



链滴

Spring Boot application.properties 配置属性

作者: [zk123](#)

原文链接: <https://ld246.com/article/1502328131816>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

application 的配置属性。

这些属性是否生效取决于对应的组件是否声明为 Spring 应用程序上下文里的 Bean (基本是自配置的), 为一个不生效的组件设置属性是没有用的。

multipart

`multipart.enabled` 开启上传支持 (默认: true)

`multipart.file-size-threshold`: 大于该值的文件会被写到磁盘上

`multipart.location` 上传文件存放位置

`multipart.max-file-size` 最大文件大小

`multipart.max-request-size` 最大请求大小

server

`server.address` 服务器地址

`server.port` 服务器端口

`server.context-parameters.[param name]` 设置 servlet 上下文参数

`server.context-path` 应用上下文路径

Jsp-servlet

`server.jsp-servlet.class-name` 针对 jsp 使用的 Servlet 类名 (默认: org.apache.jasper.servlet.JspServlet)

`server.jsp-servlet.registered` JspServlet 是否要注册到内嵌的 Servlet 容器 (默认 true)

`server.jsp-servlet.init-parameters[param name]` 设置 Jsp Servlet 初始化数

`server.servlet-path` 主分发器 Servlet 的路径 (默认: /)

Session

`server.session.cookie.domain` 会话 Cookie 的域

`server.session.cookie.comment` Cookie 注释

`server.session.cookie.max-age` Cookie 最大保存时间 (单位 s)

`server.session.cookie.name` Cookie 名称

`server.session.cookie.timeout` 超时时间

tomcat

`server.tomcat.accesslog.directory` 创建日志文件的目录

`server.tomcat.accesslog.enabled` 是否开启访问日志 (默认: false)

`server.tomcat.accesslog.pattern` 访问日志的格式 (默认: common)

`server.tomcat.accesslog.prefix` 日志名前缀 (默认: access_log)

`server.tomcat.accesslog.suffix` 日志名后缀 (默认: .log)

`server.tomcat.max-http-header-size` Http 消息头最大字节数 (默认: 0)

`server.tomcat.uri-encoding` 用来解码 URI 的字符编码

Cache

`spring.cache.cache-names` 如果底层缓存管理器支持缓存名的话, 可以在这里指定要创建的缓存名列表, 用逗号分隔。通常这会禁用运行时创建其他额外缓存的能力。

`spring.cache.config` 用来初始化 EhCache 的配置文件位置

`spring.cache.guava.spec` 用来创建缓存 Spec

`spring.cache.hazelcast.config` 用来初始化 Hazelcast 的配置文件位置

`spring.cache.infinispan.config` 用来初始化 Infinispan 配置文件位置

`spring.cache.jcache.config` 用来初始化缓存管理器的配置文件的位置, 配置文件依赖于底层的缓存实现

`spring.cache.jcache.provider` CachingProvider 实现的全限定类名, 用来获 JSR-107 兼容的缓存管理器, 仅在 Classpath 里有不只一个 JSR-107 实现时才需要这个属性。

`spring.cache.type` 缓存类型, 默认根据环境自动检测

Data

`spring.data.jpa.repositories.enabled` 开启 JPA 仓库 (默认: true)

`spring.data.mongodb.authentication-database` 身份认证数据库名

`spring.data.mongodb.database` 数据库名

`spring.data.mongodb.field-naming-strategy` 要使用的 FieldNamingStrate

y 的全限定名。 </p>

<p> <code>spring.data.mongodb.grid.fs.database</code> GridFS 数据库名称 </p>

<p> <code>spring.data.mongodb.host</code> MongoDB 服务器地址 </p>

<p> <code>spring.data.mongodb.username</code> MongoDB 账号 </p>

<p> <code>spring.data.mongodb.password</code> MongoDB 密码 </p>

<p> <code>spring.data.mongodb.port</code> 端口号 </p>

<p> <code>spring.data.mongodb.repositories.enabled</code> 开启 Mongo 仓库 (默认值: true) </p>

<p> <code>spring.data.mongodb.uri</code> Mongo 数据库 URI。设置了该属性后就主机和端口会被忽略。(默认值: localhost/test) </p>

<p> <code>spring.data.rest.base-path</code> 用于发布仓库资源的基本路径 </p>

<p> <code>spring.data.rest.default-page-size</code> 分页数据的默认页大小 (默认: 20) </p>

<p> <code>spring.data.rest.limit-param-name</code> 用于标识一次返回多少记录的 URL 查询字符串参数名。(默认值: size) </p>

<p> <code>spring.data.rest.max-page-size</code>:最大分页大小 (默认: 1000) </p>

<p> <code>spring.data.rest.page-param-name</code> URL 查询字符串参数的名称, 用来标识回哪一页。(默认值: page) </p>

<p> <code>spring.data.rest.return-body-on-create</code> 在创建实体后是否返回一个响应体 (默认: false) </p>

<p> <code>spring.data.rest.return-body-on-update</code> 在更新实体后是否返回一个响应 (默认: false) </p>

<p> <code>spring.data.rest.sort-param-name</code> URL 查询字符串参数的名称, 用来表示果排序的方向 (默认: name) </p>

<p> <code>spring.data.solr.host</code> Solr 的主机地址。如果设置了 zk-host 则忽略该属性 (默认值: http://127.0.0.1 8983/solr) </p>

<p> <code>spring.data.solr.repositories.enabled</code> 开启 solr 仓库 (默认: true) </p>

<p> <code>spring.data.solr.zk-host</code> zk 主机地址, 格式为"主机-端口" </p>

<p> <code>spring.datasource.allow-pool-suspension</code> 是否允许池暂停 (pool suspension)。在开启池暂停后会有性能会受到一定影响, 除非你 真的需要这个功能 (例如在冗余的系统下) 否则不要开启它。该属性只在使用 Hikari 数据库连接池时有用。(默认值: false。) </p>

<h3 id="DataSource">DataSource</h3>

<p> <code>spring.datasource.name</code> 数据源的名称。 </p>

<p> <code>spring.datasource.username</code> 数据库的登录用户名。 </p>

<p> <code>spring.datasource.password</code> 数据库的登录密码。 </p>

<p> <code>spring.datasource.url</code> 数据库的 JDBC URL。 </p>

<p> <code>spring.datasource.jdbc-url</code> 用来创建连接的 JDBC URL。 </p>

<p> <code>spring.datasource.driver-class-name</code> JDBC 驱动的全限定类名。默认根据 RL 自动检测。 </p>

<p> <code>spring.datasource.pool-name</code> 连接池名称。 </p>

<p> <code>spring.datasource.max-active</code> 连接池中的最大活跃连接数。 </p>

<p> <code>spring.datasource.connection-timeout</code> 连接超时 (单位毫秒) </p>

<p> <code>spring.datasource.max-age</code> 连接池中连接的最长寿命。 </p>

<p> <code>spring.datasource.max-idle</code> 连接池中的最大空闲连接数。 </p>

<p> <code>spring.datasource.max-lifetime</code> 连接池中连接的最长寿命 (单位为毫秒) 。 </p>

<p> <code>spring.datasource.max-open-prepared-statements</code> 开启状态的 Prepared statement 的数量上限。 </p>

<p> <code>spring.datasource.max-wait</code> 连接池在等待返回连接时, 最长等待多少毫秒抛出异常。 </p>

<p> <code>spring.datasource.maximum-pool-size</code> 连接池能达到的最大规模, 包含空连接的数量和使用中的连接数量。 </p>

<p> <code>spring.datasource.min-evictable-idle-time-millis</code> 一个空闲连接被空闲连

释放器（如果存在的话）优雅地释放前，最短会在连接池里停留多少时间。 </p>
<p><code>spring.datasource.min-idle</code> 连接池里始终应该保持的最小连接数。（用于 DCP 和 Tomcat 连接池。） </p>
<p><code>spring.datasource.minimum-idle</code>: HikariCP 试图在连接池里维持的最小空闲连接数。 </p>
<p><code>spring.datasource.alternate-username-allowed</code> 是否允许其它用户名 </p>
<p><code>spring.datasource.auto-commit</code> 更新操作是否自动提交 </p>
<p><code>spring.datasource.abandon-when-percentage-full</code> 一个百分比形势的阈值超过该阈值则关闭并报告被弃用的连接 </p>
<p><code>spring.datasource.catalog</code> 默认的 Catalog 名称 </p>
<p><code>spring.datasource.commit-on-return</code> 在连接归还时，连接池是否要提交挂的事务 </p>
<p><code>spring.datasource.connection-init-sql</code> 在所有新连接创建时都会执行的 SQL 语句，该语句会在连接加入连接池前执行。 </p>
<p><code>spring.datasource.connection-init-queries</code> 在物理连接第一次创建时执行的 SQL 语句列表。（用于 DBCP 连接池。） </p>
<p><code>spring.datasource.connection-properties.[key]</code> 设置创建连接时使用的属性。（用于 DBCP 连接池。） </p>
<p><code>spring.datasource.continue-on-error</code> 初始化数据库时发生错误不要终止。默认值： false) </p>
<p><code>spring.datasource.data</code> 指向数据（数据库操纵语言，Data Manipulation Language, DML）脚本资源的引用。 </p>
<p><code>spring.datasource.data-source-class-name</code> 用于获取连接的数据源的全限定类名。 </p>
<p><code>spring.datasource.data-source-jndi</code> 用于获取连接的数据源的 JNDI 位置。 </p>
<p><code>spring.datasource.data-source-properties.[key]</code> 设置创建数据源时使用的属性。（用于 Hikari 连接池。） </p>
<p><code>spring.datasource.db-properties</code> 设置创建数据源时使用的属性。（用于 Tomcat 连接池。） </p>
<p><code>spring.datasource.default-auto-commit</code> 连接上的操作是否自动提交。 </p>
<p><code>spring.datasource.default-catalog</code> 连接的默认 Catalog。 </p>
<p><code>spring.datasource.default-read-only</code> 连接的默认只读状态。 </p>
<p><code>spring.datasource.default-transaction-isolation</code> 连接的默认事务隔离级别 </p>
<p><code>spring.datasource.fair-queue</code> 是否以 FIFO 方式返回连接。 </p>
<p><code>spring.datasource.health-check-properties.[key] </code> 设置要纳入健康检查的属性。（用于 Hikari 连接池。） </p>
<p><code>spring.datasource.idle-timeout</code> 连接池中的连接能保持闲置状态的最长时间单位为毫秒。（默认值： 10。） </p>
<p><code>spring.datasource.ignore-exception-on-pre-load</code> 初始化数据库连接池时是否要忽略连接。 </p>
<p><code>spring.datasource.init-sql </code> 在连接第一次创建时运行的自定义查询。 </p>
<p><code>spring.datasource.initial-size</code> 在连接池启动时要建立的连接数。 </p>
<p><code>spring.datasource.initialization-fail-fast</code> 在连接池创建时，如果达不到最小连接数是否要抛出异常。（默认值： true。） </p>
<p><code>spring.datasource.initialize</code> 使用 data.sql 初始化数据库。（默认值： true） </p>
<p><code>spring.datasource.isolate-internal-queries</code> 是否要隔离内部请求。（默认： false。） </p>
<p><code>spring.datasource.jdbc-interceptors</code> 一个分号分隔的类名列表，这些类都展了 JdbcInterceptor 类。这些拦截器会插入 java.sql.Connection 对象的操作链里。（用于 Tomcat 连接池。） </p>

<p><code>spring.datasource.jmx-enabled</code> 开启 JMX 支持（如果底层连接池提供该功能的话）。（默认值： false 。） </p>
<p><code>spring.datasource.jndi-name</code> 数据源的 JNDI 位置。设置了该属性则忽略类 URL、用户名和密码属性。 </p>
<p><code>spring.datasource.leak-detection-threshold</code> 用来检测 Hikari 连接池连接泄露的阈值，单位为毫秒。 </p>
<p><code>spring.datasource.log-abandoned</code> 是否针对弃用语句或连接的应用程序代记录下跟踪栈。用于 DBCP 连接池。（默认值： false 。） </p>
<p><code>spring.datasource.log-validation-errors </code> 在使用 Tomcat 连接池时是否要记录验证错误。 </p>
<p><code>spring.datasource.login-timeout</code> 连接数据库的超时时间（单位为秒）。 </p>
<p><code>spring.datasource.num-tests-per-eviction-run</code> 空闲对象释放器线程（如果有的话）每次运行时要检查的对象数。 </p>
<p><code>spring.datasource.platform</code> 在 Schema 资源（schema- $\{platform\}$.sql）要使用的平台。（默认值： all 。） </p>
<p><code>spring.datasource.pool-prepared-statements</code> 是否要将 Statement 放在里。 </p>
<p><code>spring.datasource.propagate-interrupt-state</code> 对于等待连接的中断线程，是否要传播中断状态。 </p>
<p><code>spring.datasource.read-only </code> 在使用 Hikari 连接池时将数据源设置为只读。 </p>
<p><code>spring.datasource.register-mbeans</code> Hikari 连接池是否要注册 JMX MBean。 </p>
<p><code>spring.datasource.remove-abandoned</code> 被弃用的连接在到达弃用超时后是应该被移除。 </p>
<p><code>spring.datasource.remove-abandoned-timeout</code> 连接在多少秒后应该考虑用。 </p>
<p><code>spring.datasource.rollback-on-return</code> 在连接归还连接池时，是否要回滚起的事务。 </p>
<p><code>spring.datasource.schema</code> Schema（数据定义语言，Data Definition Language, DDL）脚本资源的引用。 </p>
<p><code>spring.datasource.separator</code> SQL 初始化脚本里的语句分割符。（默认值： ; 。） </p>
<p><code>spring.datasource.sql-script-encoding</code> SQL 脚本的编码。 </p>
<p><code>spring.datasource.suspect-timeout</code> 在记录一个疑似弃用连接前要等待多少。 </p>
<p><code>spring.datasource.test-on-borrow</code> 从连接池中借用连接时是否要进行测试 </p>
<p><code>spring.datasource.test-on-connect</code> 在建立连接时是否要进行测试。 </p>
<p><code>spring.datasource.test-on-return </code> 在将连接归还到连接池时是否要进行测试。 </p>
<p><code>spring.datasource.test-while-idle</code> 在连接空闲时是否要进行测试。 </p>
<p><code>spring.datasource.time-between-eviction-runs-millis</code> 在两次空闲连接验、弃用连接清理和空闲池大小调整之间睡眠的毫秒数。 </p>
<p><code>spring.datasource.transaction-isolation</code> 在使用 Hikari 连接池时设置默认事务隔离级别。 </p>
<p><code>spring.datasource.use-disposable-connection-facade</code> 连接是否要用一个面（facade）封装起来，在调用了 Connection.close() 后就不能再使用这个连接了。 </p>
<p><code>spring.datasource.use-equals </code> 在比较方法名时是否使用 String.equals() 来替 == 。 </p>
<p><code>spring.datasource.use-lock </code> 在操作连接对象时是否要加锁。 </p>
<p><code>spring.datasource.validation-interval </code> 执行连接验证的间隔时间，单位为毫秒。 </p>

<p><code>spring.datasource.validation-query</code> 在连接池里的连接返回给调用者或连接池时，要执行的验证 SQL 查询。</p>
<p><code>spring.datasource.validation-query-timeout</code> 在连接验证查询执行失败前等待的超时时间，单位为秒。</p>
<p><code>spring.datasource.validation-timeout</code> 在连接验证失败前等待的超时时间单位为秒。（用于 Hikari 连接池。）</p>
<p><code>spring.datasource.validator-class-name</code> 可选验证器类的全限定类名，用执行测试查询。</p>
<p><code>spring.datasource.xa.data-source-class-name</code> XA 数据源的全限定类名。</p>
<p><code>spring.datasource.xa.properties</code> 要传递给 XA 数据源的属性。</p>
<h3 id="FreeMarker">FreeMarker</h3>
<p><code>spring.freemarker.allow-request-override</code> HttpServletRequest 的属性是允许覆盖（隐藏）控制器生成的同名模型属性。</p>
<p><code>spring.freemarker.allow-session-override</code> HttpSession 的属性是否允许覆盖（隐藏）控制器生成的同名模型属性。</p>
<p><code>spring.freemarker.cache</code> 开启模板缓存。</p>
<p><code>spring.freemarker.charset</code> 模板编码。</p>
<p><code>spring.freemarker.check-template-location</code> 检查模板位置是否存在。</p>
<p><code>spring.freemarker.content-type</code> Content-Type 的值。</p>
<p><code>spring.freemarker.enabled</code> 开启 FreeMarker 的 MVC 视图解析。</p>
<p><code>spring.freemarker.expose-request-attributes</code> 在模型合并到模板前，是否把所有的请求属性添加到模型里。</p>
<p><code>spring.freemarker.expose-session-attributes</code> 在模型合并到模板前，是否把所有的 HttpSession 属性添加到模型里。</p>
<p><code>spring.freemarker.expose-spring-macro-helpers</code> 是否发布供 Spring 宏程序库使用的 RequestContext，并将命名其名为 springMacro- RequestContext</p>
<p><code>spring.freemarker.prefer-file-system-access</code> 加载模板时优先通过文件系统访问。文件系统访问能够实时检测到模板变更。（默认值： true 。）</p>
<p><code>spring.freemarker.prefix</code> 在构建 URL 时添加到视图名称前的前缀。</p>
<p><code>spring.freemarker.request-context-attribute</code> 在所有视图里使用的 RequestContext 属性的名称。</p>
<p><code>spring.freemarker.settings</code> 要传递给 FreeMarker 配置的各种键。</p>
<p><code>spring.freemarker.suffix</code> 在构建 URL 时添加到视图名称后的后缀。</p>
<p><code>spring.freemarker.template-loader-path</code> 模板路径列表，用逗号分隔。（默认值： ["classpath:/templates/"] 。）</p>
<p><code>spring.freemarker.view-names</code> 可解析的视图名称的白名单。</p>
<h3 id="Groovy">Groovy</h3>
<p><code>spring.groovy.template.allow-request-override</code> HttpServletRequest 的属性是否允许覆盖（隐藏）控制器生成的同名模型属性。</p>
<p><code>spring.groovy.template.allow-session-override</code> HttpSession 的属性是否允许覆盖（隐藏）控制器生成的同名模型属性。</p>
<p><code>spring.groovy.template.cache</code> 开启模板缓存。</p>
<p><code>spring.groovy.template.charset</code> 模板编码。</p>
<p><code>spring.groovy.template.check-template-location</code> 检查模板位置是否存在</p>
<p><code>spring.groovy.template.configuration.auto-escape</code> 模型变量在模板里呈现时是否要做转义。（默认值： false 。）</p>
<p><code>spring.groovy.template.configuration.auto-indent</code> 模板是否要自动呈现进。（默认值： false 。）</p>
<p><code>spring.groovy.template.configuration.auto-indent-string</code> 开启自动缩进用于缩进的字符串，可以是 SPACES，也可以是 TAB。（默认值： SPACES 。）</p>
<p><code>spring.groovy.template.configuration.auto-new-line</code> 模板里是否要呈现的空行。（默认值： false 。）</p>

<p><code>spring.groovy.template.configuration.base-template-class</code> 模板基类。 </p>
>
<p><code>spring.groovy.template.configuration.cache-templates</code> 模板是否应该缓存。 (默认值: true。) </p>
<p><code>spring.groovy.template.configuration.declaration-encoding </code> 用来写声明的编码。 </p>
<p><code>spring.groovy.template.configuration.expand-empty-elements </code> 没有正的元素该用短形式 (例如,) 还是扩展形式 (例如,) 来书写。 (默认值: false) </p>
<p><code>spring.groovy.template.configuration.locale</code> 设置模板地域。 </p>
<p><code>spring.groovy.template.configuration.new-line-string</code> 在自动空行开启后来呈现空行的字符串。 (默认为系统的 line.separator 属性值。) </p>
<p><code>spring.groovy.template.configuration.resource-loader-path</code> Groovy 模的路径。 (默认值: classpath:/templates/。) </p>
<p><code>spring.groovy.template.configuration.use-double-quotes</code> 属性是该用双号还是单引号。 (默认值: false。) </p>
<p><code>spring.groovy.template.content-type</code> Content-Type 的值。 </p>
<p><code>spring.groovy.template.enabled</code> 开启 Groovy 模板的 MVC 视图解析。 </p>
>
<p><code>spring.groovy.template.expose-request-attributes</code> 在模型合并到模板前是否要把所有的请求属性添加到模型里。 </p>
<p><code>spring.groovy.template.expose-session-attributes</code> 在模型合并到模板前是否要把所有的 HttpSession 属性添加到模型里。 </p>
<p><code>spring.groovy.template.expose-spring-macro-helpers</code> 是否发布供 Spring 宏程序库使用的 RequestContext, 并将其命名为 springMacro- RequestContext</p>
<p><code>spring.groovy.template.prefix</code> 在构建 URL 时, 添加到视图名称前的前缀。 </p>
<p><code>spring.groovy.template.request-context-attribute </code> 所有视图里使用的 RequestContext 属性的名称。 </p>
<p><code>spring.groovy.template.resource-loader-path</code> 模板路径 (默认值: classpath:/templates/。) </p>
<p><code>spring.groovy.template.suffix</code> 在构建 URL 时, 添加到视图名称后的后缀。 </p>
<p><code>spring.groovy.template.view-names</code> 可解析的视图名称白名单。 </p>
<h3 id="H2">H2</h3>
<p><code>spring.h2.console.enabled</code> 开启控制台。 (默认值: false。) </p>
<p><code>spring.h2.console.path</code> 可以找到控制台的路径。 (默认值: /h2-console) </p>
<h3 id="Hornetq">Hornetq</h3>
<p><code>spring.hornetq.embedded.cluster-password</code> 集群密码。默认在启动时随生成。 </p>
<p><code>spring.hornetq.embedded.data-directory</code> 日志文件目录。如果关闭了持久功能则不需要该属性。 </p>
<p><code>spring.hornetq.embedded.enabled</code> 如果有 HornetQ 服务器 API, 则开启入模式。 (默认值: true。) </p>
<p><code>spring.hornetq.embedded.persistent</code> 开启持久化存储。 (默认值: false) </p>
<p><code>spring.hornetq.embedded.queues</code> 启动时要创建的队列列表, 用逗号分隔 (默认值: []。) </p>
<p><code>spring.hornetq.embedded.server-id</code> 服务器 ID。默认使用自增长计数器。默认值: 0。) </p>
<p><code>spring.hornetq.embedded.topics</code> 启动时要创建的主题列表, 用逗号分隔 (默认值: []。) </p>
<p><code>spring.hornetq.host</code> HornetQ 的主机。 (默认值: localhost。) </p>
<p><code>spring.hornetq.mode</code> HornetQ 的部署模式, 默认为自动检测。可以显式地

置为 native 或 embedded 。 </p>

<p><code>spring.hornetq.port</code> HornetQ 的端口。 (默认值: 5445 。) </p>

<h3 id="Http">Http</h3>

<p><code>spring.http.converters.preferred-json-mapper</code> HTTP 消息转换时优先使用 SON 映射器。 </p>

<p><code>spring.http.encoding.charset</code> HTTP 请求和响应的字符集。如果没有显式地定 Content-Type 头, 则将该属性值作为 这个头的值。 (默认值: UTF-8 。) </p>

<p><code>spring.http.encoding.enabled </code> 开启 HTTP 编码支持。 (默认值: true 。) </p>

<p><code>spring.http.encoding.force</code> 强制将 HTTP 请求和响应编码为所配置的字符。 (默认值: true 。) </p>

<h3 id="Jackson">Jackson</h3>

<p><code>spring.jackson.date-format</code> 日期格式字符串 (yyyy-MM-dd HH:mm:ss) 日期格式类的全限定类名。 </p>

<p><code>spring.jackson.deserialization</code> 影响 Java 对象反序列化的 Jackson on/off 性。 </p>

<p><code>spring.jackson.generator</code> 用于生成器的 Jackson on/off 特性。 </p>

<p><code>spring.jackson.joda-date-time-format</code> Joda 日期时间格式字符串 (yyyy-M-dd HH:mm:ss) 。如果没有配置, 而 date-format 又配置了一个格式字符串的话, 会将它作为降配置。 </p>

<p><code>spring.jackson.local</code> e 用于格式化的地域值。 </p>

<p><code>spring.jackson.mapper</code> Jackson 的通用 on/off 特性。 </p>

<p><code>spring.jackson.parser</code> 用于解析器的 Jackson on/off 特性。 </p>

<p><code>spring.jackson.property-naming-strategy</code> Jackson 的 PropertyNamingStrategy 中的一个常量 (CAMEL_CASE_TO_LOWER_CASE_WITH_UNDERSCORES) 。也可以设置 PropertyNamingStrategy 的子类的全限定类名。 </p>

<p><code>spring.jackson.serialization</code> 影响 Java 对象序列化的 Jackson on/off 特性。 </p>

<p><code>spring.jackson.serialization-inclusion</code> 控制序列化时要包含哪些属性。可选择 Jackson 的 JsonInclude.Include 枚举里的某个值。 </p>

<p><code>spring.jackson.time-zone</code> 格式化日期时使用的时区。可以配置各种可识别时区标识符, 比如 America/Los_Angeles 或者 GMT+10 。 </p>

<h3 id="Jersey">Jersey</h3>

<p><code>spring.jersey.filter.order</code> Jersey 过滤器链的顺序。 (默认值: 0 。) </p>

<p><code>spring.jersey.init</code> 通过 Servlet 或过滤器传递给 Jersey 的初始化参数。 </p>

<p><code>spring.jersey.type</code> Jersey 集成类型。可以是 servlet 或者 filter 。 </p>

<h3 id="Jms">Jms</h3>

<p><code>spring.jms.jndi-name</code> 连接工厂的 JNDI 名字。设置了该属性, 则优先于其自动配置的连接工厂。 </p>

<p><code>spring.jms.listener.acknowledge-mode</code> 容器的应答模式 (acknowledgment mode) 。默认情况下, 监听器使用自动应答。 </p>

<p><code>spring.jms.listener.auto-startup</code> 启动时自动启动容器。 (默认值: true 。) </p>

<p><code>spring.jms.listener.concurrency</code> 并发消费者的数量下限。 </p>

<p><code>spring.jms.listener.max-concurrency </code> 并发消费者的数量上限。 </p>

<p><code>spring.jms.pub-sub-domain</code> 如果是主题而非队列, 指明默认的目的地类型否支持 Pub/Sub。 (默认值: false 。) </p>

<h3 id="Jmx">Jmx</h3>

<p><code>spring.jmx.default-domain</code> JMX 域名。 </p>

<p><code>spring.jmx.enabled</code> 将管理 Bean 发布到 JMX 域里。 (默认值: true 。) </p>

<p><code>spring.jmx.server</code> MBeanServer 的 Bean 名称。 (默认值: mbeanServer) </p>

<h3 id="Jpa">Jpa</h3>

<p><code>spring.jpa.database</code> 要操作的目标数据库，默认自动检测。也可以通过 data asePlatform 属性进行设置。 </p>
<p><code>spring.jpa.database-platform</code> 要操作的目标数据库，默认自动检测。也可通过 Database 枚举来设置。 </p>
<p><code>spring.jpa.generate-ddl</code> 启动时要初始化 Schema。（默认值： false。） <p>
<p><code>spring.jpa.hibernate.ddl-auto</code> DDL 模式（ none、 validate、 update、 c eate 和 create-drop）。这是 hibernate. hbm2ddl.auto 属性的一个快捷方式。在使用嵌入式数据时，默认为 create-drop,其他情况下默认为 none。 </p>
<p><code>spring.jpa.hibernate.naming-strategy</code> Hibernate 命名策略的全限定类名。 </p>
<p><code>spring.jpa.open-in-view</code> 注册 OpenEntityManagerInViewInterceptor，请求的整个处理过程中，将一个 JPA EntityManager 绑定到线程上。（默认值： true） </p>
<p><code>spring.jpa.properties</code> JPA 提供方要设置的额外原生属性。 </p>
<p><code>spring.jpa.show-sql</code> 在使用 Bitronix Transaction Manager 时打开 SQL 语 日志。（默认值： false。） </p>
<h3 id="Jta">Jta</h3>
<p><code>spring.jta.allow-multiple-lrc</code> 在使用 Bitronix Transaction Manager 时，事 务管理器是否应该允许一个事务涉及多个 LRC 资源。（默认值： false） </p>
<p><code>spring.jta.asynchronous2-pc</code> 在使用 Bitronix Transaction Manager 时， 否异步执行两阶段提交。（默认值： false。） </p>
<p><code>spring.jta.background-recovery-interval</code> 在使用 Bitronix Transaction Ma gager 时，多久运行一次恢复过程，单位为分钟。（默认值： 1） </p>
<p><code>spring.jta.background-recovery-interval-seconds</code> 在使用 Bitronix Transact on Manager 时，多久运行一次恢复过程，单位为秒。（默认值： 60） </p>
<p><code>spring.jta.current-node-only-recovery</code> 在使用 Bitronix Transaction Mana er 时，恢复是否要滤除不包含本 JVM 唯一 ID 的 XID。（默认值： true） </p>
<p><code>spring.jta.debug-zero-resource-transaction</code> 在使用 Bitronix Transaction anager 时，对于没有涉及任何资源的事务，是否要跟踪并记 录它们的创建和提交调用栈。（默认值： false） </p>
<p><code>spring.jta.default-transaction-timeout</code> 在使用 Bitronix Transaction Mana er 时，默认的事务超时时间，单位为秒。（默认值： 60。） </p>
<p><code>spring.jta.disable-jmx</code> 在使用 Bitronix Transaction Manager 时，是否要禁 注册 JMX MBean。（默认值： false。） </p>
<p><code>spring.jta.enabled</code> 开启 JTA 支持。（默认值： true。） </p>
<p><code>spring.jta.exception-analyzer</code> 在使用 Bitronix Transaction Manager 时用的 异常分析器。设置为 null 时使用默认异常分析器，也可以设置自定义异常分析器的全限定类名。 </ >
<p><code>spring.jta.filter-log-status</code> 在使用 Bitronix Transaction Manager 时，是 只记录必要的日志。开启该参数时能减少分段（fragment）空间用量，但调试更复杂了。（默认值： else） </p>
<p><code>spring.jta.force-batching-enabled</code> 在使用 Bitronix Transaction Manager 时， 是否批量输出至磁盘。禁用批处理会严重降低事 务管理器的吞吐量。（默认值： true。） </p>
<p><code>spring.jta.forced-write-enabled</code> 在使用 Bitronix Transaction Manager 时 日志是否强制写到磁盘上。在生产环境里不要设 置为 false，因为不强制写到磁盘上无法保证完整性 （默认值： true） </p>
<p><code>spring.jta.graceful-shutdown-interval</code> 在使用 Bitronix Transaction Mana er 时，要关闭的话，事务管理器在放弃事务前最多等它 多少秒。（默认值： 60） </p>
<p><code>spring.jta.jndi-transaction-synchronization-registry-name</code> 在使用 Bitroni Transaction Manager 时，事务同步注册表应该绑定到哪个 JNDI 下。（默认 值： java:comp/Tran actionSynchronizationRegistry） </p>
<p><code>spring.jta.jndi-user-transaction-name</code> 在使用 Bitronix Transaction Manag er 时，用户事务应该绑定到哪个 JNDI 下。（默认值： java:comp/UserTransaction。） </p>
<p><code>spring.jta.journal</code> 在使用 Bitronix Transaction Manager 时，要用的日志名

可以是 disk、null 或者全限定类名。(默认值: disk。)

`spring.jta.log-dir` 事务日志目录。

`spring.jta.log-part1-filename` 日志分段文件 1 的名称。(默认值: btm1.log。)

`spring.jta.log-part2-filename` 日志分段文件 2 的名称。(默认值: btm2.log。)

`spring.jta.max-log-size-in-mb` 在使用 Bitronix Transaction Manager 时, 日志分段文件的最大兆数。日志越大, 事务就被允许在未终结状态停留越长时间。但是, 如果文件大小限制得太小, 事务管理器在分段满了的时候就会暂停更长时间。(默认值: 2。)

`spring.jta.resource-configuration-filename` Bitronix Transaction Manager 的配置文件名。

`spring.jta.server-id` 唯一标识 Bitronix Transaction Manager 实例的 ID。

`spring.jta.skip-corrupted-logs` 是否跳过损坏的日志文件。(默认值: false。)

`spring.jta.transaction-manager-id` 事务管理器的唯一标识符。

`spring.jta.warn-about-zero-resource-transaction` 在使用 Bitronix Transaction Manager 时, 是否要对执行时没有涉及任何资源的事务作出告警。(默认值: true。)

Mail

`spring.mail.default-encoding` 默认的 MimeMessage 编码。(默认值: UTF-8。)

`spring.mail.host` SMTP 服务器主机地址。

`spring.mail.jndi-name` 会话的 JNDI 名称。设置之后, 该属性的优先级要高于其他邮件设置。

`spring.mail.password` SMTP 服务器的登录密码。

`spring.mail.port` SMTP 服务器的端口号。

`spring.mail.properties` 附加的 JavaMail 会话属性。

`spring.mail.protocol` SMTP 服务器用到的协议。(默认值: smtp。)

`spring.mail.test-connection` 在启动时测试邮件服务器是否可用。(默认值: false。)

`spring.mail.username` SMTP 服务器的登录用户名。

Messages

`spring.messages.basename` 逗号分隔的基本名称列表, 都遵循 ResourceBundle 的惯例。本质上这就是一个全限定的 Classpath 位置, 如果不包含包限定符(比如 org.mypackage), 就会从 Classpath 的根部开始解析。(默认值: messages。)

`spring.messages.cache-seconds` 加载的资源包文件的缓存失效时间, 单位秒。在设置为 -1 时, 包会永远缓存。(默认值: -1。)

`spring.messages.encoding` 消息包的编码。(默认值: UTF-8。)

Mobile

`spring.mobile.delegatingviewresolver.enable-fallback` 开启降级解支持。(默认值: false。)

`spring.mobile.delegatingviewresolver.enabled` 开启设备视图解析。(默认值: false。)

`spring.mobile.delegatingviewresolver.mobile-prefix` 添加到移动设备视图名前的前缀。(默认值: mobile/。)

`spring.mobile.delegatingviewresolver.mobile-suffix` 添加到移动设备视图名后的后缀。

`spring.mobile.delegatingviewresolver.normal-prefix` 添加到普通设备视图名前的前缀。

`spring.mobile.delegatingviewresolver.normal-suffix` 添加到普通设备视图名后的后缀。

`spring.mobile.delegatingviewresolver.tablet-prefix` 添加到平板设备视图名前的前缀。(默认值: tablet/。)

`spring.mobile.delegatingviewresolver.tablet-suffix` 添加到平板设备

视图名后的后缀。 </p>

<p> <code>spring.mobile.sitepreference.enabled</code> 开启 SitePreferenceHandler 。 (认值: true 。) </p>

<h3 id="Mongodb"> Mongodb </h3>

<p> <code>spring.mongodb.embedded.features </code> 要开启的特性列表, 用逗号分隔。 </p>

<p> <code>spring.mongodb.embedded.version</code> 要使用的 Mongo 版本。 (默认值: 2 6.10 。) </p>

<h3 id="Mustache"> Mustache </h3>

<p> <code>spring.mustache.cache</code> 开启模板缓存。 </p>

<p> <code>spring.mustache.charset</code> 模板编码。 </p>

<p> <code>spring.mustache.check-template-location</code> 检查模板位置是否存在。 </p>

<p> <code>spring.mustache.content-type</code> Content-Type 的值。 </p>

<p> <code>spring.mustache.enabled</code> 开启 Mustache 的 MVC 视图解析。 </p>

<p> <code>spring.mustache.prefix </code> 添加到模板名前的前缀。 (默认值: classpath:/ templates/ 。) </p>

<p> <code>spring.mustache.suffix</code> 添加到模板名后的后缀。 (默认值: .html 。) </p>

<p> <code>spring.mustache.view-names</code> 可解析的视图名称的白名单。 </p>

<h3 id="Mvc"> Mvc </h3>

<p> <code>spring.mvc.async.request-timeout</code> 异步请求处理超时前的等待时间 (单位毫秒) 。如果没有设置该属性, 则使用底层实现的默认超时时间, 比如, Tomcat 上使用 Servlet 3 超时时间为 10 秒。 </p>

<p> <code>spring.mvc.date-format</code> 要使用的日期格式 (比如 dd/MM/yyyy) 。 </p>

<p> <code>spring.mvc.favicon.enabled</code> 开启 favicon.ico 的解析。 (默认值: true 。 </p>

<p> <code>spring.mvc.ignore-default-model-on-redirect</code> 在重定向的场景下, 是否要略 " 默认 " 模型对象的内容。 (默认值: true 。) </p>

<p> <code>spring.mvc.locale</code> 要使用的地域配置。 </p>

<p> <code>spring.mvc.message-codes-resolver-format</code> 消息代码格式 (PREFIX_ERR R_CODE 、 POSTFIX_ERROR_CODE) 。 </p>

<p> <code>spring.mvc.view.prefix</code> Spring MVC 视图前缀。 </p>

<p> <code>spring.mvc.view.suffix</code> Spring MVC 视图后缀。 </p>

<h3 id="Mybatis"> Mybatis </h3>

<p> <code>mybatis.mapper-locations</code> mybatis 映射文件位置。

<code>mybatis.type-aliases-package</code> 别名包位置。 </p>

<h3 id="Rabbitmq"> Rabbitmq </h3>

<p> <code>spring.rabbitmq.addresses</code> 客户端应该连接的地址列表, 用逗号分隔。 </p>

<p> <code>spring.rabbitmq.dynamic</code> 创建一个 AmqpAdmin Bean。 (默认值: true) </p>

<p> <code>spring.rabbitmq.host</code> RabbitMQ 主机地址。 (默认值: localhost 。) </p>

<p> <code>spring.rabbitmq.listener.acknowledge-mode</code> 容器的应答模式。 </p>

<p> <code>spring.rabbitmq.listener.auto-startup</code> 启动时自动开启容器。 (默认值: true 。) </p>

<p> <code>spring.rabbitmq.listener.concurrency</code> 消费者的数量下限。 </p>

<p> <code>spring.rabbitmq.listener.max-concurrency</code> 消费者的数量上限。 </p>

<p> <code>spring.rabbitmq.listener.prefetch</code> 单个请求里要处理的消息数。该数值不应于事务数 (如果用到的话) 。 </p>

<p> <code>spring.rabbitmq.listener.transaction-size</code> 一个事务里要处理的消息数。为保证效果, 应该不大于预先获取的数量。 </p>

<p> <code>spring.rabbitmq.password</code> 进行身份验证的密码。 </p>

<p> <code>spring.rabbitmq.port</code> RabbitMQ 端口。 (默认值: 5672 。) </p>

<p> <code>spring.rabbitmq.requested-heartbeat</code> 请求心跳超时, 单位为秒; 0 表示启用心跳。 </p>

<p><code>spring.rabbitmq.ssl.enabled</code> 开启 SSL 支持。（默认值： false 。） </p>
<p><code>spring.rabbitmq.ssl.key-store</code> 持有 SSL 证书的 KeyStore 路径。 </p>
<p><code>spring.rabbitmq.ssl.key-store-password</code> 访问 KeyStore 的密码。 </p>
<p><code>spring.rabbitmq.ssl.trust-store</code> 持有 SSL 证书的 TrustStore。 </p>
<p><code>spring.rabbitmq.ssl.trust-store-password</code> 访问 TrustStore 的密码。 </p>
<p><code>spring.rabbitmq.username</code> 进行身份验证的用户名。 </p>
<p><code>spring.rabbitmq.virtual-host</code> 在连接 RabbitMQ 时的虚拟主机。 </p>
<h3 id="Redis">Redis</h3>
<p><code>spring.redis.database</code> 连接工厂使用的数据库索引。（默认值： 0 。） </p>
<p><code>spring.redis.host</code> Redis 服务器主机地址。（默认值： localhost 。） </p>
<p><code>spring.redis.password </code> Redis 服务器的登录密码。 </p>
<p><code>spring.redis.pool.max-active </code> 连接池在指定时间里能分配的最大连接数。负表示无限制。（默认值： 8 。） </p>
<p><code>spring.redis.pool.max-idle</code> 连接池里的最大空闲连接数。负数表示空闲连接可以是无限大。（默认值： 8 。） </p>
<p><code>spring.redis.pool.max-wait </code> 当连接池被耗尽时，分配连接的请求应该在抛异常前被阻塞多长时间（单位为秒）。负数表示一直阻塞。（默认值： -1 。） </p>
<p><code>spring.redis.pool.min-idle</code> 连接池里要维持的最小空闲连接数。该属性只有设置为正数时才有效。（默认值： 0 。） </p>
<p><code>spring.redis.port</code> Redis 服务器端口。（默认值： 6379 。） </p>
<p><code>spring.redis.sentinel.master</code> Redis 服务器的名字。 </p>
<p><code>spring.redis.sentinel.nodes </code> 形如“主机:端口”配对的列表，用逗号分隔。 <p>
<p><code>spring.redis.timeout</code> 连接超时时间，单位为秒。（默认值： 0 。） </p>
<h3 id="Resources">Resources</h3>
<p><code>spring.resources.add-mappings</code> 开启默认资源处理。（默认值： true 。） <p>
<p><code>spring.resources.cache-period</code> 资源处理器对资源的缓存周期，单位为秒。 <p>
<p><code>spring.resources.chain.cache</code> 对资源链开启缓存。（默认值： true 。） </p>
<p><code>spring.resources.chain.enabled</code> 开启 Spring 资源处理链。（默认关闭的，非至少开启了一个策略。） </p>
<p><code>spring.resources.chain.html-application-cache</code> 开启 HTML5 应用程序缓证明重写。（默认值： false 。） </p>
<p><code>spring.resources.chain.strategy.content.enabled</code> 开启内容版本策略。（默认值： false 。） </p>
<p><code>spring.resources.chain.strategy.content.paths</code> 要运用于版本策略的模式列，用逗号分隔。（默认值： [/**] 。） </p>
<p><code>spring.resources.chain.strategy.fixed.enabled </code> 开启固定版本策略。（默认： false 。） </p>
<p><code>spring.resources.chain.strategy.fixed.paths </code> 要运用于固定版本策略的模式表，用逗号分隔。 </p>
<p><code>spring.resources.chain.strategy.fixed.version</code> 用于固定版本策略的版本字符串。 </p>
<p><code>spring.resources.static-locations</code> 静态资源位置。默认为 classpath: [/META-INF/resources/, /resources/, /static/, /public/] 加上 context:/ (Servlet 上下文的根目录)。 </p>
<h3 id="SendGrid">SendGrid</h3>
<p><code>spring.sendgrid.password</code> SendGrid 密码。 </p>
<p><code>spring.sendgrid.proxy.host</code> SendGrid 代理主机地址。 </p>
<p><code>spring.sendgrid.proxy.port</code> SendGrid 代理端口。 </p>
<p><code>spring.sendgrid.username</code> SendGrid 用户名。 </p>
<h3 id="Social">Social</h3>
<p><code>spring.social.auto-connection-views</code> 针对所支持的提供方开启连接状态视

。 (默认值: false 。) </p>

<p> <code>spring.social.facebook.app-id</code> 应用程序 ID。 </p>

<p> <code>spring.social.facebook.app-secret</code> 应用程序的密钥。 </p>

<p> <code>spring.social.linkedin.app-id</code> 应用程序 ID。 </p>

<p> <code>spring.social.linkedin.app-secret</code> 应用程序的密钥。 </p>

<p> <code>spring.social.twitter.app-id</code> 应用程序 ID。 </p>

<p> <code>spring.social.twitter.app-secret</code> 应用程序的密钥。 </p>

<h3 id="Thymeleaf"> Thymeleaf </h3>

<p> <code>spring.thymeleaf.cache</code> 开启模板缓存。 (默认值: true 。) </p>

<p> <code>spring.thymeleaf.check-template-location</code> 检查模板位置是否存在。 (默认值: true 。) </p>

<p> <code>spring.thymeleaf.content-type</code> Content-Type 的值。 (默认值: text/html 。) </p>

<p> <code>spring.thymeleaf.enabled</code> 开启 MVC Thymeleaf 视图解析。 (默认值: true 。) </p>

<p> <code>spring.thymeleaf.encoding</code> 模板编码。 (默认值: UTF-8 。) </p>

<p> <code>spring.thymeleaf.excluded-view-names</code> 要被排除在解析之外的视图名称列表, 用逗号分隔。 </p>

<p> <code>spring.thymeleaf.mode</code> 要运用于模板之上的模板模式。另见 StandardTemplateModeHandlers 。 (默认值: HTML5 。) </p>

<p> <code>spring.thymeleaf.prefix</code> 在构建 URL 时添加到视图名称前的前缀。 (默认值: classpath:/templates/ 。) </p>

<p> <code>spring.thymeleaf.suffix</code> 在构建 URL 时添加到视图名称后的后缀。 (默认值: html 。) </p>

<p> <code>spring.thymeleaf.template-resolver-order</code> Thymeleaf 模板解析器在解析链中的顺序。默认情况下, 它排在第一位。顺序从 1 开始只有在定义了额外的 TemplateResolver Bean 时才需要设置这个属性。 </p>

<p> <code>spring.thymeleaf.view-names</code> 可解析的视图名称列表, 用逗号分隔。 </p>

<h3 id="Velocity"> Velocity </h3>

<p> <code>spring.velocity.allow-request-override</code> HttpServletRequest 的属性是否允许覆盖 (隐藏) 控制器生成的同名模型属性。 </p>

<p> <code>spring.velocity.allow-session-override</code> HttpSession 的属性是否允许覆盖 (隐藏) 控制器生成的同名模型属性。 </p>

<p> <code>spring.velocity.cache</code> 开启模板缓存。 </p>

<p> <code>spring.velocity.charset</code> 模板编码。 </p>

<p> <code>spring.velocity.check-template-location</code> 检查模板位置是否存在。 </p>

<p> <code>spring.velocity.content-type</code> Content-Type 的值。 </p>

<p> <code>spring.velocity.date-tool-attribute</code> DateTool 辅助对象在视图的 Velocity 上下文里呈现的名字。 </p>

<p> <code>spring.velocity.enabled</code> 开启 Velocity 的 MVC 视图解析。 </p>

<p> <code>spring.velocity.expose-request-attributes</code> 在模型合并到模板前, 是否要把有的请求属性添加到模型里。 </p>

<p> <code>spring.velocity.expose-session-attributes</code> 在模型合并到模板前, 是否要把有的 HttpSession 属性添加到模型里。 </p>

<p> <code>spring.velocity.expose-spring-macro-helpers</code> 是否发布供 Spring 宏程序使用的 RequestContext, 并将其命名为 springMacro- RequestContext 。 </p>

<p> <code>spring.velocity.number-tool-attribute</code> NumberTool 辅助对象在视图的 Velocity 上下文里呈现的名字。 </p>

<p> <code>spring.velocity.prefer-file-system-access</code> 加载模板时优先通过文件系统访问。文件系统访问能够实时检测到模板变更。 (默认值: true 。) </p>

<p> <code>spring.velocity.prefix</code> 在构建 URL 时添加到视图名称前的前缀。 </p>

<p> <code>spring.velocity.properties</code> 额外的 Velocity 属性。 </p>

<p> <code>spring.velocity.request-context-attribute</code> 所有视图里使用的 RequestContext 属性的名称。 </p>

<p><code>spring.velocity.resource-loader-path</code> 模板路径。（默认值： classpath:/ templates/。） </p>
<p><code>spring.velocity.suffix</code> 在构建 URL 时添加到视图名称后的后缀。 </p>
<p><code>spring.velocity.toolbox-config-location</code> Velocity Toolbox 的配置位置，比
/WEB-INF/toolbox.xml。自动加载 Velocity Tools 工具定义文件，将所定义的全部工具发布到指定
作用域内。 </p>
<p><code>spring.velocity.view-names </code> 可解析的视图名称白名单。 </p>
<h3 id="View">View</h3>
<p><code>spring.view.prefix</code> Spring MVC 视图前缀。 </p>
<p><code>spring.view.suffix </code> Spring MVC 视图后缀。 </p>
<h3 id="其它">其它</h3>
<p><code>spring.aop.auto</code> 添加 @EnableAspectJAutoProxy(默认： true)</p>
<p><code>spring.application.admin.enabled </code> 开启应用程序的管理功能(默认： false)</
>
<p><code>spring.artemis.embedded.cluster-password</code> 集群密码。默认在启东市随机
成</p>
<p><code>spring.artemis.embedded.persistent </code> 开启持久化存储(默认： false)</p>
<p><code>spring.autoconfigure.exclude</code> 要排除的自动配置类</p>