Upload to server of xml document via http/post using chrome postman

# Required functionality:

The agenda of the task is to upload an xml document from a client to the webserver via http/post request. The xml document to be uploaded is as follows:



The xml file shall be uploaded by means of an http POST request using the **chrome postman**, whereas the used URL (in first approach it can be localhost, later a URL including domain name or IP to real server station) directs the request to the webserver accessible from any client in the internet.

The webServer shall store the uploaded file in a specific directory on the web server. Additionally the file name of the uploaded file is to be adapted:

* The filename shall start with the timestamp (time of system in ms) indicating when the http/post request has been received on the web Server
* After the timestamp the sessionID is part of the filename indicating the precise session with which the client can be reached
* The original filename is then appended to the resulting filename

Examplefilename:   
“2015-07-14\_12h45m32s850ms\_sid=edb0e8665db4e9042fe0176a89aade16\_i\_2\_trip\_request.xml”

# Required infrastructure/components/modules: To run this task, we need a local server (later we can shift to external sever), Chrome Postman, Editor, Browser, JavaScript, jQuery, Ajax, PHP

* Server: Xammp (Further detailed in definitions)
* Chrome Postman is a plugin used to send request to the server. Here it used to send every file to server (specified directory) using http post request.
* Editor: We used Notepad++ for coding and editing
* Browser: Google chrome
* JavaScript: As client scripting language.
* jQuery: It sub part of JavaScript(Further detailed in definitions)
* Ajax: For http post request(Further detailed in definitions)
* PHP: Server side scripting language(Further detailed in definitions)

# Approach: To approach this task, various references have been used for instance PHP documentation, jQuery tutorials have been referred. To spot the errors and debug them, chrome development tools have been used. Codes were tested at every phase of the development and simplify the results. And document the results at every phase for future reference.

**In brief**

•We need to run a local server in the machine (Apache web server) and make sure it is working properly, to make it sure by opening the admin page of xampp in URL.

•Then configure the files(our files index.html, script.js,jquery.min.js,upload.php) in Apache server by saving our work space in “htdocs” of XAMPP server and even create file to save the desired files in server.

•We have to save the our files in htdocs(our files are index.html,script.js,jquery.min.js and upload.php)these files needed to save with proper extension and edit them in notepad.

•In case of errors, enable chrome development tool in chrome menu bar, it provides the efficient way to debug and reduces redundancy if we are using chrome as web browser if other browser IE or Firefox you need to enable in that browser.

•After all this, open localhost in browser then server automatically opens your “htdocs” folder in the xampp server, open the specified project (in our case it is uploadtoserver).

# Definitions:

XAMPP:

It is an open source web server, which stand for X cross origin (any platform), A for Apache HTTP server, M for MySQL, P and P for PHP and Perl but we only start apache server for our use

JQuery:

jQuery is library of JavaScript to make rich functionality of webpages and it is too open source and it is normally referred in HTML as JavaScript and it provides readymade library and these can be directly downloaded, however google and Microsoft hosted their own libraries and can be used too, we have used library just enrich the selection bar after the result from server, which adds colors and fonts readily for selection bar

Ajax:

It is a web technology, Ajax Stands for Asynchronous JavaScript XML.And it is not a new programming language but name given for these set of tools (JavaScript, xml and state of Asynchronous). The "asynchronous" part of Ajax refers to the way requests are made to the Web server.

PHP:

It stands for Hypertext Preprocessor and used at server side as a scripting language, more over it interacts with database but her we used for providing where to store the files in the server and type of files to be saved in the server after server response.

# Concrete Solution of process flow with code is been briefed

Our files used are

• index.html

• script.js

• jquery.min.js

• upload.php

index.html: It is an html file that calls script.js file and jquery.min.js file and also creates an html form (a box to save files in server)

**index.html**

**<!DOCTYPE html>**

**<html lang="en" >**

**<body>**

**<form>**

**<div class="row">**/\* the <div> tag encapsulates other page elements and divides the HTML document into sections and it equally distributes all css effects especially the colors, alignment, padding\*/

**<div class="column-small-12 padd0 align-center">**/\*here makes an alignment \*/

**<div id="drop-box">/\***it creates an box\*/

<p>Select Files</p>/\*in box it appears as select files, and the selected files are saved in server\*/

**</div>**

**</div>**

**<div class="column-small-12 padd0">**

<! -- As soon as browse the file controller calls the script.js file in that Upload.php method -->

**<input type="file" name="upload" id="upload" />**/\*for the form we type of input and name and id\*/

**</div>**

**</div>**

**</form>**

**</body>**

**</html>**

**script.js**: It is an JavaScript file with jQuery for reducing the redundancy of the processes and make it straight forward

**$(function(){**

**$("#drop-box").click(function(){**

**$("#upload").click();**

**});**

/\*the function creates an drop box for selecting files on click of the event \*/

**$(document).on('drop dragover', function {**

**preventDefault();**

**});**

/\*to prevent browsers from opening the file when it is dragged and dropped on to the page \*/

**$('input[type=file]').on('change', fileUpload);**

/\*Adds the event and changed when the file is uploaded \*/

**function fileUpload(event){**

**$("#dropbox").html("<p>"+event.target.value+" uploading...</p>");**

**files = event.target.files;**

/\* when event is called it checks for the target and uploads in specified path \*/

**var data = new FormData();**

/\*every time the form of html considers as new form and if the error condition is zero the file will be uploaded if not assigns as error 1 and asks for some amendment \*/

**var error = 0;** /\*then if state is zero the cycle repeats of next file to be uploaded \*/

**for (var i = 0; i < files.length; i++) {**

**var file = files[i];**

**console.log(file.size);**

**if(!file.type.match('image.\*')) {**

**$("#drop-box").html("<p> Images only. Select another file</p>");**

/\*These are the cases were we can the error with files and describes the state of error as 1 in case of any other format of or size of image\*/

**error = 1;**

**}else if(file.size > 1048576){**

**$("#drop-box").html("<p> Too large Payload. Select another file</p>");**

**error = 1;**

**}else{**

**data.append('image', file, file.name);**

**}**

**}**

**if(!error){**

/\* next it sends request to server as xmlhttprequest using post method and addresses the url as localhost and uploadtoserver as folder to save in the server \*/

**var xhr = new XMLHttpRequest();**

**xhr.open('POST', 'http://localhost/uploadtoserver/upload.php', true);**

**xhr.send(data);**

**xhr.onload = function () {**

**if (xhr.status === 200) {**

/\*if status of the file is received (200) then it processes the request if and asks for more files if not else conditions as some error in upload \*/

**$("#drop-box").html("<p> File Uploaded. Select more files</p>");**

**} else {**

**$("#drop-box").html("<p> Error in upload, try again.</p>");**

**}}}}});**

**Upload.php (php file**): Runs at server end

**<?php**

**if(isset($\_POST) == true){** /\* isset determine if a variable is set and is not NULL\*/

**$errors = array();**

**session\_start();**/\*php function fo session , to start a session \*/

**$file\_name = $\_FILES['file']['name'];** /\* is the name of the file\*/

**$file\_tmp = $\_FILES['file']['tmp\_name'];**//temporary file name

**$t = time();**// time declaration

**$date = date("Y-m-d",$t**);// date declaration with time

**$b = time ();** // time declaration as a new variable

**date\_default\_timezone\_set('Europe/Berlin');**/\*Time zone syntax of php\*/

**$ts = explode(":", date("G:i:s:ms",$b));** /\*The explode() function breaks a string into an array\*/

**$times = $ts[0]."h".$ts[1]."m".$ts[2]."s".$ts[3]."ms";**// appendhours, minutes, second, millisecond as string in variable times

**$session\_id = session\_id();**// session is declared as variable

**$data=$date."\_".$times."\_sid\_=\_".$session\_id**;/\*date,time, session id are concatenated as a string \*/

**if(empty($errors)==true){** // if errors are empty or true

**move\_uploaded\_file($file\_tmp,"D:/server/htdocs/server\_files/".$data.$file\_name);**//file to upload to server

**}**

**}**

**?>**

The session id is declared using readily available session\_start function in php libraries. Declared to an variable as $session\_id = session\_id(); and used in time stamp.

# Summary: To sum up the whole task, the task was to save a file in server using http post request by Postman (chrome plugin). The saved file in the server has to be saved in a specified directory with a timestamp including session id on it.For this task execution scripting languages and local server have been used .The execution and explanation of the code is been described in above section