



## Subida de archivos con Http 22/23



### RECURSOS

### APUNTES

Una necesidad trivial de un formulario suele ser la carga de un archivo. Un PDF, una foto, etc. Cuando requieres de este tipo de Inputs, tienes que considerar que este archivo está compuesto por datos binarios que tienes que enviar al servidor.

## Guardado de binarios en un servidor

En las API REST, el envío de información se realiza en formato JSON con el header

`Content-type: application/json` . Pero para el envío de archivos binarios, se utiliza el header

`Content-type: multipart/form-data` .

Veamos a continuación cómo enviar este tipo de dato con Angular.

### 1. Formulario para envío de datos

Utilizando la clase **FormData**, puedes instanciar un formulario de este tipo y adjuntarle con `append()` el **Blob** del archivo que quieres enviar al servidor

```
// services/files.services.ts
export class FileService {
```

```

    constructor(
        private http: HttpClient,
    ) { }

    uploadFile(file: Blob): Observable<any> {
        const form = new FormData();
        form.append('file', file);
        return this.http.post(`https://example.com/api/files`, form, {
            headers: {
                'Content-type': 'multipart/form-data'
            }
        });
    }
}

```

## 2. Capturando el archivo

Ahora, desde el componente, tienes que crear un Input que capture el archivo y una función que lo manipule y realice la solicitud.

```

<div>
  <input type="file" (change)="onLoad($event)">
</div>

```

```

onLoad(event: Event): void {
    const element = event.target as HTMLInputElement;

```

```
const file = element.files?.item(0);
if (file) {
  this.filesService.uploadFile(file)
    .subscribe(res => {
      console.log(res);
    });
}
```









Cada vez que el usuario cargue un archivo, se ejecutará el método `uploadFile()` y a través del evento puedes manipular el elemento HTML y capturar el archivo en formato Blob.

Apóyate de la interfaz `HTMLInputElement` para tipar el elemento y capturar así el archivo.

---

*Contribución creada por: Kevin Fiorentino.*

## Archivos de la clase

 /	
 tsconfig.app.json	
 package.json	
 angular.json	
 .eslintrc.json	

 package-lock.json	
 .browserslistrc	
 .gitignore	
 obs-promises.js	
 tsconfig.spec.json	
 proxy.config.json	
 tsconfig.json	
 README.md	
 .editorconfig	
 karma.conf.js	
 src	