multiple file upload via ajax drag-drop with jQuery and PHP. only images and PDFs. and under a div show uploaded files list with download and delete option in table.

Multiple File Upload via AJAX Drag-Drop with jQuery and PHP

Here's a complete solution for multiple file uploads with drag-and-drop functionality, restricted to images and PDFs, with a file listing that includes download and delete options.

HTML (index.html)

html Copy Download Run

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Drag & Drop File Upload</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        #drop-area {
            border: 2px dashed #ccc;
            border-radius: 5px;
            padding: 25px;
            text-align: center;
            margin: 20px auto;
            max-width: 600px;
        }
        #drop-area.highlight {
            border-color: #4CAF50;
            background-color: #f8f8f8;
        }
        .file-list {
            margin: 20px auto;
            max-width: 800px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
        .progress-bar {
            width: 100%;
            background-color: #f1f1f1;
            border-radius: 5px;
            margin-top: 5px;
        }
        .progress {
            height: 20px;
            background-color: #4CAF50;
            border-radius: 5px;
```

```
            width: 0%;
            text-align: center;
            line-height: 20px;
            color: white;
        }
        .btn {
            padding: 5px 10px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
            color: white;
            text-decoration: none;
            display: inline-block;
        }
        .btn-download {
            background-color: #2196F3;
        }
        .btn-delete {
            background-color: #f44336;
        }
    </style>
</head>
<body>
    <div id="drop-area">
        <h3>Drag & Drop Files Here</h3>
        <p>(Only images and PDFs allowed)</p>
        <input type="file" id="fileElem" multiple accept="image/*,.pdf" style="display:none">
        <button id="browseBtn">Select Files</button>
    </div>

    <div class="file-list">
        <h3>Uploaded Files</h3>
        <table id="fileTable">
            <thead>
                <tr>
                    <th>File Name</th>
                    <th>Type</th>
                    <th>Size</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody id="fileList">
                <!-- Files will be listed here -->
            </tbody>
        </table>
    </div>

    <script>
    $(document).ready(function() {
        let dropArea = document.getElementById('drop-area');
        let fileElem = document.getElementById('fileElem');
        let browseBtn = document.getElementById('browseBtn');

        // Prevent default drag behaviors
        ['dragenter', 'dragover', 'dragleave', 'drop'].forEach(eventName => {
            dropArea.addEventListener(eventName, preventDefaults, false);
            document.body.addEventListener(eventName, preventDefaults, false);
        });

        // Highlight drop area when item is dragged over it
        ['dragenter', 'dragover'].forEach(eventName => {
            dropArea.addEventListener(eventName, highlight, false);
        });

        ['dragleave', 'drop'].forEach(eventName => {
            dropArea.addEventListener(eventName, unhighlight, false);
        });

        // Handle dropped files
```

```
dropArea.addEventListener('drop', handleDrop, false);

// Handle file selection via button
browseBtn.addEventListener('click', function(e) {
    fileElem.click();
});

fileElem.addEventListener('change', function(e) {
    handleFiles(this.files);
});

function preventDefaults(e) {
    e.preventDefault();
    e.stopPropagation();
}

function highlight() {
    dropArea.classList.add('highlight');
}

function unhighlight() {
    dropArea.classList.remove('highlight');
}

function handleDrop(e) {
    let dt = e.dataTransfer;
    let files = dt.files;
    handleFiles(files);
}

function handleFiles(files) {
    files = [...files];
    files.forEach(uploadFile);
}

function uploadFile(file) {
    // Check file type
    let validTypes = ['image/jpeg', 'image/png', 'image/gif', 'application/pdf'];
    if (!validTypes.includes(file.type)) {
        alert('Only images (JPEG, PNG, GIF) and PDF files are allowed!');
        return;
    }

    // Check file size (max 5MB)
    if (file.size > 5 * 1024 * 1024) {
        alert('File size exceeds 5MB limit!');
        return;
    }

    let url = 'upload.php';
    let formData = new FormData();
    formData.append('file', file);

    // Create a unique ID for this upload
    let fileId = 'file-' + Date.now() + '-' + Math.random().toString(36).substr(2, 9);

    // Add row to table
    let row = document.createElement('tr');
    row.id = fileId;
    row.innerHTML = `
        <td>${file.name}</td>
        <td>${file.type}</td>
        <td>${formatFileSize(file.size)}</td>
        <td>
            <div class="progress-bar">
                <div class="progress" id="${fileId}-progress">0%</div>
            </div>
            <span id="${fileId}-status">Uploading...</span>
        </td>
    `;
```

```
            document.getElementById('fileList').appendChild(row);

            $.ajax({
                url: url,
                type: 'POST',
                data: formData,
                processData: false,
                contentType: false,
                xhr: function() {
                    let xhr = new XMLHttpRequest();
                    xhr.upload.addEventListener('progress', function(e) {
                        if (e.lengthComputable) {
                            let percentComplete = Math.round((e.loaded / e.total) * 100);
                            $(`#${fileId}-progress`).width(percentComplete + '%');
                            $(`#${fileId}-progress`).text(percentComplete + '%');
                        }
                    }, false);
                    return xhr;
                },
                success: function(response) {
                    try {
                        let data = JSON.parse(response);
                        if (data.success) {
                            $(`#${fileId}-status`).text('Uploaded');
                            $(`#${fileId}`).html(`
                                <td>${data.file_name}</td>
                                <td>${data.file_type}</td>
                                <td>${formatFileSize(data.file_size)}</td>
                                <td>
                                    <a href="uploads/${data.file_name}" class="btn btn-download"
download>Download</a>
                                    <button class="btn btn-delete" onclick="deleteFile('${data.f
ile_name}', '${fileId}')">Delete</button>
                                </td>
                            `);
                        } else {
                            $(`#${fileId}-status`).text('Error: ' + data.message);
                        }
                    } catch (e) {
                        $(`#${fileId}-status`).text('Error processing response');
                    }
                },
                error: function(xhr, status, error) {
                    $(`#${fileId}-status`).text('Upload failed: ' + error);
                }
            });
        }

        // Load existing files on page load
        loadExistingFiles();

        function loadExistingFiles() {
            $.ajax({
                url: 'get_files.php',
                type: 'GET',
                success: function(response) {
                    try {
                        let files = JSON.parse(response);
                        if (files.length > 0) {
                            files.forEach(function(file) {
                                let fileId = 'file-' + Date.now() + '-' + Math.random().toString
(36).substr(2, 9);

                                let row = document.createElement('tr');
                                row.id = fileId;
                                row.innerHTML = `
                                    <td>${file.name}</td>
                                    <td>${file.type}</td>
                                    <td>${formatFileSize(file.size)}</td>
                                    <td>
                                        <a href="uploads/${file.name}" class="btn btn-download"
```

```
download>Download</a>
                                        <button class="btn btn-delete" onclick="deleteFile('${fi
le.name}', '${fileId}')">Delete</button>
                                    </td>
                                `;
                                document.getElementById('fileList').appendChild(row);
                            });
                        }
                    } catch (e) {
                        console.error('Error loading existing files:', e);
                    }
                },
                error: function(xhr, status, error) {
                    console.error('Error loading existing files:', error);
                }
            });
        }

        function formatFileSize(bytes) {
            if (bytes === 0) return '0 Bytes';
            const k = 1024;
            const sizes = ['Bytes', 'KB', 'MB', 'GB'];
            const i = Math.floor(Math.log(bytes) / Math.log(k));
            return parseFloat((bytes / Math.pow(k, i)).toFixed(2) + ' ' + sizes[i];
        }
    });

    function deleteFile(fileName, rowId) {
        if (!confirm('Are you sure you want to delete this file?')) {
            return;
        }

        $.ajax({
            url: 'delete_file.php',
            type: 'POST',
            data: { file_name: fileName },
            success: function(response) {
                try {
                    let data = JSON.parse(response);
                    if (data.success) {
                        $('#' + rowId).remove();
                    } else {
                        alert('Error: ' + data.message);
                    }
                } catch (e) {
                    alert('Error processing response');
                }
            },
            error: function(xhr, status, error) {
                alert('Delete failed: ' + error);
            }
        });
    }
    </script>
</body>
</html>
```

PHP (upload.php)

php Copy Download

```php
<?php
header('Content-Type: application/json');

// Create uploads directory if it doesn't exist
if (!file_exists('uploads')) {
    mkdir('uploads', 0777, true);
}
```

```php
$allowedTypes = [
    'image/jpeg' => 'jpg',
    'image/png' => 'png',
    'image/gif' => 'gif',
    'application/pdf' => 'pdf'
];

$response = [
    'success' => false,
    'message' => '',
    'file_name' => '',
    'file_type' => '',
    'file_size' => 0
];

try {
    if (!isset($_FILES['file'])) {
        throw new Exception('No file uploaded.');
    }

    $file = $_FILES['file'];

    // Check for upload errors
    if ($file['error'] !== UPLOAD_ERR_OK) {
        throw new Exception('File upload error: ' . $file['error']);
    }

    // Check file type
    $fileType = mime_content_type($file['tmp_name']);
    if (!array_key_exists($fileType, $allowedTypes)) {
        throw new Exception('Invalid file type. Only images (JPEG, PNG, GIF) and PDF files are a
llowed.');
    }

    // Check file size (max 5MB)
    if ($file['size'] > 5 * 1024 * 1024) {
        throw new Exception('File size exceeds 5MB limit.');
    }

    // Generate a unique filename while preserving the extension
    $extension = $allowedTypes[$fileType];
    $filename = uniqid() . '.' . $extension;

    // Move the file to the uploads directory
    if (!move_uploaded_file($file['tmp_name'], 'uploads/' . $filename)) {
        throw new Exception('Failed to move uploaded file.');
    }

    $response['success'] = true;
    $response['message'] = 'File uploaded successfully.';
    $response['file_name'] = $filename;
    $response['file_type'] = $fileType;
    $response['file_size'] = $file['size'];

} catch (Exception $e) {
    $response['message'] = $e->getMessage();
}

echo json_encode($response);
?>
```

PHP (get_files.php)

php Copy Download

```php
<?php
header('Content-Type: application/json');
```

```php
$files = [];

if (file_exists('uploads') && is_dir('uploads')) {
    $dir = new DirectoryIterator('uploads');

    foreach ($dir as $fileinfo) {
        if (!$fileinfo->isDot() && $fileinfo->isFile()) {
            $filePath = $fileinfo->getPathname();
            $mimeType = mime_content_type($filePath);

            // Only include images and PDFs
            if (strpos($mimeType, 'image/') === 0 || $mimeType === 'application/pdf') {
                $files[] = [
                    'name' => $fileinfo->getFilename(),
                    'type' => $mimeType,
                    'size' => $fileinfo->getSize()
                ];
            }
        }
    }
}

echo json_encode($files);
?>
```

PHP (delete_file.php)

php Copy Download

```php
<?php
header('Content-Type: application/json');

$response = [
    'success' => false,
    'message' => ''
];

try {
    if (!isset($_POST['file_name']) || empty($_POST['file_name'])) {
        throw new Exception('No file name provided.');
    }

    $fileName = basename($_POST['file_name']);
    $filePath = 'uploads/' . $fileName;

    if (!file_exists($filePath)) {
        throw new Exception('File does not exist.');
    }

    // Verify the file is actually in our uploads directory
    if (realpath(dirname($filePath)) !== realpath('uploads')) {
        throw new Exception('Invalid file path.');
    }

    if (!unlink($filePath)) {
        throw new Exception('Failed to delete file.');
    }

    $response['success'] = true;
    $response['message'] = 'File deleted successfully.';

} catch (Exception $e) {
    $response['message'] = $e->getMessage();
}

echo json_encode($response);
?>
```

Implementation Notes

1. **Directory Structure**:

```text
/project-root/
├── index.html
├── upload.php
├── get_files.php
├── delete_file.php
└── uploads/ (this directory should be created automatically)
```

2. **Security Considerations**:
   - The uploads directory should have proper permissions (usually 755)
   - Consider adding .htaccess to prevent direct PHP execution in uploads directory
   - For production, add more security checks (virus scanning, user authentication, etc.)
3. **Features**:
   - Drag and drop file upload
   - File type validation (images and PDFs only)
   - File size limit (5MB)
   - Progress bar for uploads
   - List of uploaded files with download and delete options
   - Automatic loading of existing files on page load
4. **Extensions**:
   - Add user authentication
   - Implement file categorization
   - Add thumbnail generation for images
   - Implement file sharing functionality

This solution provides a complete, secure, and user-friendly file upload system with all the requested features.

you dont need to json.parse. returned text is already JSON formatted. it is double parsing so, getting error.

You're absolutely right! When using jQuery's AJAX with `dataType: 'json'` or when the server returns a proper `Content-Type: application/json` header, jQuery automatically parses the response into a JavaScript object. Doing `JSON.parse()` on an already parsed object causes errors.

Here's the corrected version of the JavaScript code (just the relevant parts that need changing):

Corrected JavaScript Code

```javascript
// In the uploadFile function's AJAX success handler:
success: function(data) {  // data is already parsed as JSON
    if (data.success) {
        $(`#${fileId}-status`).text('Uploaded');
        $(`#${fileId}`).html(`
            <td>${data.file_name}</td>
```

```
            <td>${data.file_type}</td>
            <td>${formatFileSize(data.file_size)}</td>
            <td>
                <a href="uploads/${data.file_name}" class="btn btn-download" download>Download</a>
                <button class="btn btn-delete" onclick="deleteFile('${data.file_name}', '${fileId}')">Delete</button>
            </td>
        `);
    } else {
        $(`#${fileId}-status`).text('Error: ' + data.message);
    }
},

// In the loadExistingFiles function's AJAX success handler:
success: function(files) {  // files is already parsed as JSON
    if (files.length > 0) {
        files.forEach(function(file) {
            let fileId = 'file-' + Date.now() + '-' + Math.random().toString(36).substr(2, 9);
            let row = document.createElement('tr');
            row.id = fileId;
            row.innerHTML = `
                <td>${file.name}</td>
                <td>${file.type}</td>
                <td>${formatFileSize(file.size)}</td>
                <td>
                    <a href="uploads/${file.name}" class="btn btn-download" download>Download</a>
                    <button class="btn btn-delete" onclick="deleteFile('${file.name}', '${fileId}')">Delete</button>
                </td>
            `;
            document.getElementById('fileList').appendChild(row);
        });
    }
},

// In the deleteFile function's AJAX success handler:
success: function(data) {  // data is already parsed as JSON
    if (data.success) {
        $('#' + rowId).remove();
    } else {
        alert('Error: ' + data.message);
    }
},
```

Full Corrected AJAX Calls

Here are the complete corrected AJAX calls with the proper configuration:

1. **File Upload**:

javascript Copy Download

```
$.ajax({
    url: url,
    type: 'POST',
    data: formData,
    processData: false,
    contentType: false,
    dataType: 'json', // Expect JSON response
    xhr: function() {
        let xhr = new XMLHttpRequest();
        xhr.upload.addEventListener('progress', function(e) {
            if (e.lengthComputable) {
                let percentComplete = Math.round((e.loaded / e.total) * 100);
                $(`#${fileId}-progress`).width(percentComplete + '%');
                $(`#${fileId}-progress`).text(percentComplete + '%');
            }
```

```
        }, false);
        return xhr;
    },
    success: function(data) {
        if (data.success) {
            $(`#${fileId}-status`).text('Uploaded');
            $(`#${fileId}`).html(`
                <td>${data.file_name}</td>
                <td>${data.file_type}</td>
                <td>${formatFileSize(data.file_size)}</td>
                <td>
                    <a href="uploads/${data.file_name}" class="btn btn-download" download>Downlo
ad</a>
                    <button class="btn btn-delete" onclick="deleteFile('${data.file_name}', '${f
ileId}')">Delete</button>
                </td>
            `);
        } else {
            $(`#${fileId}-status`).text('Error: ' + data.message);
        }
    },
    error: function(xhr, status, error) {
        $(`#${fileId}-status`).text('Upload failed: ' + error);
    }
});
```

2. **Load Existing Files:**

javascript Copy Download

```
$.ajax({
    url: 'get_files.php',
    type: 'GET',
    dataType: 'json', // Expect JSON response
    success: function(files) {
        if (files.length > 0) {
            files.forEach(function(file) {
                let fileId = 'file-' + Date.now() + '-' + Math.random().toString(36).substr(2,
9);
                let row = document.createElement('tr');
                row.id = fileId;
                row.innerHTML = `
                    <td>${file.name}</td>
                    <td>${file.type}</td>
                    <td>${formatFileSize(file.size)}</td>
                    <td>
                        <a href="uploads/${file.name}" class="btn btn-download" download>Downloa
d</a>
                        <button class="btn btn-delete" onclick="deleteFile('${file.name}', '${fi
leId}')">Delete</button>
                    </td>
                `;
                document.getElementById('fileList').appendChild(row);
            });
        }
    },
    error: function(xhr, status, error) {
        console.error('Error loading existing files:', error);
    }
});
```

3. **Delete File:**

javascript Copy Download

```
$.ajax({
    url: 'delete_file.php',
    type: 'POST',
```

```
    data: { file_name: fileName },
    dataType: 'json', // Expect JSON response
    success: function(data) {
        if (data.success) {
            $('#' + rowId).remove();
        } else {
            alert('Error: ' + data.message);
        }
    },
    error: function(xhr, status, error) {
        alert('Delete failed: ' + error);
    }
});
```

Key Changes Made:

1. Removed all `JSON.parse()` calls since jQuery automatically parses the response when:
   - The server sends the correct `Content-Type: application/json` header (which your PHP scripts already do)
   - Or when you specify `dataType: 'json'` in the AJAX settings
2. Added `dataType: 'json'` to all AJAX calls to explicitly tell jQuery to expect and parse JSON responses
3. The success callback functions now directly work with the parsed JavaScript object (`data` or `files`) instead of trying to parse it again

These changes will prevent the "double parsing" error while maintaining all the functionality of your file upload system.

New chat

Message DeepSeek

DeepThink (R1) Search

AI-generated, for reference only