AJAX - Monitoring Progress

AJAX provides a special feature named Monitoring Progress. Using this feature we can able to track the progress of the asynchronous request made by the AJAX from the web browser to the web server. Or we can say that using a progress monitor we can also monitor how much amount of data is uploaded or downloaded from the server to the user. With the help of monitoring progress, we can send feedback to the user which contains the following points −

Data Transfer Progress − We can monitor the progress of the data transferred from the server to the client. Or we can also track how much data is transferred or received as compared to the total size of the given file.

Request Status − Wan can also monitor the overall status (like whether the request is still in progress or it is complete or pending) of the request we made. It helps the programmer to provide proper feedback to the user of the current request.

Error Handling − Apart from tracking the current status it is also important to handle if any error occurs while requesting data like server-side errors, network issues, etc. So using error handling we can easily send a notification to the user so that he/she can take proper action to the occurred error.

How to Monitor Progress

To monitor the progress of the AJAX request we can use the following methods −

Using onprogress event − To monitor the progress of the request we can define an "onprogress" event which triggers periodically while the data transfer is processed. It is generally used to monitor the progress of file downloads or large data/file transfers. It monitors the progress of the information like how much data is loaded, the total size of the transferred data, etc.

Example

In the following program, we will monitor the current status of the request with the help of the onprogress event. Here we create a Javascript function named as displayStatus() which shows the states of how much data is being transferred. This function makes an AJAX request to send data to the given URL. So it uses an XMLHttpRequest object to create a request and then defines a callback function to handle the response provided by the server. Inside the callback function. The onprogress event checks the current progress of the transferred data. Inside the onprogress event handler, we can check if the pr s data is computable to avoid dividing zero errors. If it is computable, then we can calculate the percentage of data transferred to the server.

```
<script>
function displayStatus() {
   // Creating XMLHttpRequest object
   var myObj = new XMLHttpRequest();

   // Creating call back function
   // Here onprogress return the percentage of transferred data
   myObj.onprogress = function(myEvent) {
      if (myEvent.lengthComputable){
         var dataTarnsferPercentage = (myEvent.loaded/myEvent.total)*100;
         console.log("Current progress of the data transfer:", dataTarnsferPercent
      }
   };
   // Open the given file
   myObj.open("GET", "https://jsonplaceholder.typicode.com/todos", true);

   // Sending the request to the server
   myObj.send();
}
</script>
```

Using onreadystatechange event − We can monitor the progress of the request by creating an onreadystatechnage event. This event triggers whenever the readyState property of the XMLHttpRequest changes. The readyState property returns the current state of the request.

Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Example

In the following program, we will monitor the current status of the request with the help of the onreadystatechange event. Here we create a Javascript function named as displayStatus() which shows the states of the current status of the request. This function makes an AJAX request to retrieve data from the given URL. So it uses an XMLHttpRequest object to create a request and then defines a callback function to handle the response provided by the server. Inside the callback function. The onreadystatechange event checks the current status of the request with the help of the readyState property. If the readyState is XMLHttpRequest.DONE, that means the request is completed and print "Request is completed". Otherwise print "Request is in-progress".

```
<script>
function displayStatus() {
   // Creating XMLHttpRequest object
   var myObj = new XMLHttpRequest();

   // Creating call back function
   // Here onreadystatechange return the current state of the resuest
   myObj.onreadystatechange = function() {
      if (this.readyState == XMLHttpRequest.DONE){
         console.log("Request is completed")
      }else{
         console.log("Request is in-progress")
      }
   };
   // Open the given file
   myObj.open("GET", "https://jsonplaceholder.typicode.com/todos", true);

   // Sending the request to the server
   myObj.send();
}
</script>
```

Conclusion

So this is how we can monitor the progress of the requests. So that we can easily track down how much data is being transferred, how much data is handled successfully, errors, etc. Now in the next article, we will see the status codes supported by the AJAX.