Technical Articles

lokesh kumar

July 26, 2021 | 4 minute read

File Upload and Download in SAP UI5

0 7 854

Follow

Like

RSS Feed

Hi all,

In this blog post I will explain how to upload and download a text file or image file or xlsx sheet in SAP UI5 Applications.

Prerequisites:-

you should have knowledge about

- base64
- file reader

- blob
- atob
- bota

Base64

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. The term Base64 originates from a specific MIME content transfer encoding.

Base64 encoding schemes are commonly used when there is a need to encode binary data that needs to be stored and transferred over media that are designed to deal with ASCII. This is to ensure that the data remain intact without modification during transport. Base64 is commonly used in a number of applications including email via MIME, and storing complex data in XML.

One common application of Base64 encoding on the web is to encode binary data so it can be included in a data: URL.

In JavaScript there are two functions respectively for decoding and encoding Base64 strings:

atob() Method:-

The **atob**() method decodes a base-64 encoded string. This method decodes a string of data which has been encoded by the btoa() method.

decodes a Base64-encoded string(“atob” should be read as “ASCII to binary”).

btoa() Method:-

The btoa() method encodes a string in base-64. (“btoa” should be read as “binary to ASCII”).

This method uses the “A-Z”, “a-z”, “0-9”, “+”, “/” and “=” characters to encode the string.

Blob

The Blob object represents a blob, which is a file-like object of immutable, raw data; they can be read as text or binary data, or converted into a ReadableStream so its methods can be used for processing the data

FileReader

The FileReader object lets web applications asynchronously read the contents of files (or raw data buffers) stored on the user's computer, using File or Blob objects to specify the file or data to read.

Event handlers:-

FileReader.onerror

A handler for the error event. This event is triggered each time the reading operation encounter an error.

FileReader.onload

A handler for the load event. This event is triggered each time the reading operation is successfully completed.

Methods:-

FileReader.readAsBinaryString()

Starts reading the contents of the specified Blob, once finished, the result attribute contains the raw binary data from the file as a string.

FileReader.readAsDataURL()

Starts reading the contents of the specified Blob, once finished, the result attribute contains a data: URL representing the file's data.

To Upload a File

We will use FileUploader Control, By using Id we can get the FileUploader object in the controller

View1.view.xml File

xmlns:ui="sap.ui.unified"

```
<ui:FileUploader id="fileUploaderFS" />

 <Button text="Upload" type="Accept" press="onUploadFile"/>
```

View1.controller.js File

when we trigger onUploadFile event in the button it will trigger the function in controller.

```
//This code is used for uploading documents and images
onUploadFile:function(){
 var oFileUpload =
this.getView().byId("fileUploaderFS");

 var domRef = oFileUpload.getFocusDomRef();
 var file = domRef.files[0];
 var that = this;

//This code is used for uploading image or document file

 this.fileName =
 this.getView().byId("FileName").getValue();
 this.fileType = file.type;

  var reader = new FileReader();
           reader.onload = function (e) {
var vContent = e.currentTarget.result

 that.updateFile(that.fileName, that.fileType, vContent);
           }
            reader.readAsDataURL(file);
       }



//This code is used for uploading xlsx sheets


   var oFileUpload =
this.getView().byId("fileUploaderTS");
  var domRef = oFileUpload.getFocusDomRef();
```

```
  var file = domRef.files[0];
  var that = this;

 this.fileName = this.getView().byId("TSFileName")
.getValue()+".csv";
 this.fileType = file.type;
 this.json_object=null

               var reader = new FileReader();
                 reader.onload = function(e) {
     var data = e.target.result;
     var workbook = XLSX.read(data, {
       type: 'binary'
     });
     console.log(workbook)

     workbook.SheetNames.forEach(function(sheetName) {

       var XL_row_object =
XLSX.utils.sheet_to_row_object_array(
workbook.Sheets[sheetName]);
       if(XL_row_object.length!==0){

    that.json_object = JSON.stringify(XL_row_object);

    }

    })
that.updateFile(that.fileName, that.fileType,
 that.json_object);

   };

   reader.onerror = function(ex) {
     console.log(ex);
```

```
   };

   reader.readAsBinaryString(file);

           }
```

To update the content in backend services

```
updateFile:function(fileName, fileType, vContent){
            debugger
            var payLoad={
                Filename:fileName,
                Filetype:fileType,
                Filecontent:vContent

            }
   var that = this;
   var serviceurl="/sap/opu/odata/sap/ZAPP_EMP1_SRV/";

  var oModel =
new sap.ui.model.odata.ODataModel(serviceurl); oModel .update("/FILESet('"+payLoad.Filename+"')/$valu
payLoad,{
                 method: "PUT",
                 success: function(data) {

  sap.m.MessageToast.show("FILE UPDATED SUCCESSFULLY");

                 },
                 error: function(e) {
                  alert("error");
                }
```

```
            })
           },
```

To download file from backend services

```
  getFile:function(fileName){

       fileName= fileName.toLowerCase()
           var that = this;
           var serviceurl="/sap/opu/odata/sap/servicename_SRV/";

         var oModel =  new sap.ui.model.odata.ODataModel(serviceurl);
          oModel .read("/EntitySet('"+fileName+"')/$value",{
              method: "GET",
              success: function(data,response) {
                  debugger;

             var fName=data.Filename
             var fType= data.Filetype;
             var fContent =data.Filecontent;

// If the file is document IS  pdf/msword/plain text


  if(fType==="text/plain" || fType==="application/pdf"                          || fType==="applicat
                 fContent =atob(fContent);

            File.save(fContent, fName, "txt", fType);

              }
//If the file is excel sheet
```

```
 else if(fName.includes(".csv")){
              var  JSONData=fContent;
        Var    ReportTitle =fName;


If JSONData is not an object then JSON.parse will parse the JSON string in an Object
     var arrData = typeof JSONData !=
'object' ? JSON.parse(JSONData) : JSONData;

            var CSV = '';
            Set Report title in first row or line

            CSV += ReportTitle + '\r\n\n';



            //1st loop is to extract each row
            for (var i = 0; i < arrData.length; i++) {
                var row = "";

       //2nd loop will extract each column and convert it in
 string comma-seprated
                for (var index in arrData[i]) {
                    row += '"' + arrData[i][index] + '",';
                }

                row.slice(0, row.length - 1);

                //add a line break after each row
                CSV += row + '\r\n';
            }

            if (CSV == '') {
                alert("Invalid data");
```

```
                return;
            }

            //Generate a file name
            var fileName = "MyNewReport_";
            //this will remove the blank-spaces from the title and replace it with an underscore
            fileName += ReportTitle.replace(/ /g,"_");

            //Initialize file format you want csv or xls
     var uri = 'data:text/csv;charset=utf-8,' + escape(CSV);

            // Now the little tricky part.
            // you can use either>> window.open(uri);
            // but this will not work in some browsers
            // or you will not get the correct file extension

            //this trick will generate a temp <a /> tag
            var link = document.createElement("a");
            link.href = uri;

 //set the visibility hidden so it will not effect on your web-layout
            link.style = "visibility:hidden";
            link.download = fileName + ".csv";
             //this part will append the anchor tag and remove it after automatic click
            document.body.appendChild(link);
            link.click();
            document.body.removeChild(link);


             }

//If the file is image


              else{
```

```
                fContent =atob(data.Filecontent);
            var byteNumbers= new Array(fContent.length);
for (let index = 0; index < fContent.length; index++) {
         byteNumbers[index]=fContent.charCodeAt(index)

            }
            var byteArray= new Uint8Array(byteNumbers)
            var blob= new Blob([byteArray],{type:fType});
            var url=URL.createObjectURL(blob)
            window.open(url,"_blank");
        }
       MessageToast.show("FILE Downloaded Succesfully");
        },
        error: function(e) {
            console.log(e)
         alert("error");
       }
    })
   },
```

Conclusion:-

When ever we want to upload or download a file we can use FileUploader control to load the file and we can read the data present in FileUploader Object by using FileReader Object and its methods and get the content present in it . and store it in backend DB. By using atob object we can convert data present in base64 to string format and using File object we can download the file data in specific format.

Alert Moderator

Assigned tags

SAPUI5

Similar Blog Posts

Upload/download File in SAP UI5 Application using Gateway

By Bilal Muhammad Nov 08, 2018

Document management using MongoDB (Part 2: Downloading document)

By Former Member Jan 06, 2017

SAP UI5 1.60 PATCH UPGRADE ABAP

By Ansari Mohammed Shah azim Apr 01, 2020

Related Questions

Upload PDF file to Drive using SAP HANA XS Advance

By Vadhiraj Mysore Apr 18, 2021

File Upload

By Former Member Feb 04, 2014

Open the attachments in SAP UI5?

By Former Member Sep 17, 2014

Be the first to leave a comment

You must be Logged on to comment or reply to a post.

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support