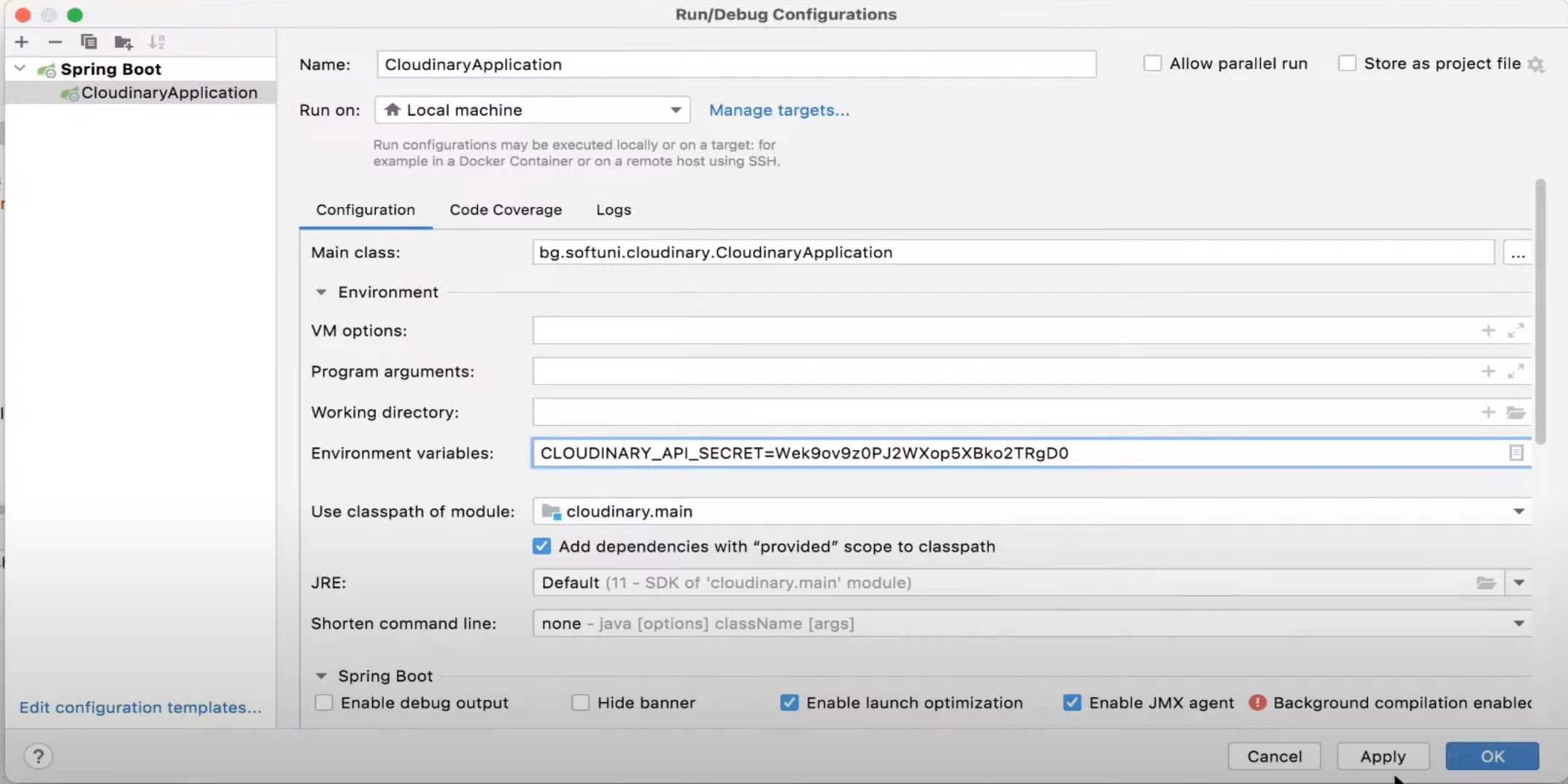
1. Добавя се dependency

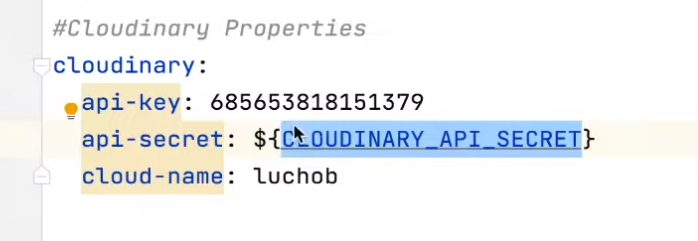


1. Добявя се environment variable – API\_SECRET



1. Добавя се настройките в application.properties

* Важно е да няма спейс пред cloudinary, защото че го причисли към Spring йерархията

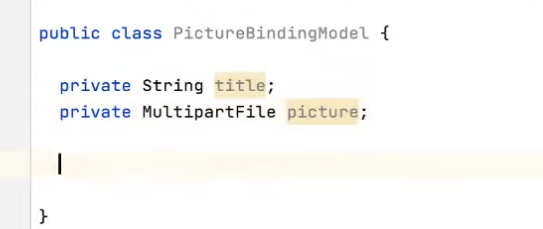


1. Създава се форма в html с enctype = “multpart/form-data” и type=”file” на снимката



1. Създава се bindingModel

* Необходимо е полетата в BindingModel-a да имат същите имена както в HTML формата



1. Добявя се депендънси за конфигурацията през файл за Annotation Processor



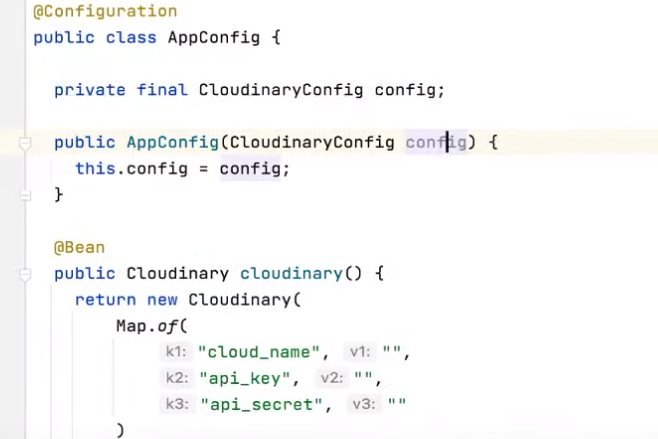
1. Създава се клас за настройки на Cloudinary

* Трябва имената да се camel case и автоматично ще си вземе имената на полетата от application.properties

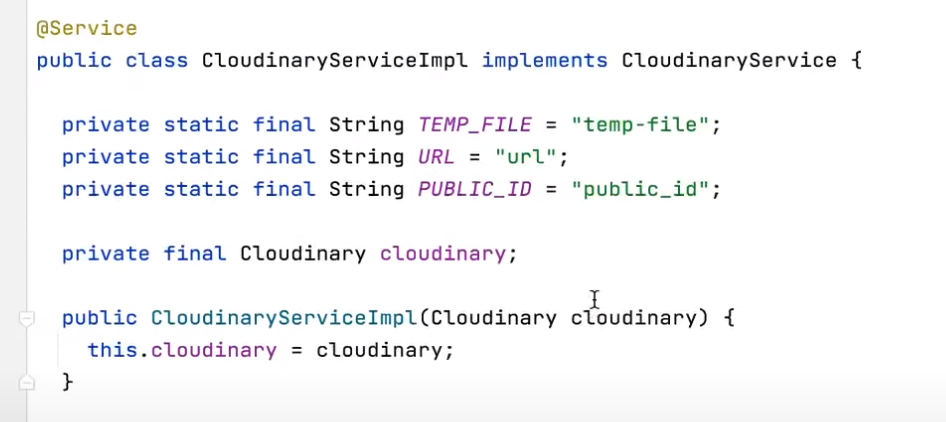


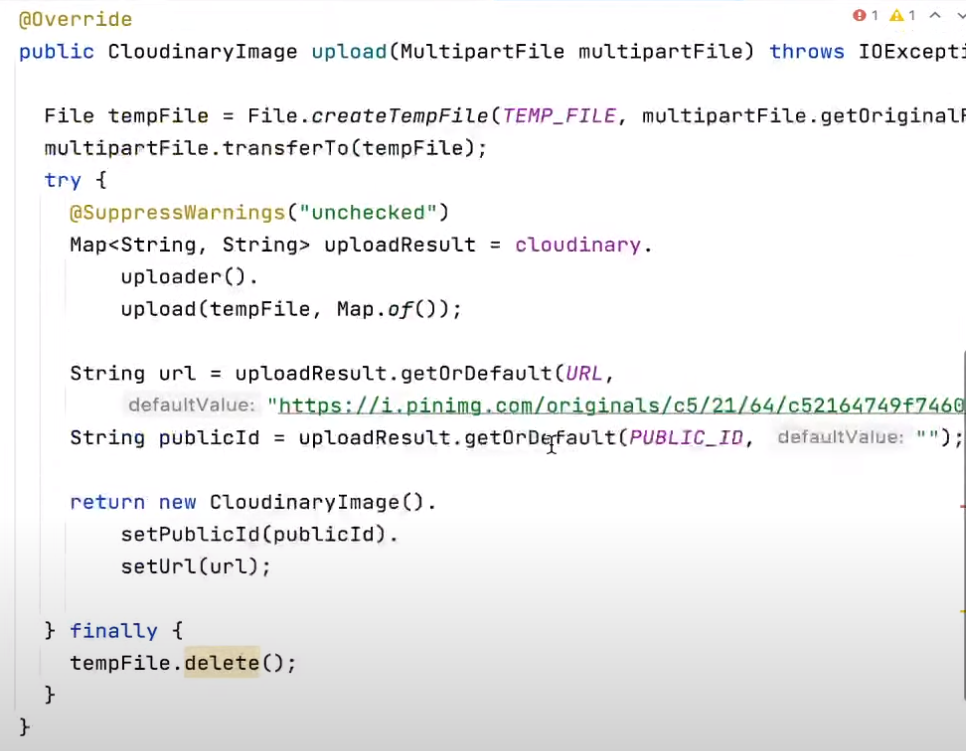
1. Съдава се друг конфигурационен файл, който ще се свърже с cloud-a чрез @Bean Cloudinary

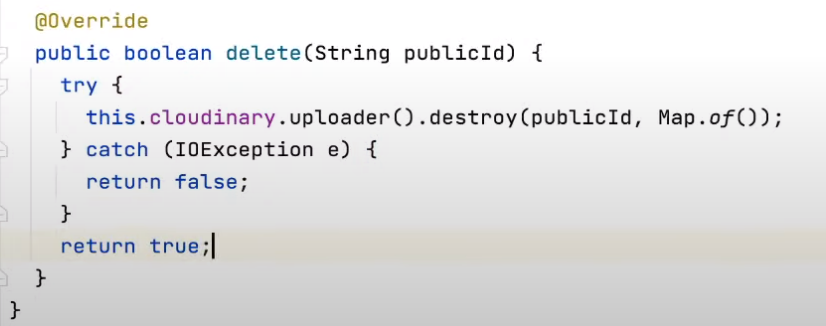
* V1,v2,v3 се задават чрез гетърите на config обекта



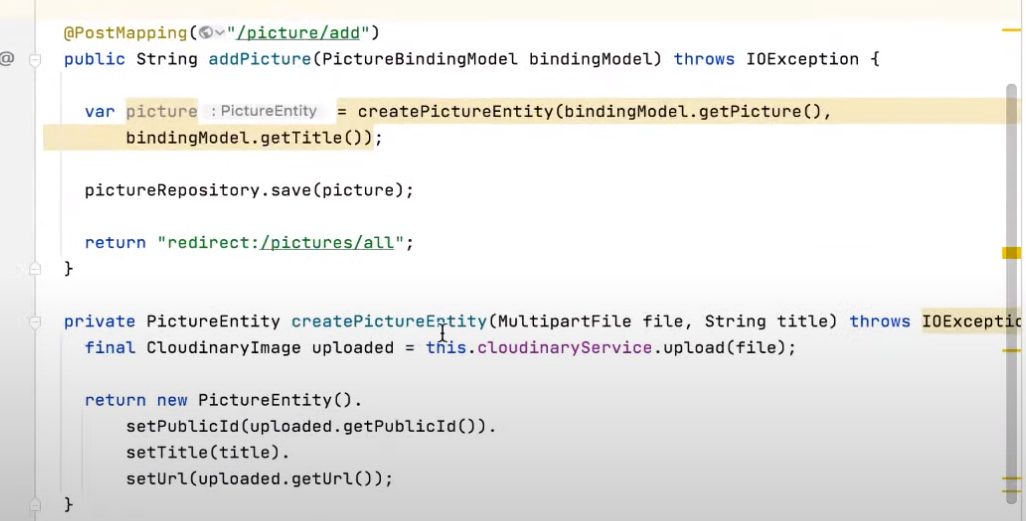
1. Създава се Service за Cloudinary







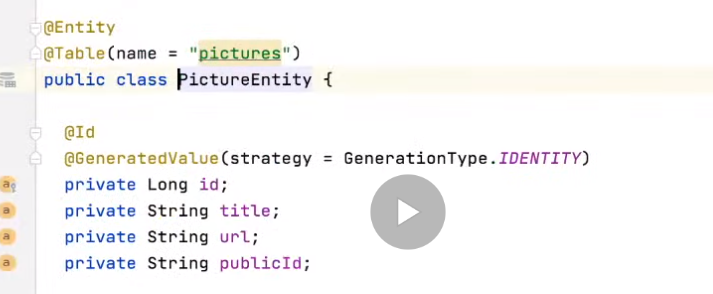
1. През контролера се инжектира CloudinaryService и се извикават методи upload/delete



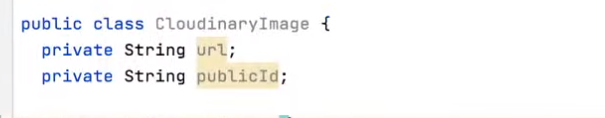
* Слага се @Transactional



1. PictureEntity



1. Cloudinary image – съдържа url и publicId



1. Визуализиране на всички pictures

Name=”public\_id” се чете като @RequestParam(“public\_id”) от контролера

