

— JAVASCRIPT —

File API. The Firebase Storage

FILE API

Доступ к выбранным файлам

```
1 | <input type="file" id="input">
```

```
1 | var selectedFile = document.getElementById('input').files[0];
```

Обращение к выбранным файлам через событие change

```
1 | <input type="file" id="input" onchange="handleFiles(this.files)">
```

```
1 | <input type="file" id="input" multiple onchange="handleFiles(this.files)">
```

```
1 | var inputElement = document.getElementById("input");  
2 | inputElement.addEventListener("change", handleFiles, false);  
3 | function handleFiles() {  
4 |     var fileList = this.files; /* теперь вы можете работать со списком файлов */  
5 | }
```

FILE API

Получение информации о выделенных файлах

```
1 | for (var i = 0, numFiles = files.length; i < numFiles; i++) {  
2 |     var file = files[i];  
3 |     ..  
4 | }
```

Всего существует три атрибута, предоставляемых объектом File, которые содержат полезную информацию о файле.

name

Имя файла как строка доступная только для чтения. Это просто имя файла и оно не включает в себя информацию о пути.

size

Размер файла в байтах, как 64-битное целое число (возможно только чтение).

type

MIME тип файла, как строка доступная только для чтения, или пустая строка ("") если тип файла не может быть определён.

Cloud Storage for Firebase

Получение ссылки на папку либо на конкретную картинку

```
// Get a reference to the storage service, which is used to create references in your storage bucket  
var storage = firebase.storage();
```

```
// Create a storage reference from our storage service  
var storageRef = storage.ref();
```

```
// Create a child reference  
var imagesRef = storageRef.child('images');  
// imagesRef now points to 'images'
```

```
// Child references can also take paths delimited by '/'  
var spaceRef = storageRef.child('images/space.jpg');  
// spaceRef now points to "images/space.jpg"  
// imagesRef still points to "images"
```

Cloud Storage for Firebase

Получение ссылки на папку либо на конкретную картинку

```
// Points to the root reference
var storageRef = firebase.storage().ref();

// Points to 'images'
var imagesRef = storageRef.child('images');

// Points to 'images/space.jpg'
// Note that you can use variables to create child values
var fileName = 'space.jpg';
var spaceRef = imagesRef.child(fileName);

// File path is 'images/space.jpg'
var path = spaceRef.fullPath

// File name is 'space.jpg'
var name = spaceRef.name

// Points to 'images'
var imagesRef = spaceRef.parent;
```

Cloud Storage for Firebase

Загрузка файлов в хранилище

```
var file = ... // use the Blob or File API
ref.put(file).then(function(snapshot) {
  console.log('Uploaded a blob or file!');
});
```

```
// Create file metadata including the content type
var metadata = {
  contentType: 'image/jpeg',
};

// Upload the file and metadata
var uploadTask = storageRef.child('images/mountains.jpg').put(file, metadata);
```

Cloud Storage for Firebase

Слушатель загрузки

```
// Register three observers:
// 1. 'state_changed' observer, called any time the state changes
// 2. Error observer, called on failure
// 3. Completion observer, called on successful completion
uploadTask.on('state_changed', function(snapshot){
  // Observe state change events such as progress, pause, and resume
  // Get task progress, including the number of bytes uploaded and the total number of bytes to be up
  var progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
  console.log('Upload is ' + progress + '% done');
  switch (snapshot.state) {
    case firebase.storage.TaskState.PAUSED: // or 'paused'
      console.log('Upload is paused');
      break;
    case firebase.storage.TaskState.RUNNING: // or 'running'
      console.log('Upload is running');
      break;
  }
}, function(error) {
  // Handle unsuccessful uploads
}, function() {
  // Handle successful uploads on complete
  // For instance, get the download URL: https://firebasestorage.googleapis.com/...
  var downloadURL = uploadTask.snapshot.downloadURL;
});
```

Cloud Storage for Firebase

Удаление файлов из хранилища

```
// Create a reference to the file to delete
var desertRef = storageRef.child('images/desert.jpg');

// Delete the file
desertRef.delete().then(function() {
    // File deleted successfully
}).catch(function(error) {
    // Uh-oh, an error occurred!
});
```


— JAVASCRIPT —

File API. The Firebase Storage