Patrick Roanhouse

CS 80 1678 - GEDDES JR J K (1-18)

Pages 603-604 of our textbook:

16.1

a.) asynchronous
b.) XMLHttpRequest
c.) callback
d.) innerHTML
e.) XHR
f.) JSON
g.) Cross-Site Scripting or XSS
h.) eval
i.) try block
j.) responseXML

16.2

a.) False. and Ajax app can use any text data as a response, commonly you will see JSON used
b.) True
c.) True
d.) True
e.) False, when the XMLHttpRequest object receives XML data, the XMLHttpRequest object parses and stores the data as a DOM object in the responseXML property.
f.) False, an exception is an indication of a problem that occurs during a program's execution.
g.) False, the third argument to XMLHttpRequest method 'open' must be TRUE to make an asynchronous request.

16.3 Describe the differences between client/server interactions in traditional web applications and client/server interactions in Ajax web

applications.

In a traditional client / server interaction in a traditional web application you are getting and sending a request once at a time. So it is single game of call and response ping pong for the request and then the sending data back with a full page load and reload for each usage of the application. In an Ajax applications the Ajax app separates client-side user interaction and server communication and run them in parallel in a separate layer, making the delays of server-side processing more transparent to the user. This then allows for only partial page loads with updated content from the interactions from the user allowing a far more responsive application.

If you want the exact book definition: on page 27-15 through 27-17 the explain the following:

Traditional Web application:

The user first fills in the form's fields, then submits the form (Fig. 27.9, Step 1). The browser generates a request to the server, which receives the request and processes it (Step 2). The server generates and sends a response containing the exact page that the browser will render (Step 3), which causes the browser to load the new page (Step 4) and temporarily makes the browser win- dow blank. The client waits for the server to respond and reloads the entire page with the data from the response (Step 4). While such a synchronous request is being processed on the server, the user cannot interact with the client web page. If the user interacts with and submits another form, the process begins again (Steps 5–8). This model was originally designed for a web of hypertext documents, what some people call the "brochure web." As the web evolved into a full-scale applications platform

Ajax Web Applications

Ajax applications add a layer between the client and the server to manage communication between the two (Fig. 27.10). When the user interacts with the page, the client creates an XMLHttpRequest object to manage a request (Step 1). This object sends the request to the server (Step 2) and awaits the response. The requests are asynchronous, so the user can continue interacting with the application on the client side while the server processes the ear- lier request concurrently. Other user interactions could

result in additional requests to the server (Steps 3 and 4). Once the server responds to the original request (Step 5), the XMLHttpRequest object that issued the request calls a client-side function to process the data returned by the server. This function—known as a callback function—uses partial page updates (Step 6) to display the data in the existing web page without reloading the entire page. At the same time, the server may be responding to the second request (Step 7) and the client side may be starting to do another partial page update (Step 8). The callback function updates only a designated part of the page. Such partial page updates help make web applications more responsive, making them feel more like desktop applications. The web application does not load a new page while the user interacts with it.

16.5 Describe each of the following terms in the context of Ajax:
a) type-ahead - As the user enters each keystroke, the application asynchronously calls the server to obtain the list of names in which the last name starts with the characters the user has entered so far

b) edit-in-place - used for quickly amending text on screen. A simple click loads the text into an edit box, right there on the page.

c) partial page update - updates only a designated part of the page help make web applications more responsive, making them feel more like desktop applications. The web application does not load a new page while the user interacts with it.

d) asynchronous request - If you use XMLHttpRequest from an extension, you should use it asynchronously. In this case, you receive a callback when the data has been received, which lets the browser continue to work as normal while your request is being handled

e) XMLHttpRequest - The XMLHttpRequest object (which resides on the client) is the layer between the client and the server that manages

asynchronous requests in Ajax applications. This object is supported on most browsers, though they may implement it differently.

f) "raw" Ajax - uses JavaScript to send asynchronous requests to the server, then updates the page using the DOM. "Raw" Ajax is best suited for creating small Ajax components that asynchronously update a section of the page

g) callback function - the XMLHttpRequest object that issued the request calls a client-side function to process the data returned by the server. It uses partial page updates to display the data in the existing web page without reloading the entire page.

h) same origin policy - For security purposes, the XMLHttpRequest object doesn't allow a web application to request resources from domains other than the one that served the application. For this reason, the web application and its resources must reside on the same web server to prevent Cross Site Scripting (XSS)

i) Ajax libraries - pre-written Javascript code with defined AJAX features / functions developers use to speed up development by handling common reoccurring problems such as browser incompatibilities.

j) RIA - Rich Internet Applications (RIAs) are web applications that approximate the look, feel and usability of desktop applications. Two key attributes of RIAs are performance and a rich GUI.