnos de escribir todo el formulario de subida, así que nuestra etiqueta **HTML** quedaría de la siguiente forma:

[html] Upload [/html]

Esto es todo en la parte del html, ahora falta hacerlo funcionar, para esto usaremos **javaScript**.

El JavaScript

Lo primero que tenemos que hacer después de que comprobemos si el documento está listo es llamar a la **función UploadFile** a nuestro div vacío:

[js]

```
$(document).ready(function() {
$("#upload").uploadFile({
// Parameters here
});
});
```

[/js]

Ahora tenemos que pensar en qué tipo de funcionalidad deseamos para este formulario de subida, en primer lugar, queremos que se ejecute la secuencia de comandos **upload.php** por lo que necesitamos el parámetro url con el enlace a ese fichero, entonces queremos permitir múltiples cargas de archivos, también queremos que el usuario solo cargue imágenes:

[js]

```
$("#upload").uploadFile({
url:"upload.php",
multiple: true,
allowedTypes: "jpg,jpeg,png,gif"
});
```

[/js]

Lo anterior es una configuración básica, pero si queremos personalizar errores o eventos satisfactorios? Para esto

aplicaremos las siguiente lineas:

[js]

```
$("#upload").uploadFile({
url:"upload.php",
multiple: true,
allowedTypes: "jpg,jpeg,png,gif",
doneStr:"Cargado !",
extErrorStr:"Solo puedes realizar carga de archivos! ",
uploadErrorStr:"Ocurrio un error al carga. Intentelo de nuevo!"
});
```

[/js]

En esta parte hemos cambiado el mensaje cuando se carga el archivo, el valor predeterminado es «Done!» y con **doneStr** cambiamos ese mensaje a «Cargado !» También se utilizo el **extErrorStr y uploadErrorStr** para cambiar el mensaje de error cuando el usuario intenta subir algo que no sea una imagen con **uploadErrorStr**. Eso es todo el código JavaScript que necesitamos, ahora sólo tenemos que añadir el CSS para mejorar la apariencia de nuestro cargador.

El CSS

Nuestro cargador es funcional y ahora sólo tenemos que hacer algunos cambios para hacer nuestra página más atractiva. Se añadió algunos elementos, como un titulo, se rodeo el div, y todo esto se adorno de la siguiente forma:

[css]

```
body {
background: #2d3e50;
}
section {
width: 45%;
```

```
margin: auto;
margin-top: 50px;
}
h1 {
Open Sans', Arial, serif;
color: #edf1f4;
font-weight: 400;
text-align: center;
}
```

[/css]

Como puedes ver, se cambiado el fondo de la página a un color azul, se cambio el ancho para ***section*** y se ha centrado en la página, y luego se añadio algunos estilos a nuestros **h1**. Siguiendo cambiamos el color de nuestro botón de subida:

[css]

```
.ajax-file-upload {
Open Sans', Arial, serif;
background: #e84c3d;
box-shadow: 0 2px 0 0 #c13929;
color: #edf1f4;
}
.ajax-file-upload:hover {
background: #d24336;
box-shadow: 0 2px 0 0 #a62f23;
}
.ajax-upload-dragdrop {
margin: auto;
}
```

[/css]

Después de esto tenemos que colocar nuestra imagen correctamente para que la barra de texto y el progreso flote a la derecha. También cambiar el color del texto que muestra el nombre del archivo subido, fondo de la barra de carga y luego

algunos toques finales:

[css]

```
.ajax-file-upload-image img {
float: left;
margin-right: 10px;
border-radius: 3px;
transition: all 0.2s;
-moz-transition: all 0.2s;
-webkit-transition: all 0.2s;
}
.ajax-file-upload-image img:hover {
box-shadow: 1px 0 5px rgba(0,0,0,0.5);
}
.ajax-file-upload-filename {
color: #edf1f4;
}
.ajax-file-upload-bar {
background-color: #8dbf4a;
}
.ajax-file-upload-progress {
border: none;
margin: 0;
margin-right: 10px;
width: 45%;
}
.ajax-file-upload-statusbar {
margin: 20px auto;
border: 1px solid rgba(255,255,255,0.5);
height: 100px;
}
```

[/css]

Eso es todo, el resultado seria algo como se muestra en la siguiente imagen: