**JQuery AJAX**

JQuery AJAX is a set of methods in the jQuery JavaScript library that allow you to make asynchronous HTTP requests to a server. This means that you can send and receive data from the server without having to reload the entire web page.

AJAX stands for Asynchronous JavaScript and XML. It is a technique that allows web pages to update their content without having to reload the entire page. This makes web pages more responsive and interactive.

See the code below…

|  |
| --- |
| $.ajax({  url: 'https://example.com/api/data',  method: 'GET',  dataType: 'json',  success: function(response) {  // Handle the response from the server  console.log(response);  },  error: function(xhr, status, error) {  // Handle any errors that occurred during the request  console.error(error);  }  }); |

**preventDefault() function**

The preventDefault() method in jQuery is used to stop the default action of an element from happening. For example, you can use it to prevent a submit button from submitting a form, or to prevent a link from following the URL.

The preventDefault() method does not prevent further propagation of an event through the DOM. To handle this, you can use the stopPropagation() method.

|  |
| --- |
| <script>  $(document).ready(function(){  let n=0;  $('#submit').click(function(e){  e.preventDefault();  console.log("button clicked!"+n);  n++;  });  });  </script> |

Real life usage in form: generally using this function we CAN PREVENT FORM RELOAD. WHENEVER CLIENT HIT THE SUBMIT TYPE BUTTON IN HTML FORM THE PAGE GET RELOAD FOR ATLEAST ONE TIME. USING THIS FUNCTION WE CAN EASILY PREVENT PAGE RELOAD.

**Getting form data into array object into jQuery**

Now we generally get form’s input value and then store into specific variable. After then we initialize an array object with those variables. We initialize those as ***key:value***

**Sends HTML FORM DATA VIA AJAX to PHP File [INSERT/UPDATE/DELETE]**

Let’s see the jQuery code part

|  |
| --- |
| $(document).ready(function(){  $('#submit').click(function(e){  e.preventDefault();  let nm = $('#uname').val();  let ph = $('#phone').val();    console.log(nm+' '+ph);    // initialize object's array to ajax  $.ajax({  url:"insert.php",  method:"POST",  data:{  name:nm,  phone:ph  },  success:function(data){  console.log(data);  }  });  });  }); |

Let’s see ‘insert.php’ file….

|  |
| --- |
| <?php  include 'conn.php';  $fname = $\_POST['name'];  $Phone = $\_POST['phone'];  if(!empty($fname) && !empty($Phone)){  $sql = "INSERT INTO customers(fullname,phone) VALUES('$fname','$Phone')";    if(mysqli\_query($conn,$sql)){  echo "record saved!";  }else{  echo "record insert failed!";  }  }else{    }  ?> |

**Note:**

|  |
| --- |
| Note: INSERT,UPDATE,DELETE => POST method and SELECT/SEARCH => GET method  learn php with ajax:  <https://www.studentstutorial.com/ajax/insert-data?expand_article=1>  live search:php+ajax+mysql: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-ajax-live-search.php> |

**Note 2:**

When you do live search functionality with Laravel then,

1. Make proper routes for them
2. Include jquery file inside your blade template
3. Add csrf\_token() function value as metatag ***<meta name="csrf-token" content="{{ csrf\_token() }}" />***
4. In AJAX code in blade file, url type will like ***$.get("{{url('/')}}/search"***
5. Pass the parameters with csrf token ***{\_token: $('meta[name="csrf-token"]').attr('content'),term: inputVal})***
6. For getting live search code help[php+ajax+mysql]: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-ajax-live-search.php>