# کلاس های دسترسی و آپلود فایل

## آپلود فایل‌ها

APIهای فایل در نسخه `appsan-js:0.1.51` اضافه شده‌اند و امکاناتی برای آپلود فایل‌ها فراهم می‌کنند. در این بخش به معرفی کلاس‌های `Blob`، `File` و `FileList` و توابع مربوط به آن‌ها پرداخته می‌شود.

## کلاس Blob

کلاس `Blob` برای مدیریت داده‌های دودویی (binary) استفاده می‌شود.

### 1. `constructor`

این تابع برای ایجاد یک شیء `Blob` استفاده می‌شود.

**ورودی‌ها:**

- `blobParts: (string|Blob)[]` - بخش‌های مختلف بلاپ که می‌تواند رشته ای بلاپ باشد.
- `options?: { type?: string }` - (اختیاری) نوع داده‌ها

**نوع بازگشتی:** `Blob`

**مثال:**

```
const blob = new Blob(['Hello, world!'], { type: 'text/plain' });
```

این کد یک شیء `Blob` با محتوای `Hello, world!` و نوع `text/plain` ایجاد می‌کند.

## کلاس File

کلاس `File` که از کلاس `Blob` ارث‌بری می‌کند، برای مدیریت فایل‌ها استفاده می‌شود.

### 1. `constructor`

این تابع برای ایجاد یک شیء `File` استفاده می‌شود.

**ورودی‌ها:**

- `fileBits: (string|Blob)[]` - بخش‌های مختلف فایل که می‌تواند رشته ای بلاپ باشد.
- `fileName: string` - نام فایل
- `options?: { type?: string, lastModified?: number }` - (اختیاری) نوع داده‌ها و زمان آخرین ویرایش

**نوع بازگشتی:** `File`

**مثال:**

```
const file = new File(['Hello, world!'], 'hello.txt', { type: 'text/plain', lastModified: Date.now() });
```

این کد یک شیء `File` با محتوای `Hello, world!`، نام `hello.txt` و نوع `text/plain` ایجاد می‌کند.

# کلاس FileList

کلاس `FileList` برای مدیریت لیستی از فایل‌ها استفاده می‌شود.

## 1. `item`

این تابع یک فایل را با استفاده از اندیس آن در لیست برمی‌گرداند.

**ورودی‌ها:**

- `index: number` - اندیس فایل مورد نظر در لیست

**نوع بازگشتی:** `File | null`

**مثال:**

```
const fileList = document.getElementById('fileInput').files;
const file = fileList.item(0);
console.log(file.name);
```

این کد اولین فایل در ورودی فایل را برمی‌گرداند و نام آن را در کنسول چاپ می‌کند.

## 2. `[index: number]`

این روش نیز مشابه `item` عمل می‌کند و یک فایل را با استفاده از اندیس آن در لیست برمی‌گرداند.

**ورودی‌ها:**

- `index: number` - اندیس فایل مورد نظر در لیست

**نوع بازگشتی:** `File`

**مثال:**

```
const file = fileList[0];
console.log(file.name);
```

این کد اولین فایل در لیست را برمی‌گرداند و نام آن را در کنسول چاپ می‌کند.

# مثال: گرفتن فایل از انتخابگر فایل و آپلود آن به سرور

در این مثال یک صفحه برای آپلود فایل ساخته شده است که شامل سه دکمه برای انتخاب فایل از گالری، انتخاب فایل از دوربین و آپلود فایل می‌باشد.

## کد صفحه:

```
<page title="File Uploader">
    <text text="File Uploader"/>
    <button text="choose from gallery" onclick="imagepicker/gallery-
>script/script:fileSelected($data)"/>
    <button text="choose from camera" onclick="imagepicker/camera-
>script/script:fileSelected($data)"/>
    <button text="upload file" onclick="script/script:uploadFile()"/>

    <script id="script">
        var selectedFile;

        function uploadFile() {
            if (!selectedFile) {
                Appsan.runAction("snackbar/'No file selected.'");
                return;
            }

            const formData = new FormData();
            formData.set('file', selectedFile);
```

```
            HTTPClient.post('http://myUploadServer/upload', formData, {
                headers:{
                    Authorization: 'Bearer 123...'
                }
            }).subscribe(res => {
                console.log(res);
            }, err => {
                console.log(err);
            });
        }

        function fileSelected(files) {
            selectedFile = files[0];
            console.log(files);
        }
    </script>
</page>
```

## توضیح کد:

هر یک از دکمه‌ها توضیحات و عملکرد خاصی دارند:

**دکمه "choose from gallery":**

```
<button text="choose from gallery" onclick="imagepicker/gallery-
>script/script:fileSelected($data)"/>
```

این دکمه یک اکشن `imagepicker/gallery` را فراخوانی می‌کند که انتخاب فایل از گالری را انجام می‌دهد. پس از تکمیل اکشن، داده‌ها به صورت `$data` در دسترس خواهند بود و به تابع `fileSelected($data)` در اسکریپت `script` (با شناسه `script`) ارسال می‌شود. در این اکشن:

- `imagepicker/gallery`: اکشن انتخاب فایل از گالری
- `->`: نشان‌دهنده اتصال دو اکشن است
- `script/script:fileSelected($data)`: اکشن دوم که نام اکشن `script`، شناسه اسکریپت `script` و تابع `fileSelected` را فراخوانی می‌کند و داده‌های `$data` را به آن ارسال می‌کند.

**دکمه "choose from camera":**

```
<button text="choose from camera" onclick="imagepicker/camera-
>script/script:fileSelected($data)"/>
```

این دکمه یک اکشن `imagepicker/camera` را فراخوانی می‌کند که انتخاب فایل از دوربین را انجام می‌دهد. پس از تکمیل اکشن، داده‌ها به صورت `$data` در دسترس خواهند بود و به تابع `fileSelected($data)` در اسکریپت `script` (با شناسه `script`) ارسال می‌شود. در این اکشن:

- `imagepicker/camera`: اکشن انتخاب فایل از دوربین
- `->`: نشان‌دهنده اتصال دو اکشن است
- `script/script:fileSelected($data)`: اکشن دوم که نام اکشن `script`، شناسه اسکریپت `script` و تابع `fileSelected` را فراخوانی می‌کند و داده‌های `$data` را به آن ارسال می‌کند.

**دکمه "upload file":**

```
<button text="upload file" onclick="script/script:uploadFile()"/>
```

این دکمه تابع `uploadFile()` را در اسکریپت `script` (با شناسه `script`) فراخوانی می‌کند. در این اکشن:

- `script/script:uploadFile()`: اکشن آپلود فایل که نام اکشن `script`، شناسه اسکریپت `script` و تابع `uploadFile` را فراخوانی می‌کند.

**اسکریپت:**

```
<script id="script">
    var selectedFile;

    function uploadFile() {
        if (!selectedFile) {
            Appsan.runAction("snackbar/'No file selected.'");
            return;
        }

        const formData = new FormData();
        formData.set('file', selectedFile);

        HTTPClient.post('http://myUploadServer/upload', formData, {
            headers:{
                Authorization: 'Bearer 123...'
            }
        }).subscribe(res => {
            console.log(res);
        }, err => {
            console.log(err);
        });
```

```
    }

    function fileSelected(files) {
        selectedFile = files[0];
        console.log(files);
    }
</script>
```

**متغیر `selectedFile`:**

```
var selectedFile;
```

این متغیر برای ذخیره فایل انتخاب شده استفاده می‌شود.

**تابع `uploadFile`:**

```
function uploadFile() {
    if (!selectedFile) {
        Appsan.runAction("snackbar/'No file selected.'");
        return;
    }

    const formData = new FormData();
    formData.set('file', selectedFile);

    HTTPClient.post('http://myUploadServer/upload', formData, {
        headers:{
            Authorization: 'Bearer 123...'
        }
    }).subscribe(res => {
        console.log(res);
    }, err => {
        console.log(err);
    });
}
```

این تابع بررسی می‌کند که آیا فایلی انتخاب شده است یا خیر. اگر فایلی انتخاب نشده باشد، یک پیام خطا نمایش داده می‌شود. در غیر این صورت، یک شیء `FormData` ایجاد می‌کند و فایل انتخاب شده را به آن اضافه می‌کند. سپس با استفاده از `HTTPClient` درخواست POST برای آپلود فایل به سرور ارسال می‌شود.

**تابع `fileSelected`:**

```
function fileSelected(files) {
    selectedFile = files[0];
    console.log(files);
}
```

این تابع فایل انتخاب شده را در متغیر `selectedFile` ذخیره می‌کند و لیست فایل‌ها را در کنسول چاپ می‌کند. این تابع توسط اکشن script پس از انتخاب فایل در کد صدا می شود.

---

Revision #7
Created 2024-07-06 14:57:05 UTC by a.dadgari
Updated 2024-07-06 15:30:19 UTC by a.dadgari