

Alterando classe FileSaver

Transcrição

[00:00] Nós criamos a classe "AmazonConfiguration", com as respectivas configurações necessárias, para que a gente possa realizar autenticação com Amazon e depois chamar os respectivos métodos para enviar as imagens dos livros para o bucket da Amazon. É essa etapa que vamos fazer, justamente, alterar o nosso projeto aqui para que ele envie as imagens dos livros para o bucket que criamos na Amazon.

[00:21] Então quando vamos naquela parte de formulário para cadastrar um novo livro na aplicação da Casa do Código, essas informações vão ser passadas na nossa classe "ProdutosController" e aqui para o método "gravar". Se reparar, o método gravar, ele vai receber de todos os dados vindo do formulário, inclusive com a imagem do livro que seria esse objeto do tipo "MultipartFile".

[00:44] Esse livro aqui, essa imagem do livro, será passado para esse método de escrita nesse método "write". Então vamos clicar com Ctrl e o botão esquerdo do mouse, para ser redirecionado para ver o que este método está fazendo.

[00:59] Vemos que esse método está tratando e salvando a imagem localmente aqui no TOMCAT. E agora precisamos mudar isso, porque a gente quer salvar as imagens no bucket que criamos na Amazon. Então vamos apagar o conteúdo pessoal, desse nosso try e vamos apagar também essa parte desse atributo para poder fazer todo o tratamento agora para salvar a imagem no bucket da Amazon.

[01:20] Para poder fazer isso, precisamos daquele objeto do tipo "AmazonS3", que pedimos para o "Spring" fazer o gerenciamento desse objeto aqui para gente. Então agora eu vou pedir para o "Spring" fazer a injeção desse objeto aqui na classe "FileSaver". Vamos aqui e colocamos a anotação "@Autowired", e coloca o atributo como sendo "private", nosso objeto do tipo "AmazonS3", e coloca aqui "AmazonS3". E vamos colocar Ctrl+Shift+O, para fazer a importação dos pacotes certinho e chama esse atributo nosso "AmazonS3".

[01:53] Então o que a gente quer? Quer colocar um objeto agora, que seria essas imagens dos livros, no bucket que criamos na Amazon. Precisamos aqui de um método que é, justamente, colocar um objeto "PutObject".

[02:09] E reparem pessoal, que esse método espera receber aqui um objeto do tipo "PutObjectRequest". Então vamos instanciar essa classe "new PutObjectRequest". E olha só, agora o construtor dessa classe espera receber primeiramente o que é? O nome do bucket. Então, vamos voltar aqui no "Painel de Console" da Amazon para gente pegar o nome do nosso bucket.

[02:30] Eu tenho esse meu bucket aqui, deixa eu só aproveitar e copiar ele para não ter nenhum erro de digitação, e eu vou colocar também essa informação como sendo um atributo estático da classe, vamos colocar "private static final String BUCKET", e temos o nome desse nosso bucket, e simplesmente vem aqui, colocar o nome do nosso bucket. Agora pessoal, temos que colocar qual que seria o nome desses arquivos lá na Amazon.

[02:59] Então se a voltarmos aqui pessoal, esses arquivos, essas imagens dos livros, tem seus respectivos nomes originais "algoritmos.JPEG", "amazona_aws.JPEG" e assim por diante. Poderia salvar esses livros com esses mesmos nomes originais na Amazon.

[03:19] Eu quero pegar, justamente, esses nomes originais desses arquivos. Então podemos vir aqui e chamar o método do "MultiPartFile", já vai pegar aqui, justamente, esses nomes originais desses arquivos para salvarmos com esses mesmos nomes originais no bucket da Amazon.

[03:32] Vamos aqui, salvar esses arquivos, o "File" com o nome original. Pegamos um método "GetOriginalFilename". E agora temos que falar, de fato, o que contém esse arquivo, qual é o conteúdo, o fluxo de informações que formam esse arquivo. Então temos que passar, justamente, o conteúdo desse arquivo, basta chamar o método "getInputStream".

[03:58] Pessoal, agora deixa eu colocar um "enter" para poder visualizar melhor. Agora para poder finalizar aqui, poderia passar também, informações dos metadados, como o tipo do arquivo que está passando, informação de tempo de expiração, por exemplo.

[04:12] Então no nosso caso caso não precisamos nos preocupar com essa questão de metadados de informação desses arquivos, simplesmente pode passar o último argumento como sendo nulo, se quisesse passar esses metadados, poderíamos passar nesse quarto parâmetro aqui, mas não precisa, vamos colocar o nulo mesmo.

[04:30] Com isso estamos fazendo o que? Está passando os arquivos com seus respectivos nomes originais para serem salvos no bucket da Amazon. Só que por padrão a Amazon não deixa esses arquivos aqui, que vão estar no bucket, com uma leitura pública, ou seja, os usuários que forem acessar aplicação da Casa do Código depois, eles não conseguiriam ver as imagens dos livros.

[04:52] Então temos que mudar essa permissão. Logo aqui antes de fechar esse método do "PutObject", vamos vir aqui para configurar essa permissão para que a leituras dessas imagens sejam uma leitura pública.

[05:12] Então temos que vir aqui e chamar o método que é "withCannedAcl", ou seja, temos que configurar a lista de acesso, essa lista de permissão, para fazer com que esses arquivos do bucket tenha uma leitura pública. Esse método espera receber aqui um [ENUM] onde vamos especificar que a leitura desses arquivos deve ser o que?

[05:31] Deve ser uma leitura pública, deixa eu só colocar aqui, "PublicRead". Com isso estamos falando que todos os arquivos vão ter uma leitura pública, e aí todos os usuários que forem acessar aplicação da Casa do Código vão conseguir visualizar as imagens dos livros. Feito isso colocamos o ponto e vírgula.

[05:51] E agora pessoal, o que temos que fazer? Temos que retornar para esse método e ele vai retornar um "String", justamente, com a URL de acesso desses arquivos dentro do bucket. E não temos muito o que fazer, temos que ir pesquisar na documentação da Amazon para ver como seriam essas "URLs" formadas pela Amazon, quando temos essas imagens nos buckets.

[06:15] Então vamos pesquisar aqui, numa nova janela, para justamente, procurar essas informações na documentação. Vou colocar aqui em inglês pessoal, como que seria "trabalhando com S3" na Amazon, para a gente ser redirecionado para documentação. Então vou colocar "Working with Amazon S3", para ver como que seria trabalhar com esses buckets que a gente cria com o serviço "S3".

[06:39] Logo no primeiro link aqui, eu já tenho a documentação da Amazon, e aqui se a gente for para baixo o pessoal, da documentação da Amazon, um pouquinho mais pra baixo aqui, olha só, a Amazon especifica como que seria a "URL" de acesso desses arquivos que vão ficar dentro do "bucket". Então, temos essa informação aqui que seria por essa "URL" "s3.amazonaws.com", e passaríamos o nome do nosso bucket.

[07:06] Vamos copiar essa informação dessa URL para poder acessar os arquivos dentro do bucket. E a gente vai o que? Vamos, simplesmente, retornar essa URL aqui pessoal, para poder acessar esses arquivos. Então vou colocar as aspas, e colocamos aqui, só não podemos esquecer que o bucket agora, tem que ser com o nome do nosso bucket. Vamos só colocar aqui as aspas para fechar, o mais para realizar a concatenação, e vamos passar o nome do nosso bucket.

[07:34] Só que o bucket pode ter vários arquivos lá dentro, temos que especificar que arquivo que a gente quer pegar. E a gente está salvando os arquivos como? Com os seus respectivos nomes originais no bucket da Amazon.

[07:46] Então eu vou concatenar e vou passar a barra, e dentro do bucket quer pegar o que? Quer pegar os nomes dos arquivos com seus respectivos nomes originais. Chamamos o método aqui, "GetOriginalFileName", e basta a gente vir aqui, esqueci de fazer a concatenação, agora sim.

[08:06] Maravilha pessoal, então com isso a gente está o que? Está conseguindo passar esses arquivos para o bucket da Amazon. E a gente está o que? Estamos retornando a URL para poder acessar esses arquivos posteriormente no bucket da Amazon. Agora falta a gente fazer um teste aqui na nossa aplicação e ver se a gente consegue salvar as imagens dos livros no bucket da Amazon. Vamos fazer isso lá na próxima etapa.