How response Is converted to json :-

@ResponseBody + JacksonHttpMessageConverterAutoConfiguration class or JacksonAutoConfiguration

Class

ErrorMvcAutoConfiguration

Is responsible for getting that white label error page , from this autoconfiguration class only that text is coming from., I mean if mappings are not found for the requested url then those scenarios are handled by this class

# **What is the advantage of spring boot**

1. Auto Configuration –refer spring.factories file in auto configuration jar and their Packages
2. Starter dependency - wide variety of starters –no need to solve version conflicts,
3. Servers- just need to add jars –tomcat server will come automatically
4. With spring boot we can develop microservices

**Auto configuration example :-**

All the logic for Spring Boot Auto Configuration

is present in a jar file

called Spring Boot Auto Configure dot Jar.

Beans like datasource, Dispatcher servlet, Internal view resolver will be configured automatically

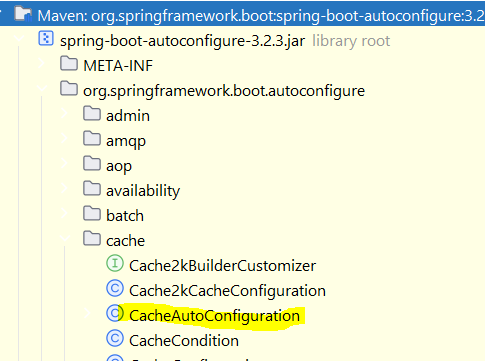
And misc beans like json bean converter and many other required beans all will be created automatically by \*Autoconfiguration classes, which will internally work based on

Annotations like @ConditionalOnClass() , @ConditionalOnBean(), @ConditionalOnMissingBean() annotations

## Auto configuration

1. If u want to know what beans have been automatically configured for you by spring boot for us then

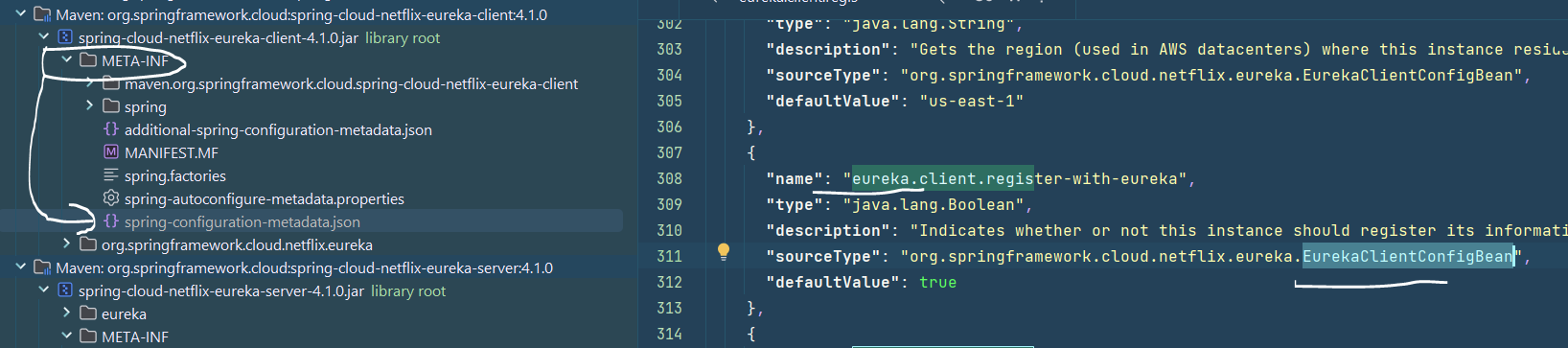
Open jar named “spring-boot-autoconfigure” and open related package which u are looking for as below



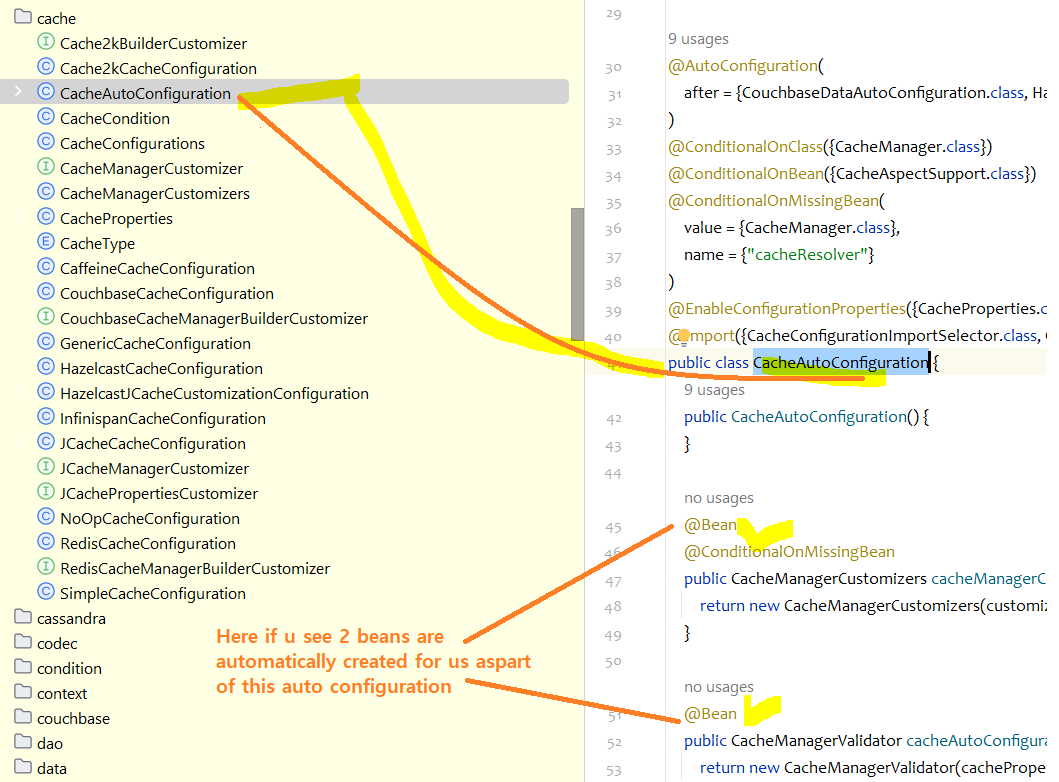
1. All the properties u keep will be binded to some @ConfigurationProperties bean class

If u want to know to which class it is binded then either “Ctrl+shift+F” and search for that property using “Directory” or “scope” button

Or Open any/relevant jar’s META-INF/“spring-configuration-metadata.json” there ur property will be present and it have the class name to which it is going to bind



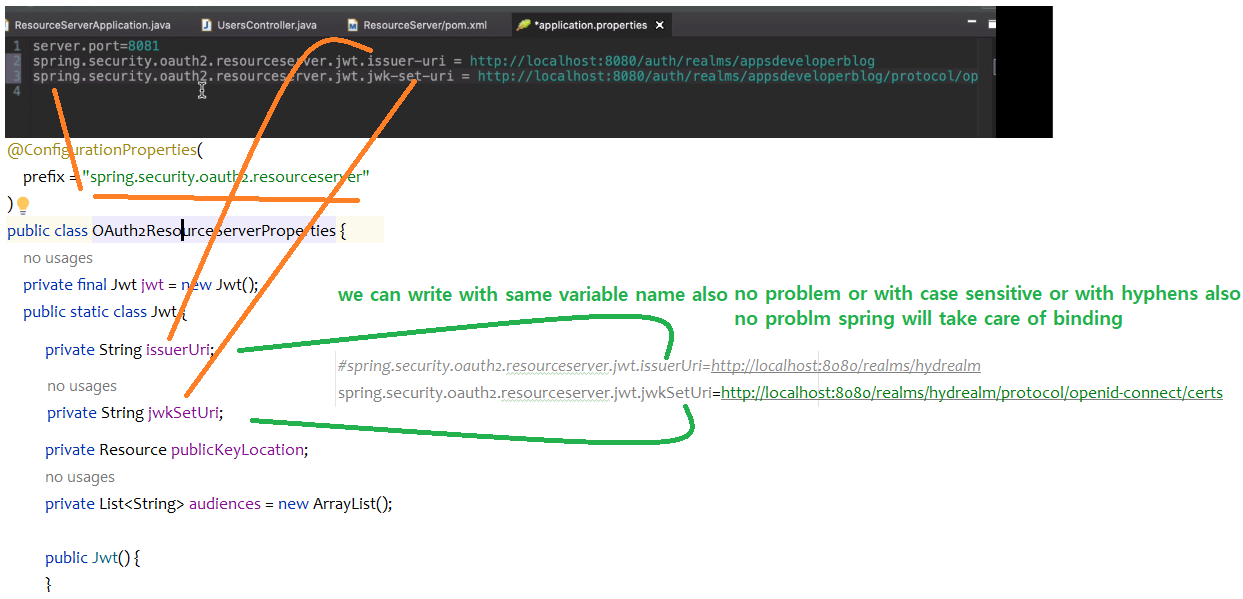
1. Open related \*Autoconfiguration class and see which beans are automatically configured by that auto configuration class



## Auto configuration properties

1. Generally, if u add a property some bean will be created for us as part of @ConditionalOnProperty annotation, if u want to see what are those properties and

To which class these are getting binded then follow below



If u see here these 2 properties are binded to this properties class