

Handling File Uploads in Express

Express and Multi-Part Uploads

Historically I've used the express [Multer](#) middleware and associated typescript types (`@types/multer`).

Multer accepts files encoded as part of a `multipart/form-data` request. The file is added to the `req` object as `req.file : Express.Multer.File`

This is done by adding a middleware to the route that accepts files like so:

```
import multer from 'multer'

...

const uploadMiddleware = multer({limits:{
  fileSize: config.uploads.maxSize
}})

router.post('/blah',
  uploadMiddleware.single('file'),
  controllerFunction );
```

Storing Uploaded Files in S3

It is possible to use minio to immediately pass the file through to an S3 layer via the buffer object. Using a Minio client object:

```
async function handleFileUpload(req: Express.Request, res: Express.Response) {  
  
  const fileObj : Express.Multer.File = req.file  
  let file = await Minio.putObject(config.minio.bucket, objName, fileObj.buffer);  
  
  return {"status":"uploaded", file}  
  
}
```

Dealing with CSVs

Use [neat-csv](#) for a promisy experience

Revision #5

Created 1 November 2022 14:14:04 by James

Updated 21 January 2024 14:54:02 by James