# Audio Recording

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Audio Test</title>

<script>

</script>

</head>

<body>

<a id=*"download"*>Download</a>

<button id=*"stop"*>Stop</button>

<br />

<div id=*"status"*>????</div>

</body>

<script>

let shouldStop = **false**;

let stopped = **false**;

**const** downloadLink = document.getElementById('download');

**const** stopButton = document.getElementById('stop');

**const** statusText = document.getElementById('status');

statusText.innerHTML = "waiting....";

console.log (document.featurePolicy.allowedFeatures());

stopButton.addEventListener('click', **function**() {

shouldStop = **true**;

statusText.innerHTML = "clicked";

});

// MediaDevice successfully retrieved

**const** handleSuccess = **function**(stream) {

console.log("media device success...");

//Init the Media Recorder

**const** options = {

mimeType : 'audio/webm'

};

**const** recordedChunks = [];

**const** mediaRecorder = **new** MediaRecorder(stream, options);

mediaRecorder.start(1000); // maximale sample time

console.log("media recorder init/start done ...");

// Event handler

mediaRecorder.ondataavailable = **function**(e) {

console.log("Handling on data available");

**if** (e.data.size > 0) {

recordedChunks.push(e.data);

console.log(e.data);

}

**if** (shouldStop === **true** && stopped === **false**) {

mediaRecorder.stop();

stopped = **true**;

statusText.innerHTML = "clicked... and ... stopped";

}

};

mediaRecorder.onerror = **function**(e) {

log('Error: ' + e);

console.log('Error: ', e);

};

mediaRecorder.onstart = **function**() {

console.log('Started, state = ' + mediaRecorder.state);

};

mediaRecorder.onstop = **function**(e) {

downloadLink.href = URL

.createObjectURL(**new** Blob(recordedChunks));

downloadLink.download = 'acetest.wav';

console.log('media recorder stopped');

statusText.innerHTML = "stopped";

};

console.log ("media recorder started...");

};

**var** constraints = { video: **false**, audio: **true** };

navigator.mediaDevices.getUserMedia(constraints)

.then(handleSuccess)

.**catch**(e => console.error(e));

</script>

</html>

# DataBlob Upload

<!DOCTYPE html>

<html lang=*"de"*>

<head>

<script src=*"http://code.jquery.com/jquery-1.11.0.min.js"*></script>

<title>JS - File Upload-1</title>

<script type=*"text/javascript"*>

<!--

//-->

**function** uploadSOund () {

**var** fd = **new** FormData();

fd.append('name', 'test.wav');

fd.append('data', soundBlob);

$.ajax(

{

type: 'POST',

url: '/upload.php',

data: fd,

processData: **false**,

contentType: **false**

}

).done(**function**(data) {

console.log(data);

}

);

};

// javascript function that uploads a blob to upload.php

**function** uploadBlob(){

// create a blob here for testing

**var** blob = **new** Blob(["i am a blob üüüüüääääääää"]);

console.log("create filereader...");

//var blob = yourAudioBlobCapturedFromWebAudioAPI;// for example

**var** reader = **new** FileReader();

// this function is triggered once a call to readAsDataURL returns

reader.onload = **function**(event){

console.log("this function is triggered once a call to readAsDataURL returns");

**var** fd = **new** FormData();

fd.append('name', 'test.txt');

//fd.append('file', event.target.result);

fd.append('datei', blob, 'test.xyz');

console.log(fd);

$.ajax({

type: 'POST',

url: 'http://audio.stichprobe.eu/upload.php',

data: fd,

dataType: "text",

crossDomain: **true**,

headers: {

"Access-Control-Allow-Origin" : "\*",

"Access-Control-Allow-Methods" : "GET,POST,PUT,DELETE,OPTIONS",

"Access-Control-Allow-Headers": "Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"

},

processData: **false**,

//contentType: "Content-Type: audio/mpeg"

//contentType: 'text/html'

contentType: **false**

}).done(**function**(data) {

// print the output from the upload.php script

console.log(data);

});

};

//reader.onload = function() {

// console.log(reader.result);

//};

reader.onerror = **function**() {

console.log(reader.error);

};

console.log("trigger the read from the reader...");

// trigger the read from the reader...

reader.readAsDataURL(blob);

//reader.readAsText (blob);

console.log("... reader.readAsDataURL is triggered");

}

</script>

</head>

<body>

<form>

<button id=*"send"* type=*"button"*>Send</button>

</form>

</body>

<script type=*"text/javascript"*>

**const** sendButton = document.getElementById('send');

sendButton.addEventListener('click', **function**() {

uploadBlob ();

//statusText.innerHTML = "clicked";

});

</script>

</html>