链滴

Springboot application.properties 常用配置

作者：yp

原文链接：https://ld246.com/article/1523118485780

来源网站：链滴

<h4 id="SPRING-CONFIG--ConfigFileApplicationListener-">SPRING CONFIG (ConfigFileAppli
ationListener)</h4>
spring.config.name
配置文件名称，默认为 application
spring.config.location
配置文件存放位置，默认为 classpath 目录下
<h4 id="mvc">mvc</h4>
spring.mvc.async.request-timeout
设定 async 请求的超时时间，以毫秒为单位，如果没有设置的话，以具体实现的超时时间为准，比如
omcat 的 servlet3 的话是 10 秒.
spring.mvc.date-format
设定日期的格式，比如 dd/MM/yyyy.
spring.mvc.favicon.enabled
是否支持 favicon.ico，默认为: true
spring.mvc.ignore-default-model-on-redirect
在重定向时是否忽略默认 model 的内容，默认为 true
spring.mvc.locale
指定使用的 Locale.
spring.mvc.message-codes-resolver-format
指定 message codes 的格式化策略(PREFIX_ERROR_CODE,POSTFIX_ERROR_CODE).
spring.mvc.view.prefix
指定 mvc 视图的前缀.
spring.mvc.view.suffix
指定 mvc 视图的后缀.
<h4 id="messages">messages</h4>
spring.messages.basename
指定 message 的 basename，多个以逗号分隔，如果不加包名的话，默认从 classpath 路径开始，
认: messages
spring.messages.cache-seconds
设定加载的资源文件缓存失效时间，-1 的话为永不过期，默认为-1
spring.messages.encoding
设定 Message bundles 的编码，默认: UTF-8
<h4 id="mobile">mobile</h4>
spring.mobile.devicedelegatingviewresolver.enable-fallback
是否支持 fallback 的解决方案，默认 false
spring.mobile.devicedelegatingviewresolver.enabled
是否开始 device view resolver，默认为: false
spring.mobile.devicedelegatingviewresolver.mobile-prefix
设定 mobile 端视图的前缀，默认为:mobile/
spring.mobile.devicedelegatingviewresolver.mobile-suffix
设定 mobile 视图的后缀
spring.mobile.devicedelegatingviewresolver.normal-prefix
设定普通设备的视图前缀
spring.mobile.devicedelegatingviewresolver.normal-suffix
设定普通设备视图的后缀
spring.mobile.devicedelegatingviewresolver.tablet-prefix
设定平板设备视图前缀，默认:tablet/
spring.mobile.devicedelegatingviewresolver.tablet-suffix
设定平板设备视图后缀.
spring.mobile.sitepreference.enabled
是否启用 SitePreferenceHandler，默认为: true
<h4 id="view">view</h4>
spring.view.prefix
设定 mvc 视图的前缀.

spring.view.suffix
设定 mvc 视图的后缀.
<h4 id="resource">resource</h4>
spring.resources.add-mappings
是否开启默认的资源处理，默认为 true
spring.resources.cache-period
设定资源的缓存时效，以秒为单位.
spring.resources.chain.cache
是否开启缓存，默认为: true
spring.resources.chain.enabled
是否开启资源 handling chain，默认为 false
spring.resources.chain.html-application-cache
是否开启 h5 应用的 cache manifest 重写，默认为: false
spring.resources.chain.strategy.content.enabled
是否开启内容版本策略，默认为 false
spring.resources.chain.strategy.content.paths
指定要应用的版本的路径，多个以逗号分隔，默认为:[/**]
spring.resources.chain.strategy.fixed.enabled
是否开启固定的版本策略，默认为 false
spring.resources.chain.strategy.fixed.paths
指定要应用版本策略的路径，多个以逗号分隔
spring.resources.chain.strategy.fixed.version
指定版本策略使用的版本号
spring.resources.static-locations
指定静态资源路径，默认为 classpath:[/META-INF/resources/,/resources/, /static/, /public/]以及
ontext:/
<h4 id="multipart">multipart</h4>
multipart.enabled
是否开启文件上传支持，默认为 true
multipart.file-size-threshold
设定文件写入磁盘的阈值，单位为 MB 或 KB，默认为 0
multipart.location
指定文件上传路径.
multipart.max-file-size
指定文件大小最大值，默认 1MB
multipart.max-request-size
指定每次请求的最大值，默认为 10MB
<h4 id="freemarker">freemarker</h4>
spring.freemarker.allow-request-override
指定 HttpServletRequest 的属性是否可以覆盖 controller 的 model 的同名项
spring.freemarker.allow-session-override
指定 HttpSession 的属性是否可以覆盖 controller 的 model 的同名项
spring.freemarker.cache
是否开启 template caching.
spring.freemarker.charset
设定 Template 的编码.
spring.freemarker.check-template-location
是否检查 templates 路径是否存在.
spring.freemarker.content-type
设定 Content-Type.
spring.freemarker.enabled
是否允许 mvc 使用 freemarker.
spring.freemarker.expose-request-attributes
设定所有 request 的属性在 merge 到模板的时候，是否要都添加到 model 中.

spring.freemarker.expose-session-attributes
设定所有 HttpSession 的属性在 merge 到模板的时候，是否要都添加到 model 中.
spring.freemarker.expose-spring-macro-helpers
设定是否以 springMacroRequestContext 的形式暴露 RequestContext 给 Spring’s macro library 使用
spring.freemarker.prefer-file-system-access
是否优先从文件系统加载 template，以支持热加载，默认为 true
spring.freemarker.prefix
设定 freemarker 模板的前缀.
spring.freemarker.request-context-attribute
指定 RequestContext 属性的名.
spring.freemarker.settings
设定 FreeMarker keys.
spring.freemarker.suffix
设定模板的后缀.
spring.freemarker.template-loader-path
设定模板的加载路径，多个以逗号分隔，默认: [“classpath:/templates/”]
spring.freemarker.view-names
指定使用模板的视图列表.
<h4 id="velocity">velocity</h4>
spring.velocity.allow-request-override
指定 HttpServletRequest 的属性是否可以覆盖 controller 的 model 的同名项
spring.velocity.allow-session-override
指定 HttpSession 的属性是否可以覆盖 controller 的 model 的同名项
spring.velocity.cache
是否开启模板缓存
spring.velocity.charset
设定模板编码
spring.velocity.check-template-location
是否检查模板路径是否存在.
spring.velocity.content-type
设定 ContentType 的值
spring.velocity.date-tool-attribute
设定暴露给 velocity 上下文使用的 DateTool 的名
spring.velocity.enabled
设定是否允许 mvc 使用 velocity
spring.velocity.expose-request-attributes
是否在 merge 模板的时候，将 request 属性都添加到 model 中
spring.velocity.expose-session-attributes
是否在 merge 模板的时候，将 HttpSession 属性都添加到 model 中
spring.velocity.expose-spring-macro-helpers
设定是否以 springMacroRequestContext 的名来暴露 RequestContext 给 Spring’s macro 类库用
spring.velocity.number-tool-attribute
设定暴露给 velocity 上下文的 NumberTool 的名
spring.velocity.prefer-file-system-access
是否优先从文件系统加载模板以支持热加载，默认为 true
spring.velocity.prefix
设定 velocity 模板的前缀.
spring.velocity.properties
设置 velocity 的额外属性.
spring.velocity.request-context-attribute
设定 RequestContext attribute 的名.
spring.velocity.resource-loader-path

设定模板路径，默认为: classpath:/templates/
spring.velocity.suffix
设定 velocity 模板的后缀.
spring.velocity.toolbox-config-location
设定 Velocity Toolbox 配置文件的路径，比如 /WEB-INF/toolbox.xml.
spring.velocity.view-names
设定需要解析的视图名称.
<h4 id="thymeleaf">thymeleaf</h4>
spring.thymeleaf.cache
是否开启模板缓存，默认 true
spring.thymeleaf.check-template-location
是否检查模板路径是否存在，默认 true
spring.thymeleaf.content-type
指定 Content-Type，默认为: text/html
spring.thymeleaf.enabled
是否允许 MVC 使用 Thymeleaf，默认为: true
spring.thymeleaf.encoding
指定模板的编码，默认为: UTF-8
spring.thymeleaf.excluded-view-names
指定不使用模板的视图名称，多个以逗号分隔.
spring.thymeleaf.mode
指定模板的模式，具体查看 StandardTemplateModeHandlers，默认为: HTML5
spring.thymeleaf.prefix
指定模板的前缀，默认为:classpath:/templates/
spring.thymeleaf.suffix
指定模板的后缀，默认为:.html
spring.thymeleaf.template-resolver-order
指定模板的解析顺序，默认为第一个.
spring.thymeleaf.view-names
指定使用模板的视图名，多个以逗号分隔.
<h4 id="mustcache">mustcache</h4>
spring.mustache.cache
是否 Enable template caching.
spring.mustache.charset
指定 Template 的编码.
spring.mustache.check-template-location
是否检查默认的路径是否存在.
spring.mustache.content-type
指定 Content-Type.
spring.mustache.enabled
是否开启 mustcache 的模板支持.
spring.mustache.prefix
指定模板的前缀，默认: classpath:/templates/
spring.mustache.suffix
指定模板的后缀，默认: .html
spring.mustache.view-names
指定要使用模板的视图名.
<h4 id="groovy模板">groovy 模板</h4>
spring.groovy.template.allow-request-override
指定 HttpServletRequest 的属性是否可以覆盖 controller 的 model 的同名项
spring.groovy.template.allow-session-override
指定 HttpSession 的属性是否可以覆盖 controller 的 model 的同名项
spring.groovy.template.cache
是否开启模板缓存.

spring.groovy.template.charset
指定 Template 编码.
spring.groovy.template.check-template-location
是否检查模板的路径是否存在.
spring.groovy.template.configuration.auto-escape
是否在渲染模板时自动排查 model 的变量，默认为: false
spring.groovy.template.configuration.auto-indent
是否在渲染模板时自动缩进，默认为 false
spring.groovy.template.configuration.auto-indent-string
如果自动缩进启用的话，是使用 SPACES 还是 TAB，默认为: SPACES
spring.groovy.template.configuration.auto-new-line
渲染模板时是否要输出换行，默认为 false
spring.groovy.template.configuration.base-template-class
指定 template base class.
spring.groovy.template.configuration.cache-templates
是否要缓存模板，默认为 true
spring.groovy.template.configuration.declaration-encoding
在写入 declaration header 时使用的编码
spring.groovy.template.configuration.expand-empty-elements
是使用
这种形式，还是
这种展开模式，默认为: false)
spring.groovy.template.configuration.locale
指定 template locale.
spring.groovy.template.configuration.new-line-string
当启用自动换行时，换行的输出，默认为系统的 line.separator 属性的值
spring.groovy.template.configuration.resource-loader-path
指定 groovy 的模板路径，默认为 classpath:/templates/
spring.groovy.template.configuration.use-double-quotes
指定属性要使用双引号还是单引号，默认为 false
spring.groovy.template.content-type
指定 Content-Type.
spring.groovy.template.enabled
是否开启 groovy 模板的支持.
spring.groovy.template.expose-request-attributes
设定所有 request 的属性在 merge 到模板的时候，是否要都添加到 model 中.
spring.groovy.template.expose-session-attributes
设定所有 request 的属性在 merge 到模板的时候，是否要都添加到 model 中.
spring.groovy.template.expose-spring-macro-helpers
设定是否以 springMacroRequestContext 的形式暴露 RequestContext 给 Spring’s macro library 使用
spring.groovy.template.prefix
指定模板的前缀.
spring.groovy.template.request-context-attribute
指定 RequestContext 属性的名.
spring.groovy.template.resource-loader-path
指定模板的路径，默认为: classpath:/templates/
spring.groovy.template.suffix
指定模板的后缀
spring.groovy.template.view-names
指定要使用模板的视图名称.
<h4 id="http">http</h4>
spring.hateoas.apply-to-primary-object-mapper
设定是否对 object mapper 也支持 HATEOAS，默认为: true

spring.http.converters.preferred-json-mapper
是否优先使用 JSON mapper 来转换.
spring.http.encoding.charset
指定 http 请求和相应的 Charset，默认: UTF-8
spring.http.encoding.enabled
是否开启 http 的编码支持，默认为 true
spring.http.encoding.force
是否强制对 http 请求和响应进行编码，默认为 true
<h4 id="json">json</h4>
spring.jackson.date-format
指定日期格式，比如 yyyy-MM-dd HH:mm:ss，或者具体的格式化类的全限定名
spring.jackson.deserialization
是否开启 Jackson 的反序列化
spring.jackson.generator
是否开启 json 的 generators.
spring.jackson.joda-date-time-format
指定 Joda date/time 的格式，比如 yyyy-MM-dd HH:mm:ss). 如果没有配置的话，dateformat 会为 backup
spring.jackson.locale
指定 json 使用的 Locale.
spring.jackson.mapper
是否开启 Jackson 通用的特性.
spring.jackson.parser
是否开启 jackson 的 parser 特性.
spring.jackson.property-naming-strategy
指定 PropertyNamingStrategy (CAMEL_CASE_TO_LOWER_CASE_WITH_UNDERSCORES)或者指 PropertyNamingStrategy 子类的全限定类名.
spring.jackson.serialization
是否开启 jackson 的序列化.
spring.jackson.serialization-inclusion
指定序列化时属性的 inclusion 方式，具体查看 JsonInclude.Include 枚举.
spring.jackson.time-zone
指定日期格式化时区，比如 America/Los_Angeles 或者 GMT+10.
<h4 id="jersey">jersey</h4>
spring.jersey.filter.order
指定 Jersey filter 的 order，默认为: 0
spring.jersey.init
指定传递给 Jersey 的初始化参数.
spring.jersey.type
指定 Jersey 的集成类型，可以是 servlet 或者 filter.
<h4 id="PROFILES">PROFILES</h4>
spring.profiles.active= # comma list of active profiles
spring.profiles.include= # unconditionally activate the specified comma separated profiles
<h4 id="APPLICATION-SETTINGS--SpringApplication-">APPLICATION SETTINGS (SpringApplication)</h4>
spring.main.sources=
spring.main.web-environment= # detect by default
spring.main.show-banner=true
spring.main....= # see class for all properties
<h4 id="LOGGING">LOGGING</h4>
logging.path=/var/logs
logging.file=myapp.log
logging.config= # location of config file (default classpath:logback.xml for logback)

logging.level.*= # levels for loggers, e.g. “logging.level.org.springframework=DEBUG” (TRA
E, DEBUG, INFO, WARN, ERROR, FATAL, OFF)
<h4 id="IDENTITY--ContextIdApplicationContextInitializer-">IDENTITY (ContextIdApplication
ontextInitializer)</h4>
spring.application.name=
spring.application.index=
<h4 id="EMBEDDED-SERVER-CONFIGURATION--ServerProperties-">EMBEDDED SERVER CO
FIGURATION (ServerProperties)</h4>
server.port=8080
server.address= # bind to a specific NIC
server.session-timeout= # session timeout in seconds
server.context-parameters.*= # Servlet context init parameters, e.g. server.context-parameters
a=alpha
server.context-path= # the context path, defaults to ‘/’
server.servlet-path= # the servlet path, defaults to ‘/’
server.ssl.enabled=true # if SSL support is enabled
server.ssl.client-auth= # want or need
server.ssl.key-alias=
server.ssl.ciphers= # supported SSL ciphers
server.ssl.key-password=
server.ssl.key-store=
server.ssl.key-store-password=
server.ssl.key-store-provider=
server.ssl.key-store-type=
server.ssl.protocol=TLS
server.ssl.trust-store=
server.ssl.trust-store-password=
server.ssl.trust-store-provider=
server.ssl.trust-store-type=
server.tomcat.access-log-pattern= # log pattern of the access log
server.tomcat.access-log-enabled=false # is access logging enabled
server.tomcat.internal-proxies=10\.\d{1,3}\.\d{1,3}\.\d{1,3}|\
192\.168\.\d{1,3}\.\d{1,3}|\
169\.254\.\d{1,3}\.\d{1,3}|\
127\.\d{1,3}\.\d{1,3}\.\d{1,3} # regular expression matching trusted IP addresses
server.tomcat.protocol-header=x-forwarded-proto # front end proxy forward header
server.tomcat.port-header= # front end proxy port header
server.tomcat.remote-ip-header=x-forwarded-for
server.tomcat.basedir=/tmp # base dir (usually not needed, defaults to tmp)
server.tomcat.background-processor-delay=30; # in seconds
server.tomcat.max-http-header-size= # maximum size in bytes of the HTTP message header
br>
server.tomcat.max-threads = 0 # number of threads in protocol handler
server.tomcat.uri-encoding = UTF-8 # character encoding to use for URL decoding
<h4 id="SPRING-MVC--WebMvcProperties-">SPRING MVC (WebMvcProperties)</h4>
spring.mvc.locale= # set fixed locale, e.g. en_UK
spring.mvc.date-format= # set fixed date format, e.g. dd/MM/yyyy
spring.mvc.message-codes-resolver-format= # PREFIX_ERROR_CODE / POSTFIX_ERROR_COD

spring.mvc.ignore-default-model-on-redirect=true # If the the content of the “default” mo
el should be ignored redirects
spring.view.prefix= # MVC view prefix
spring.view.suffix= # ... and suffix
spring.resources.cache-period= # cache timeouts in headers sent to browser

spring.resources.add-mappings=true # if default mappings should be added
<h4 id="SPRING-HATEOS--HateoasProperties-">SPRING HATEOS (HateoasProperties)</h4>
spring.hateoas.apply-to-primary-object-mapper=true # if the primary mapper should als
be configured
<h4 id="HTTP-encoding--HttpEncodingProperties-">HTTP encoding (HttpEncodingPropertie
)</h4>
spring.http.encoding.charset=UTF-8 # the encoding of HTTP requests/responses
spring.http.encoding.enabled=true # enable http encoding support
spring.http.encoding.force=true # force the configured encoding
<h4 id="JACKSON--JacksonProperties-">JACKSON (JacksonProperties)</h4>
spring.jackson.date-format= # Date format string (e.g. yyyy-MM-dd HH:mm:ss), or a fully
qualified date format class name (e.g. com.fasterxml.jackson.databind.util.ISO8601DateFormat

spring.jackson.property-naming-strategy= # One of the constants on Jackson’s PropertyNa
ingStrategy (e.g. CAMEL_CASE_TO_LOWER_CASE_WITH_UNDERSCORES) or the fully-qualified
lass name of a PropertyNamingStrategy subclass
spring.jackson.deserialization.= # see Jackson’s DeserializationFeature
spring.jackson.generator.= # see Jackson’s JsonGenerator.Feature
spring.jackson.mapper.= # see Jackson’s MapperFeature
spring.jackson.parser.= # see Jackson’s JsonParser.Feature
spring.jackson.serialization.*= # see Jackson’s SerializationFeature
<h4 id="THYMELEAF--ThymeleafAutoConfiguration-">THYMELEAF (ThymeleafAutoConfigura
ion)</h4>
spring.thymeleaf.check-template-location=true
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.excluded-view-names= # comma-separated list of view names that should
e excluded from resolution
spring.thymeleaf.view-names= # comma-separated list of view names that can be resolved<b
>
spring.thymeleaf.suffix=.html
spring.thymeleaf.mode=HTML5
spring.thymeleaf.encoding=UTF-8
spring.thymeleaf.content-type=text/html # ;charset= is added
spring.thymeleaf.cache=true # set to false for hot refresh
<h4 id="REEMARKER--FreeMarkerAutoConfiguration-">REEMARKER (FreeMarkerAutoConfig
ration)</h4>
spring.freemarker.allow-request-override=false
spring.freemarker.cache=true
spring.freemarker.check-template-location=true
spring.freemarker.charset=UTF-8
spring.freemarker.content-type=text/html
spring.freemarker.expose-request-attributes=false
spring.freemarker.expose-session-attributes=false
spring.freemarker.expose-spring-macro-helpers=false
spring.freemarker.prefix=
spring.freemarker.request-context-attribute=
spring.freemarker.settings.*=
spring.freemarker.suffix=.ftl
spring.freemarker.template-loader-path=classpath:/templates/ # comma-separated list
spring.freemarker.view-names= # whitelist of view names that can be resolved
<h4 id="GROOVY-TEMPLATES--GroovyTemplateAutoConfiguration-">GROOVY TEMPLATES
GroovyTemplateAutoConfiguration)</h4>
spring.groovy.template.cache=true
spring.groovy.template.charset=UTF-8

spring.groovy.template.configuration.*= # See Groovy’ s TemplateConfiguration
spring.groovy.template.content-type=text/html
spring.groovy.template.prefix=classpath:/templates/
spring.groovy.template.suffix=.tpl
spring.groovy.template.view-names= # whitelist of view names that can be resolved
<h4 id="VELOCITY-TEMPLATES--VelocityAutoConfiguration-">VELOCITY TEMPLATES (Velocit
AutoConfiguration)</h4>
spring.velocity.allow-request-override=false
spring.velocity.cache=true
spring.velocity.check-template-location=true
spring.velocity.charset=UTF-8
spring.velocity.content-type=text/html
spring.velocity.date-tool-attribute=
spring.velocity.expose-request-attributes=false
spring.velocity.expose-session-attributes=false
spring.velocity.expose-spring-macro-helpers=false
spring.velocity.number-tool-attribute=
spring.velocity.prefer-file-system-access=true # prefer file system access for template loadin

spring.velocity.prefix=
spring.velocity.properties.*=
spring.velocity.request-context-attribute=
spring.velocity.resource-loader-path=classpath:/templates/
spring.velocity.suffix=.vm
spring.velocity.toolbox-config-location= # velocity Toolbox config location, for example “/W
B-INF/toolbox.xml”
spring.velocity.view-names= # whitelist of view names that can be resolved
<h4 id="JERSEY--JerseyProperties-">JERSEY (JerseyProperties)</h4>
spring.jersey.type=servlet # servlet or filter
spring.jersey.init= # init params
spring.jersey.filter.order=
<h4 id="INTERNATIONALIZATION--MessageSourceAutoConfiguration-">INTERNATIONALIZ
TION (MessageSourceAutoConfiguration)</h4>
spring.messages.basename=messages
spring.messages.cache-seconds=-1
spring.messages.encoding=UTF-8
<h4 id="SECURITY--SecurityProperties-">SECURITY (SecurityProperties)</h4>
security.user.name=user # login username
security.user.password= # login password
security.user.role=USER # role assigned to the user
security.require-ssl=false # advanced settings ...
security.enable-csrf=false
security.basic.enabled=true
security.basic.realm=Spring
security.basic.path= # /**
security.filter-order=0
security.headers.xss=false
security.headers.cache=false
security.headers.frame=false
security.headers.content-type=false
security.headers.hsts=all # none / domain / all
security.sessions=stateless # always / never / if_required / stateless
security.ignored=false
<h4 id="DATASOURCE--DataSourceAutoConfiguration---DataSourceProperties-">DATASOU

CE (DataSourceAutoConfiguration & DataSourceProperties)</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl">spring.datasource.name= # name of the data source
spring.datasource.
nitialize=true # populate using data.sql
spring.datasource.
chema= # a schema (DDL) script resource reference
spring.datasource
data= # a data (DML) script resource reference
spring.datasource.
ql-script-encoding= # a charset for reading SQL scripts
spring.datasource
platform= # the platform to use in the schema resource (schema-${platform}.sql)
spring.datasource.
ontinue-on-error=false # continue even if can't be initialized
spring.datasource.
eparator=; # statement separator in SQL initialization scripts
spring.datasource
driver-class-name= # JDBC Settings...
spring.datasource
url=
spring.datasource
username=
spring.datasource
password=
spring.datasource.
ndi-name= # For JNDI lookup (class, url, username & password are ignored when set)
spring.datasource
max-active=100 # Advanced configuration...
spring.datasource
max-idle=8
spring.datasource
min-idle=8
spring.datasource.
nitial-size=10
spring.datasource
validation-query=
spring.datasource.
est-on-borrow=false
spring.datasource.
est-on-return=false
spring.datasource.
est-while-idle=
spring.datasource.
ime-between-eviction-runs-millis=
spring.datasource
min-evictable-idle-time-millis=
spring.datasource
max-wait=
spring.datasource.
mx-enabled=false # Export JMX MBeans (if supported)
spring.dao.except
ontranslation.enabled
是否开启Persisten

eExceptionTranslationPostProcessor，默认为true
spring.datasource
abandon-when-percentage-full
设定超时被废弃的
接占到多少比例时要被关闭或上报
spring.datasource
allow-pool-suspension
使用Hikari pool时
是否允许连接池暂停，默认为: false
spring.datasource
alternate-username-allowed
是否允许替代的用
名.
spring.datasource
auto-commit
指定updates是否
动提交.
spring.datasource.
atalog
指定默认的catalog.
spring.datasource.
ommit-on-return
设置当连接被归还
，是否要提交所有还未完成的事务
spring.datasource.
onnection-init-sql
指定连接被创建，
被添加到连接池之前执行的sql.
spring.datasource.
onnection-init-sqls
使用DBCP connect
on pool时，指定初始化时要执行的sql
spring.datasource.
onnection-properties.[key]
在使用DBCP conn
ction pool时指定要配置的属性
spring.datasource.
onnection-test-query
指定校验连接合法
执行的sql语句
spring.datasource.
onnection-timeout
指定连接的超时时
，毫秒单位.
spring.datasource.
ontinue-on-error
在初始化数据库时
遇到错误是否继续，默认false
spring.datasource
data
指定Data (DML)
本
spring.datasource
data-source-class-name

指定数据源的全限
名.
spring.datasource
data-source-jndi
指定jndi的地址
spring.datasource
data-source-properties.[key]
使用Hikari connect
on pool时，指定要设置的属性
spring.datasource
db-properties
使用Tomcat conn
ction pool，指定要设置的属性
spring.datasource
default-auto-commit
是否自动提交.
spring.datasource
default-catalog
指定连接默认的cata
og.
spring.datasource
default-read-only
是否设置默认连接
读.
spring.datasource
default-transaction-isolation
指定连接的事务的
认隔离级别.
spring.datasource
driver-class-name
指定driver的类名
默认从jdbc url中自动探测.
spring.datasource.
air-queue
是否采用FIFO返回
接.
spring.datasource
health-check-properties.[key]
使用Hikari connect
on pool时，在心跳检查时传递的属性
spring.datasource.
dle-timeout
指定连接多久没被
用时，被设置为空闲，默认为10ms
spring.datasource.
gnore-exception-on-pre-load
当初始化连接池时
是否忽略异常.
spring.datasource.
nit-sql
当连接创建时，执
的sql
spring.datasource.
nitial-size

指定启动连接池时初始建立的连接数量
spring.datasource.nitialization-fail-fast
当创建连接池时，法创建指定最小连接数量是否抛异常
spring.datasource.nitialize
指定初始化数据源是否用data.sql来初始化，默认: true
spring.datasource.solate-internal-queries
指定内部查询是否被隔离，默认为false
spring.datasource.dbc-interceptors
使用Tomcat connction pool时，指定jdbc拦截器，分号分隔
spring.datasource.dbc-url
指定JDBC URL.
spring.datasource.mx-enabled
是否开启JMX，默为: false
spring.datasource.ndi-name
指定jndi的名称.
spring.datasource.eak-detection-threshold
使用Hikari connecton pool时，多少毫秒检测一次连接泄露.
spring.datasource.og-abandoned
使用DBCP connecton pool，是否追踪废弃statement或连接，默认为: false
spring.datasource.og-validation-errors
当使用Tomcat conection pool是否打印校验错误.
spring.datasource.ogin-timeout
指定连接数据库的时时间.
spring.datasourcemax-active
指定连接池中最大活跃连接数.
spring.datasourcemax-age
指定连接池中连接最大年龄
<span class="highlight-line"><span class="highlight-cl">spring.datasourcemax-idle

指定连接池最大的闲连接数量.
spring.datasource.max-lifetime
指定连接池中连接最大生存时间，毫秒单位.
spring.datasource.max-open-prepared-statements
指定最大的打开的prepared statements数量.
spring.datasource.max-wait
指定连接池等待连返回的最大等待时间，毫秒单位.
spring.datasource.maximum-pool-size
指定连接池最大的接数，包括使用中的和空闲的连接.
spring.datasource.min-evictable-idle-time-millis
指定一个空闲连接少空闲多久后可被清除.
spring.datasource.min-idle
指定必须保持连接最小值(For DBCP and Tomcat connection pools)
spring.datasource.minimum-idle
指定连接维护的最空闲连接数，当使用HikariCP时指定.
spring.datasource.name
指定数据源名.
spring.datasource.num-tests-per-eviction-run
指定运行每个idle bject evictor线程时的对象数量
spring.datasource.password
指定数据库密码.
spring.datasource.platform
指定schema要使的Platform(schema-${platform}.sql)，默认为: all
spring.datasource.pool-name
指定连接池名字.
spring.datasource.pool-prepared-statements
指定是否池化statements.
spring.datasource.propagate-interrupt-state
在等待连接时，如

线程被中断，是否传播中断状态.
spring.datasource.
ead-only
当使用Hikari conn
ction pool时，是否标记数据源只读
spring.datasource.
egister-mbeans
指定Hikari connect
on pool是否注册JMX MBeans.
spring.datasource.
emove-abandoned
指定当连接超过废
超时时间时，是否立刻删除该连接.
spring.datasource.
emove-abandoned-timeout
指定连接应该被废
的时间.
spring.datasource.
ollback-on-return
在归还连接时，是
回滚等待中的事务.
spring.datasource.
chema
指定Schema (DDL
脚本.
spring.datasource.
eparator
指定初始化脚本的
句分隔符，默认: ;
spring.datasource.
ql-script-encoding
指定SQL scripts
码.
spring.datasource.
uspect-timeout
指定打印废弃连接
的超时时间.
spring.datasource.
est-on-borrow
当从连接池借用连
时，是否测试该连接.
spring.datasource.
est-on-connect
创建时，是否测试
接
spring.datasource.
est-on-return
在连接归还到连接
时是否测试该连接.
spring.datasource.
est-while-idle
当连接空闲时，是
执行连接测试.
spring.datasource.

ime-between-eviction-runs-millis
指定空闲连接检查废弃连接清理、空闲连接池大小调整之间的操作时间间隔
spring.datasource.ransaction-isolation
指定事务隔离级别使用Hikari connection pool时指定
spring.datasource url
指定JDBC URL.
spring.datasource use-disposable-connection-facade
是否对连接进行包，防止连接关闭之后被使用.
spring.datasource use-equals
比较方法名时是否用String.equals()替换==.
spring.datasource use-lock
是否对连接操作加

spring.datasource username
指定数据库名.
spring.datasource validation-interval
指定多少ms执行次连接校验.
spring.datasource validation-query
指定获取连接时连校验的sql查询语句.
spring.datasource validation-query-timeout
指定连接校验查询超时时间.
spring.datasource validation-timeout
设定连接校验的超时间，当使用Hikari connection pool时指定
spring.datasource validator-class-name
用来测试查询的validator全限定名.
spring.datasource.a.data-source-class-name
指定数据源的全限名.
spring.datasource.a.properties
指定传递给XA data source的属性
</code></pre>

<h4 id="DATASOURCE--PersistenceExceptionTranslationAutoConfiguration">DATASOURCE (
ersistenceExceptionTranslationAutoConfiguration</h4>
spring.dao.exceptiontranslation.enabled=true
<h4 id="MONGODB--MongoProperties-">MONGODB (MongoProperties)</h4>
spring.data.mongodb.host= # the db host
spring.data.mongodb.port=27017 # the connection port (defaults to 27107)
spring.data.mongodb.uri= # connection URL
spring.data.mongodb.database=
spring.data.mongodb.authentication-database=
spring.data.mongodb.grid-fs-database=
spring.data.mongodb.username=
spring.data.mongodb.password=
spring.data.mongodb.repositories.enabled=true # if spring data repository support is enable

<h4 id="JPA--JpaBaseConfiguration--HibernateJpaAutoConfiguration-">JPA (JpaBaseConfigu
ation, HibernateJpaAutoConfiguration)</h4>
spring.jpa.properties.*= # properties to set on the JPA connection
spring.jpa.open-in-view=true
spring.jpa.show-sql=true
spring.jpa.database-platform=
spring.jpa.database=
spring.jpa.generate-ddl=false # ignored by Hibernate, might be useful for other vendors
spring.jpa.hibernate.naming-strategy= # naming classname
spring.jpa.hibernate.ddl-auto= # defaults to create-drop for embedded dbs
spring.data.jpa.repositories.enabled=true # if spring data repository support is enabled
<h4 id="JTA--JtaAutoConfiguration-">JTA (JtaAutoConfiguration)</h4>
spring.jta.log-dir= # transaction log dir
spring.jta.*= # technology specific configuration
<h4 id="SOLR--SolrProperties--">SOLR (SolrProperties})</h4>
spring.data.solr.host=<a href="https://ld246.com/forward?goto=http%3A%2F%2F127.0.0
1%3A8983%2Fsolr" target="_blank" rel="nofollow ugc">http://127.0.0.1:8983/solr
spring.data.solr.zk-host=
spring.data.solr.repositories.enabled=true # if spring data repository support is enabled
<h4 id="ELASTICSEARCH--ElasticsearchProperties--">ELASTICSEARCH (ElasticsearchPropertie
})</h4>
spring.data.elasticsearch.cluster-name= # The cluster name (defaults to elasticsearch)<b
>
spring.data.elasticsearch.cluster-nodes= # The address(es) of the server node (comma-separa
ed; if not specified starts a client node)
spring.data.elasticsearch.repositories.enabled=true # if spring data repository support is enab
ed
<h4 id="DATA-RESET--RepositoryRestConfiguration--">DATA RESET (RepositoryRestConfigu
ation})</h4>
spring.data.rest.base-uri= # base URI against which the exporter should calculate its link

<h4 id="FLYWAY--FlywayProperties-">FLYWAY (FlywayProperties)</h4>
flyway.check-location=false # check that migration scripts location exists
flyway.locations=classpath:db/migration # locations of migrations scripts
flyway.schemas= # schemas to update
flyway.init-version= 1 # version to start migration
flyway.init-sqls= # SQL statements to execute to initialize a connection immediately after obta
ning it
flyway.sql-migration-prefix=V
flyway.sql-migration-suffix=.sql

flyway.enabled=true
flyway.url= # JDBC url if you want Flyway to create its own DataSource
flyway.user= # JDBC username if you want Flyway to create its own DataSource
flyway.password= # JDBC password if you want Flyway to create its own DataSource
<h4 id="LIQUIBASE--LiquibaseProperties-">LIQUIBASE (LiquibaseProperties)</h4>
liquibase.change-log=classpath:/db/changelog/db.changelog-master.yaml
liquibase.check-change-log-location=true # check the change log location exists
liquibase.contexts= # runtime contexts to use
liquibase.default-schema= # default database schema to use
liquibase.drop-first=false
liquibase.enabled=true
liquibase.url= # specific JDBC url (if not set the default datasource is used)
liquibase.user= # user name for liquibase.url
liquibase.password= # password for liquibase.url
<h4 id="JMX">JMX</h4>
spring.jmx.enabled=true # Expose MBeans from Spring
<h4 id="RABBIT--RabbitProperties-">RABBIT (RabbitProperties)</h4>
spring.rabbitmq.host= # connection host
spring.rabbitmq.port= # connection port
spring.rabbitmq.addresses= # connection addresses (e.g. myhost:9999,otherhost:1111)
spring.rabbitmq.username= # login user
spring.rabbitmq.password= # login password
spring.rabbitmq.virtual-host=
spring.rabbitmq.dynamic=
<h4 id="REDIS--RedisProperties-">REDIS (RedisProperties)</h4>
spring.redis.database= # database name
spring.redis.host=localhost # server host
spring.redis.password= # server password
spring.redis.port=6379 # connection port
spring.redis.pool.max-idle=8 # pool settings ...
spring.redis.pool.min-idle=0
spring.redis.pool.max-active=8
spring.redis.pool.max-wait=-1
spring.redis.sentinel.master= # name of Redis server
spring.redis.sentinel.nodes= # comma-separated list of host:port pairs
Redis 数据库索引（默认为 0）
-**spring.redis.database=0**
Redis 服务器地址
-**spring.redis.host=192.168.0.58**
Redis 服务器连接端口
-**spring.redis.port=6379**
- Redis 服务器连接密码（默认为空）
-**spring.redis.password=**
连接池最大连接数（使用负值表示没有限制）
-**spring.redis.pool.max-active=8**
连接池最大阻塞等待时间（使用负值表示没有限制）
-**spring.redis.pool.max-wait=-1**
连接池中的最大空闲连接
-**spring.redis.pool.max-idle=8**
连接池中的最小空闲连接
-**spring.redis.pool.min-idle=0**
连接超时时间（毫秒）
-**spring.redis.timeout=0**
<h4 id="ACTIVEMQ--ActiveMQProperties-">ACTIVEMQ (ActiveMQProperties)</h4>

spring.activemq.broker-url= # connection URL
spring.activemq.user=
spring.activemq.password=
spring.activemq.in-memory=true # broker kind to create if no broker-url is specified
spring.activemq.pooled=false
<h4 id="HornetQ--HornetQProperties-">HornetQ (HornetQProperties)</h4>
spring.hornetq.mode= # connection mode (native, embedded)
spring.hornetq.host=localhost # hornetQ host (native mode)
spring.hornetq.port=5445 # hornetQ port (native mode)
spring.hornetq.embedded.enabled=true # if the embedded server is enabled (needs hornetq-ms-server.jar)
spring.hornetq.embedded.server-id= # auto-generated id of the embedded server (integer)<r>
spring.hornetq.embedded.persistent=false # message persistence
spring.hornetq.embedded.data-directory= # location of data content (when persistence is enbled)
spring.hornetq.embedded.queues= # comma-separated queues to create on startup
spring.hornetq.embedded.topics= # comma-separated topics to create on startup
spring.hornetq.embedded.cluster-password= # customer password (randomly generated by efault)
<h4 id="JMS--JmsProperties-">JMS (JmsProperties)</h4>
spring.jms.jndi-name= # JNDI location of a JMS ConnectionFactory
spring.jms.pub-sub-domain= # false for queue (default), true for topic
<h4 id="Email--MailProperties-">Email (MailProperties)</h4>
spring.mail.host=smtp.acme.org # mail server host
spring.mail.port= # mail server port
spring.mail.username=
spring.mail.password=
spring.mail.default-encoding=UTF-8 # encoding to use for MimeMessages
spring.mail.properties.*= # properties to set on the JavaMail session
<h4 id="SPRING-BATCH--BatchDatabaseInitializer-">SPRING BATCH (BatchDatabaseInitialize)</h4>
spring.batch.job.names=job1,job2
spring.batch.job.enabled=true
spring.batch.initializer.enabled=true
spring.batch.schema= # batch schema to load
<h4 id="AOP">AOP</h4>
spring.aop.auto=
spring.aop.proxy-target-class=
<h4 id="FILE-ENCODING--FileEncodingApplicationListener-">FILE ENCODING (FileEncoding pplicationListener)</h4>
spring.mandatory-file-encoding=false
<h4 id="SPRING-SOCIAL--SocialWebAutoConfiguration-">SPRING SOCIAL (SocialWebAutoCnfiguration)</h4>
spring.social.auto-connection-views=true # Set to true for default connection views or fale if you provide your own
<h4 id="SPRING-SOCIAL-FACEBOOK--FacebookAutoConfiguration-">SPRING SOCIAL FACEOOK (FacebookAutoConfiguration)</h4>
spring.social.facebook.app-id= # your application’s Facebook App ID
spring.social.facebook.app-secret= # your application’s Facebook App Secret
<h4 id="SPRING-SOCIAL-LINKEDIN--LinkedInAutoConfiguration-">SPRING SOCIAL LINKEDI (LinkedInAutoConfiguration)</h4>
spring.social.linkedin.app-id= # your application’s LinkedIn App ID
spring.social.linkedin.app-secret= # your application’s LinkedIn App Secret

<h4 id="SPRING-SOCIAL-TWITTER--TwitterAutoConfiguration-">SPRING SOCIAL TWITTER (witterAutoConfiguration)</h4>
spring.social.twitter.app-id= # your application’ s Twitter App ID
spring.social.twitter.app-secret= # your application’ s Twitter App Secret
<h4 id="SPRING-MOBILE-SITE-PREFERENCE--SitePreferenceAutoConfiguration-">SPRING M BILE SITE PREFERENCE (SitePreferenceAutoConfiguration)</h4>
spring.mobile.sitepreference.enabled=true # enabled by default
<h4 id="SPRING-MOBILE-DEVICE-VIEWS--DeviceDelegatingViewResolverAutoConfiguration-">SPRING MOBILE DEVICE VIEWS (DeviceDelegatingViewResolverAutoConfiguration)</h4>
spring.mobile.devicedelegatingviewresolver.enabled=true # disabled by default
spring.mobile.devicedelegatingviewresolver.normal-prefix=
spring.mobile.devicedelegatingviewresolver.normal-suffix=
spring.mobile.devicedelegatingviewresolver.mobile-prefix=mobile/
spring.mobile.devicedelegatingviewresolver.mobile-suffix=
spring.mobile.devicedelegatingviewresolver.tablet-prefix=tablet/
spring.mobile.devicedelegatingviewresolver.tablet-suffix=
<h4 id="MANAGEMENT-HTTP-SERVER--ManagementServerProperties-">MANAGEMENT HT P SERVER (ManagementServerProperties)</h4>
management.port= # defaults to ‘server.port’
management.address= # bind to a specific NIC
management.context-path= # default to ‘/’
management.add-application-context-header= # default to true
management.security.enabled=true # enable security
management.security.role=ADMIN # role required to access the management endpoint
management.security.sessions=stateless # session creating policy to use (always, never, if_req ired, stateless)
<h4 id="PID-FILE--ApplicationPidFileWriter-">PID FILE (ApplicationPidFileWriter)</h4>
spring.pidfile= # Location of the PID file to write
<h4 id="ENDPOINTS--AbstractEndpoint-subclasses-">ENDPOINTS (AbstractEndpoint subclas es)</h4>
endpoints.autoconfig.id=autoconfig
endpoints.autoconfig.sensitive=true
endpoints.autoconfig.enabled=true
endpoints.beans.id=beans
endpoints.beans.sensitive=true
endpoints.beans.enabled=true
endpoints.configprops.id=configprops
endpoints.configprops.sensitive=true
endpoints.configprops.enabled=true
endpoints.configprops.keys-to-sanitize=password,secret,key # suffix or regex
endpoints.dump.id=dump
endpoints.dump.sensitive=true
endpoints.dump.enabled=true
endpoints.env.id=env
endpoints.env.sensitive=true
endpoints.env.enabled=true
endpoints.env.keys-to-sanitize=password,secret,key # suffix or regex
endpoints.health.id=health
endpoints.health.sensitive=true
endpoints.health.enabled=true
endpoints.health.mapping.*= # mapping of health statuses to HttpStatus codes
endpoints.health.time-to-live=1000
endpoints.info.id=info
endpoints.info.sensitive=false

endpoints.info.enabled=true
endpoints.mappings.enabled=true
endpoints.mappings.id=mappings
endpoints.mappings.sensitive=true
endpoints.metrics.id=metrics
endpoints.metrics.sensitive=true
endpoints.metrics.enabled=true
endpoints.shutdown.id=shutdown
endpoints.shutdown.sensitive=true
endpoints.shutdown.enabled=false
endpoints.trace.id=trace
endpoints.trace.sensitive=true
endpoints.trace.enabled=true
<h4 id="HEALTH-INDICATORS--previously-health---">HEALTH INDICATORS (previously heal
h.*)</h4>
management.health.db.enabled=true
management.health.diskspace.enabled=true
management.health.mongo.enabled=true
management.health.rabbit.enabled=true
management.health.redis.enabled=true
management.health.solr.enabled=true
management.health.diskspace.path=.
management.health.diskspace.threshold=10485760
management.health.status.order=DOWN, OUT_OF_SERVICE, UNKNOWN, UP
<h4 id="MVC-ONLY-ENDPOINTS">MVC ONLY ENDPOINTS</h4>
endpoints.jolokia.path=jolokia
endpoints.jolokia.sensitive=true
endpoints.jolokia.enabled=true # when using Jolokia
<h4 id="JMX-ENDPOINT--EndpointMBeanExportProperties-">JMX ENDPOINT (EndpointMBe
nExportProperties)</h4>
endpoints.jmx.enabled=true
endpoints.jmx.domain= # the JMX domain, defaults to ‘org.springboot’
endpoints.jmx.unique-names=false
endpoints.jmx.static-names=
<h4 id="JOLOKIA--JolokiaProperties-">JOLOKIA (JolokiaProperties)</h4>
jolokia.config.*= # See Jolokia manual
<h4 id="REMOTE-SHELL">REMOTE SHELL</h4>
shell.auth=simple # jaas, key, simple, spring
shell.command-refresh-interval=-1
shell.command-path-patterns= # classpath*:/commands/**, classpath*:/crash/commands/**
br>
shell.config-path-patterns= # classpath*:/crash/*
shell.disabled-commands=jpa*,jdbc*,jndi* # comma-separated list of commands to disable<b
>
shell.disabled-plugins=false # don’t expose plugins
shell.ssh.enabled= # ssh settings ...
shell.ssh.key-path=
shell.ssh.port=
shell.telnet.enabled= # telnet settings ...
shell.telnet.port=
shell.auth.jaas.domain= # authentication settings ...
shell.auth.key.path=
shell.auth.simple.user.name=
shell.auth.simple.user.password=

shell.auth.spring.roles=
<h4 id="GIT-INFO">GIT INFO</h4>
spring.git.properties= # resource ref to generated git info properties file
<h4 id="解决SpringBoot连接池TOMCAT-JDBC-默认--DBCP或C3P0连接超时异常问题">解决 SpringBoot 连接池 TOMCAT-JDBC（默认） DBCP 或 C3P0 连接超时异常问题</h4>
配置数据源连接池的时候如果没有配置连接检查或者超时回收，造成之后应用通过连接池 getConnection 时，都可能获取到这些不可用的连接。druid 没有上述问题，因为默认有缺省值。
隔多少时间回收废弃连接 一般比 minEvictableIdleTimeMillis 小
spring.datasource.time-between-eviction-runs-millis=300000
指定运行每个 eviction 线程时的对象数量
spring.datasource.num-tests-per-eviction-run=3
指定一个空闲连接最少空闲多久后可被清除.
spring.datasource.min-evictable-idle-time-millis=3600000
上面的配置是配置数据源回收连接，只要配置的回收时间小于 mysql 的超时时间就可以。
spring.datasource.testOnBorrow=true
spring.datasource.validationQuery=SELECT 1
TestOnBorrow 检测池里连接的可用性，假如连接池中的连接被数据库关闭了，应用通过连接池etConnection 时会重新创建。validationQuery 是检查时用的 sql 语句
SpringBoot1.4.1 需要分别单独配置连接池的配置
比如使用 hikari 连接池
spring.datasource.type=com.zaxxer.hikari.HikariDataSource
spring.datasource.hikari.minimum-idle=5