Externalized properties

* property-source
* property-source-env
* property-source-multifile
* property-source-spring-boot

Profile specific

* spring-profiles
* spring-profiles-unit-test
* spring-profiles-runtime
* profile-properties
* properites-code-assign
* profile-yaml-props

mysql

* mysql-maven
* mysql-qa-datasource
* service-account
* code-assignment-complete

CI

* [https://tecadmin.net/install-java-8-on-centos-rhel-and-fedora/#](https://tecadmin.net/install-java-8-on-centos-rhel-and-fedora/)
* <https://wiki.jenkins.io/display/JENKINS/Installing+Jenkins+on+Red+Hat+distributions>

artifactory

* <https://docs.docker.com/install/linux/docker-ee/rhel/>
* https://www.jfrog.com/confluence/display/RTF/Configuring+Apache
* artifactory-distro-config
* artifactory-docker
* artifactory-resolving

virtual cloud deployment

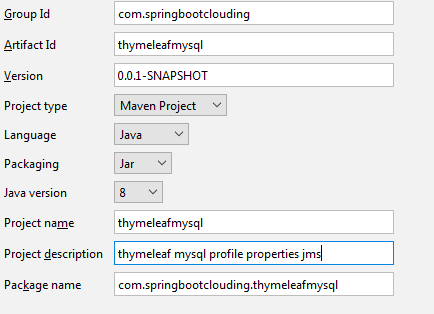
* prod-mysql-config
* running-spring-boot-application
* springbootservice

rds

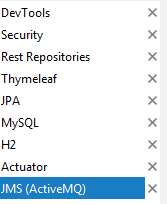
* rds-profile

start project

1.

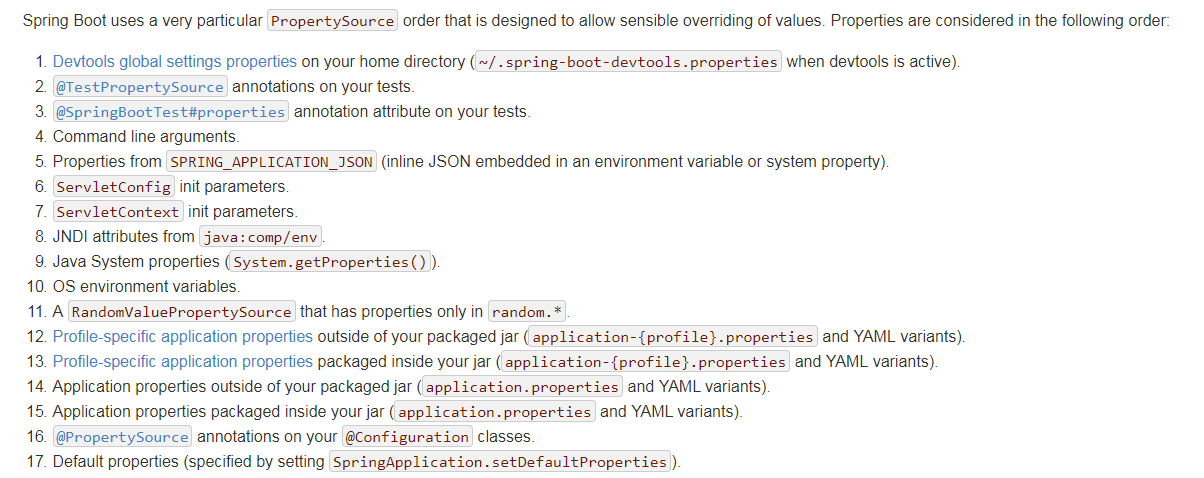


2



Properties load hierarchy

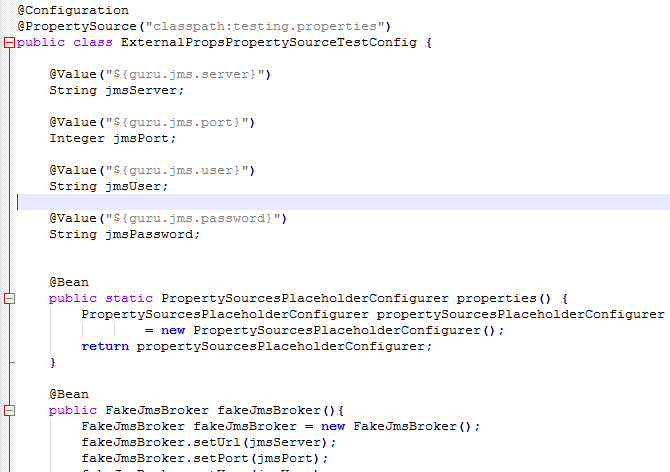
https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-external-config.html



Load external properties when integration test

runtime

@value properties



Environment property



@profile

When specify the profile value, the class with the profile will be active

1. Define three datasource classes.(devdatasource,qadatasource,proddatasource)
2. @profile(“dev”) @profile(“qa”) @profile(“prod”) in front of each datasource
3. @componentscan(“ all datasource class path”)

* Set default profile

@profile(“dev”,”default”)

@activeprofile(“dev”)