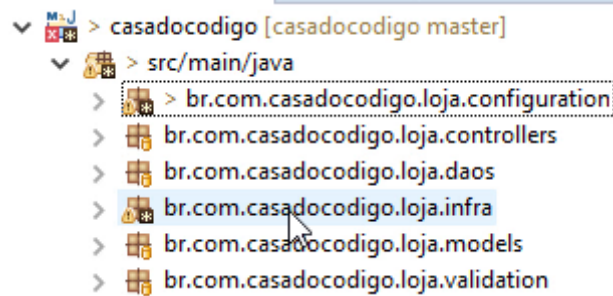


Agora que já fizemos as devidas configurações para realizarmos a autenticação com a Amazon, devemos fazer a configuração do envio das imagens para o Bucket que criamos. Para isso vá até a classe **FileSaver** que está localizado dentro do pacote **infra**



Na classe **FileSaver** tínhamos o projeto da Casa do Código salvando as imagens dos livros inicialmente no Tomcat, vamos apagar o atributo da classe e remover também o conteúdo dentro do **try** para que possamos assim fazer o envio para o Bucket na Amazon. Para isso, vamos precisar do objeto **AmazonS3** que está sendo gerenciado pelo Spring, vamos pedir para o Spring fazer a injeção desse objeto na classe **FileSaver**.

@Autowired

```
private AmazonS3 amazonS3;
```

COPIAR CÓDIGO

Feito isso, dentro do try chame o método **putObject** da classe **AmazonS3** e dentro desse método instancie a classe **PutObjectRequest** para que assim possamos passar informações do nome do Bucket, nome do arquivo, fluxo de dados que compõem o arquivo e que serão passados para o Bucket bem como se desejamos colocar metadados de informações. Não podemos nos esquecer também de deixar tais arquivos do Bucket com visibilidade pública, ou seja, qualquer usuário terá permissão de visualizar tais arquivos de imagem e retornar a URL de acesso ao arquivo:

```
private static final String BUCKET="[NOME DO MEU BUCKET]"
```

```
try {
```

```
    amazonS3.putObject(new PutObjectRequest(BUCKET,  
        file.getOriginalFilename(), file.getInputStream(),null)  
        .withCannedAcl(CannedAccessControlList.PublicRead));
```

```
    return "http://s3.amazonaws.com/"+BUCKET+"/"+file.getOriginalFilename();
```

```
}
```

[COPIAR CÓDIGO](#)

Na sequência, iremos fazer um teste e veremos se conseguimos enviar as imagens dos livros para esse Bucket.