Progress Bar

Forms

* To upload a file we use form with method ="post" and the enctype="multipart/form-data". This encrypt type allows to upload files correctly.

<form action="/upload.php" method="POST" enctype="multipart/form-data" id="upload">

<div class="form-group">

<label for="fileupload">File : </label>

**<input type="file" id="fileupload" name="file" class="form-control-file">**

</div>

<button class="btn btn-danger">submit</button>

</form>

* Add an eventListener on submit. PreventDefault stops the form to submit on the action mentioned above.

form.addEventListener('submit' , function(event) {

event.preventDefault();

}

* To get the file

let file = this.querySelector(`#fileupload`).**files[0]**

FormData Class

* is used to encrypt the data

let formData = New **FormData**()

formData.**append**('name', 'Derry') // sends the data name = Derry in formData encryption

formdata.append('file', file);// put the file in a property called file

let ajax = new XMLHttpRequest();

ajax.addEventListener('load' , completeHandler)

ajax.open('POST' , '/upload.php'); // the route

ajax.send(formdata)

How to get upload progress?

* The XMLhttpRequest has a property called upload, and on that you can set a listener on progressEvent.
* Event in progressEvent has two values loaded and total.
  + Loaded is the part loaded that activates the progress event
  + event.total is the total size

ajax.**upload**.addEventListener("progress", progressHandler)

function progressHandler(event) {

let percent = Math.**round**(( event.**loaded** / event.**total** ) \* 100);

progressBar.style.width = `${percent}%`

progressBar.innerHTML = `${percent}%`

}

* You can use progressBar style from Bootstrap

Using axios package

* Axios has a method post, that can send data with post method. Axios returns a promise.
* You can install axios by npm, or by their CDN.
* Axios.post(send to what address, data, {onUploadProgress: function}) // the third parameter is options you can set.

**axios**.**post('/upload.php', formdata, {onUploadProgress: progressHander})**

**.then(response => console.log(response))**

**.catch(error => console.log(error))**