

AJAX

Ajax is not a programming language or a tool, but a concept. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. The best for Ajax is “the method of exchanging data with a server, and updating parts of a web page – without reloading the entire page.” Ajax itself is mostly a generic term for various JavaScript techniques used to connect to a web server dynamically without necessarily loading multiple pages. In a more narrowly-defined sense, it refers to the use of XMLHttpRequest objects to interact with a web server dynamically via JavaScript.

Benefits of Ajax

There are 4 main benefits of using Ajax in web applications:

1. **Callbacks:** *Ajax is used to perform a callback, making a quick round trip to and from the server to retrieve and/or save data without posting the entire page back to the server. By not performing a full postback and sending all form data to the server, network utilization is minimized and quicker operations occur. In sites and locations with restricted bandwidth, this can greatly improve network performance. Most of the time, the data being sent to and from the server is minimal. By using callbacks, the server is not required to process all form elements. By sending only the necessary data, there is limited processing on the server. There is no need to process all form elements, process the ViewState, send images back to the client, or send a full page back to the client.*
2. **Making Asynchronous Calls:** *Ajax allows you to make asynchronous calls to a web server. This allows the client browser to avoid waiting for all data to arrive before allowing the user to act once more.*
3. **User-Friendly:** *Because a page postback is being eliminated, Ajax enabled applications will always be more responsive, faster and more user-friendly.*
4. **Increased Speed:** *The main purpose of Ajax is to improve the speed, performance and usability of a web application.*

Technical Aspects of Ajax

Ajax callbacks can be done by instantiating an XMLHttpRequest object in the client-side JavaScript. The XMLHttpRequest object can be used to directly call server-side objects like pages and web services. These pages and web services will either save and/or return data.

Ajax was originally an acronym for Asynchronous JavaScript and XML. “Asynchronous” means that multiple events are happening independently of one another. Once a client initializes an Ajax callback to the server, the client will not need to wait for a response and can continue to use the web application while the request is being processed. Once done, the server will send a response back to the client and the client will process it as necessary.

What Advances have Been Made to Ajax?

JavaScript is the client-side programming language and XML is a markup language to define data. JSON is another markup language to define data. JSON (JavaScript Object Notation) is much easier to use with JavaScript than XML. When it comes to Ajax and JavaScript, JSON Web Services are replacing XML Web Services.

Another major advance to JavaScript and Ajax is the JavaScript object library called jQuery. This free, open-source software is a wrapper around JavaScript. jQuery is used to easily write client-side JavaScript to navigate and manipulate a page and make asynchronous Ajax callbacks.

By using jQuery and JSON Web Services, Ajax callbacks have become standard programming practices for designing and developing web applications.

The Ajax Control Toolkit is a suite of controls created by Microsoft that is integrated into Visual Studio and can be dragged and dropped onto web forms just like html and server controls. These controls are intended to be used for Ajax callbacks. However, they can also be used as normal client and or server controls. For example, Asp.Net does not come with the Tabs controls. However, the Ajax Control Toolkit does. The Tab control can postback to the server just like server controls.

Where Should Ajax be Used?

Ajax should be used anywhere in a web application where small amounts of information could be saved or retrieved from the server without posting back the entire pages. A good example of this is data validation on save actions. Another example would be to change the values in a drop down list-box based on other inputs, such as state and college list boxes. When the user selects a state, the college list box will repopulate with only colleges and universities in that state.

Other features include text hints and autocomplete text boxes. The client types in a couple of letters and a list of all values that start with those letters appear below. A callback is made to a web service that will retrieve all values that begin with these

characters. This is a fantastic feature that would be impossible without Ajax and is also part of the Ajax Control Toolkit.

To solve this problem, we created JSON Web Services on the web server in order to retrieve the details about the selected item. The JSON web service would retrieve the data and convert into JSON and return a JSON string. Instead of posting back to the server, the client would call the web service when an item was selected from the list box. We used jQuery to make an asynchronous Ajax call to the web service. Once the client retrieved the data back from the web service, more client side processing was done to display the information on the page. The time it took to display the details on the page after the item was selected was instantaneous. There was no page flicker, refresh or postback involved.

Code Snippet for jQuery ajax:-

```
$.ajax({  
    url: URLName, //The service url you want to hit  
    type: "POST", //Http method either GET,POST or other methods  
    async: false, // for asynchronous request , you can put either true or false  
    data: {  
        'service': serviceName, // data is the payload obj you want to send to controller  
        'chartPreference': chartPreference,  
    },  
    success: function(response) { //success function is executed if the response code is 200  
        response = JSON.parse(response); //means the request processed and success  
    },  
    error: function(response) { // if any thing went wrong this error method will execute  
        alert("Failed To Connect Service");  
    }  
});
```

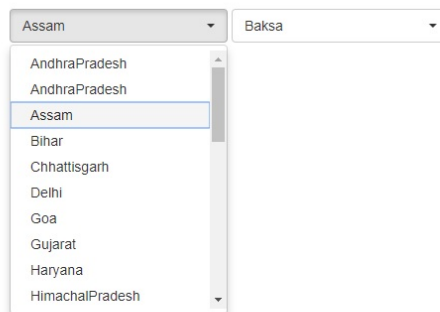
Sample AJAX with servlet;-

Functionality: -Based on you select a state it will hit an ajax and show the district or cities respectively

Simple code: - <https://github.com/Rajani-Igit/AjaxMultipleselectOption.git>

After downloading the project import in to eclipse or ur favorite id, Do clean the project and deploy in tomcat,

Screenshots attached: -



Odisha ▾

Angul ▾

Angul

Balangir

Balasore

Bargarh

Bhadrak

Boudh

Cuttack

Deogarh

Dhenkanal

Gajapati