



WEB SERVER EXPRESS

ALMACENAMIENTO DE FICHEROS

Diagrama general



SISTEMA DE ALMACENAMIENTO DE
FICHEROS COMPATIBLE CON AWS S3

INSTALACIÓN EN LOCAL DEL SOFTWARE DE MINIO

```
curl https://dl.min.io/server/minio/release/windows-amd64/minio.exe -o minio.exe
```

```
.\minio.exe server .\minio --address localhost:9009 --console-address :9091
```

RootUser: minioadmin

RootPass: minioadmin

Instalar paquete para manejar minio desde nodejs

yarn add minio

```
const { MinioClient } = require('minio');

const minioClient = new Minio.Client({

  endPoint: "localhost",
  port: 9009,
  accessKey: "minioadmin",
  secretKey: "minioadmin",
  useSSL: false,

});
```

Creando un bucket

```
app.post("/ficheros/createBucket", async (req, res) => {  
  try {  
    await await minioClient.makeBucket(req.body.nombre, 'us-east-1')  
    res.send({ res: "00", description: "OK", data: { bucket: req.body.nombre } })  
  } catch (error) {  
    res.status(500).send({ error: error, res: "500", description: "ERR",  
      data: { bucket: req.body.nombre } })  
  }  
});
```

Añadir un fichero a un bucket

```
>  
<form action="/ficheros/addFile" method="post" enctype="multipart/form-data">  
  <p><input type="text" name="bucket" value="curso">  
  <p><input type="file" name="fichero">  
  <p><button type="submit">Submit</button>  
</form>
```

Public/formularioMinio.html



curso

Seleccionar archivo _10 JS JSON.mp4

Submit

```
app.post("/ficheros/addFile", async (req, res) => {  
  const file = req.files.fichero;  
  const bucket = req.body.bucket;  
  try {  
    const minioObject = await minioClient.putObject(  
      bucket,  
      file.name,  
      file.data,  
      file.size);  
    res.send({ res: "00", description: "OK" })  
  } catch (error) {  
    res.status(500).send({ error, res: "500", description: error })  
  }  
});
```

Ruta para subir un fichero

Recuperar un fichero de un bucket

```
app.get("/ficheros/:bucket/:nombre", async (req, res) => {  
  try {  
    try {  
      const info = await minioClient.statObject(  
        req.params.bucket, req.params.nombre);  
      const dataStream = await  
minioClient.getObject(req.params.bucket, req.params.nombre);  
  
      res.set("content-type", info.metaData["content-type"]);  
      dataStream.pipe(res);  
  
    } catch (error) {  
      res.status(500).send({r:11, error})  
      return  
    }  
  } catch (e) {  
    res.send(e);  
  }  
});
```

Delete un fichero de un bucket

```
app.delete("/delete/:bucket/:fichero", async (req, res) => {  
  minioClient.removeObject(req.params.bucket, req.params.fichero, function(err) {  
    if (err) {  
      res.status(500).send({err})  
    }  
    res.send({bucket:req.params.bucket, fichero:req.params.fichero})  
  })  
})
```


Delete de un fichero con async await

```
app.delete("/ficheros/m2/:bucket/:fichero", async (req, res) => {  
  try {  
    await minioClient.removeObject(req.params.bucket, req.params.fichero);  
    res.send({ bucket: req.params.bucket, fichero: req.params.fichero })  
  } catch (error) {  
    res.status(500).send({ err })  
  }  
})
```