



链滴

Spring boot配置文件 application.properties

作者: [xiaoD](#)

原文链接: <https://ld246.com/article/1463637982614>

来源网站: [链滴](#)

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

```

<pre class="prettyprint cs"><span class="preprocessor"># =====
=====
</span>
<span class="preprocessor"># COMMON SPRING BOOT PROPERTIES</span>
<span class="preprocessor">#</span>
<span class="preprocessor"># This sample file is provided as a guideline. Do NOT copy it in i
s</span>
<span class="preprocessor"># entirety to your own application.      ^^^</span>
<span class="preprocessor"># =====
=====
</span>

<span class="preprocessor"># -----</span>
<span class="preprocessor"># CORE PROPERTIES</span>
<span class="preprocessor"># -----</span>

<span class="preprocessor"># SPRING CONFIG (ConfigFileApplicationListener)</span>
spring.config.name= <span class="preprocessor"># config file name (default to 'application'
</span>
</span>
spring.config.location= <span class="preprocessor"># location of config file</span>

<span class="preprocessor"># PROFILES</span>
spring.profiles= <span class="preprocessor"># comma list of active profiles</span>

<span class="preprocessor"># APPLICATION SETTINGS (SpringApplication)</span>
spring.main.sources=
spring.main.web-environment= <span class="preprocessor"># detect by default</span>
spring.main.show-banner= <span class="keyword">>true</span>
spring.main....= <span class="preprocessor"># see class for all properties</span>

<span class="preprocessor"># LOGGING</span>
logging.path=/<span class="keyword">var</span>/logs
logging.file=myapp.log
logging.config=

<span class="preprocessor"># IDENTITY (ContextIdApplicationContextInitializer)</span>
spring.application.name=
spring.application.index=

<span class="preprocessor"># EMBEDDED SERVER CONFIGURATION (ServerProperties)</sp
n>
server.port=<span class="number">8080</span>
server.address= <span class="preprocessor"># bind to a specific NIC</span>
server.session-timeout= <span class="preprocessor"># session timeout in seconds</span>
server.context-path= <span class="preprocessor"># the context path, defaults to '/'</span>
server.servlet-path= <span class="preprocessor"># the servlet path, defaults to '/'</span>
server.tomcat.access-log-pattern= <span class="preprocessor"># log pattern of the access l
g</span>
server.tomcat.access-log-enabled=<span class="keyword">>false</span> <span class="preprocessor"># is access logging enabled</span>
server.tomcat.protocol-header=x-forwarded-proto <span class="preprocessor"># ssl forward
headers</span>
server.tomcat.remote-ip-header=x-forwarded-<span class="keyword">for</span>
server.tomcat.basedir=/tmp <span class="preprocessor"># base dir (usually not needed, def
ults to tmp)</span>
server.tomcat.background-processor-delay=<span class="number">30</span>; <span class="preprocessor">#

```

```
"preprocessor"># in seconds</span>
server.tomcat.max-threads = <span class="number">0</span> <span class="preprocessor"
# number of threads in protocol handler</span>
server.tomcat.uri-encoding = UTF-<span class="number">8</span> <span class="preproces
or"># character encoding to use for URL decoding</span>
```

```
<span class="preprocessor"># SPRING MVC (HttpMapperProperties)</span>
http.mappers.json-pretty-print= <span class="keyword">>false</span> <span class="preproc
ssor"># pretty print JSON</span>
http.mappers.json-sort-keys= <span class="keyword">>false</span> <span class="preproces
or"># sort keys</span>
spring.mvc.locale= <span class="preprocessor"># set fixed locale, e.g. en_UK</span>
spring.mvc.date-format= <span class="preprocessor"># set fixed date format, e.g. dd/MM/y
yy</span>
spring.mvc.message-codes-resolver-format= <span class="preprocessor"># PREFIX_ERROR_
ODE / POSTFIX_ERROR_CODE</span>
spring.view.prefix= <span class="preprocessor"># MVC view prefix</span>
spring.view.suffix= <span class="preprocessor"># ... and suffix</span>
spring.resources.cache-period= <span class="preprocessor"># cache timeouts in headers sen
to browser</span>
spring.resources.add-mappings= <span class="keyword">>true</span> <span class="preproc
ssor"># <span class="keyword">if</span> default mappings should be added</span>
```

```
<span class="preprocessor"># THYMELEAF (ThymeleafAutoConfiguration)</span>
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html
spring.thymeleaf.mode=HTML5
spring.thymeleaf.encoding=UTF-<span class="number">8</span>
spring.thymeleaf.content-type=text/html <span class="preprocessor"># ;charset= <encodin
> is added</span>
spring.thymeleaf.cache= <span class="keyword">>true</span> <span class="preprocessor">#
set to false for hot refresh</span>
```

```
<span class="preprocessor"># FREEMARKER (FreeMarkerAutoConfiguration)</span>
spring.freemarker.allowRequestOverride= <span class="keyword">>false</span>
spring.freemarker.allowSessionOverride= <span class="keyword">>false</span>
spring.freemarker.cache= <span class="keyword">>true</span>
spring.freemarker.checkTemplateLocation= <span class="keyword">>true</span>
spring.freemarker.contentType=text/html
spring.freemarker.exposeRequestAttributes= <span class="keyword">>false</span>
spring.freemarker.exposeSessionAttributes= <span class="keyword">>false</span>
spring.freemarker.exposeSpringMacroHelpers= <span class="keyword">>false</span>
spring.freemarker.prefix=
spring.freemarker.requestContextAttribute=
spring.freemarker.settings.*=
spring.freemarker.suffix=.ftl
spring.freemarker.templateEncoding=UTF-<span class="number">8</span>
spring.freemarker.templateLoaderPath=classpath:/templates/
spring.freemarker.viewNames= <span class="preprocessor"># whitelist of view names that c
n be resolved</span>
```

```
<span class="preprocessor"># GROOVY TEMPLATES (GroovyTemplateAutoConfiguration)</
pan>
spring.groovy.template.allowRequestOverride= <span class="keyword">>false</span>
```

```
spring.groovy.template.allowSessionOverride=<span class="keyword">false</span>
spring.groovy.template.cache=<span class="keyword">>true</span>
spring.groovy.template.configuration.*=<span class="preprocessor"># See Groovy's TemplateConfiguration</span>
spring.groovy.template.contentType=text/html
spring.groovy.template.prefix=classpath:/templates/
spring.groovy.template.suffix=.tpl
spring.groovy.template.templateEncoding=UTF-8<span class="number">8</span>
spring.groovy.template.viewNames=<span class="preprocessor"># whitelist of view names that can be resolved</span>
```

```
<span class="preprocessor"># VELOCITY TEMPLATES (VelocityAutoConfiguration)</span>
```

```
spring.velocity.allowRequestOverride=<span class="keyword">false</span>
spring.velocity.allowSessionOverride=<span class="keyword">false</span>
spring.velocity.cache=<span class="keyword">>true</span>
spring.velocity.checkTemplateLocation=<span class="keyword">>true</span>
spring.velocity.contentType=text/html
spring.velocity.dateToolAttribute=
spring.velocity.exposeRequestAttributes=<span class="keyword">false</span>
spring.velocity.exposeSessionAttributes=<span class="keyword">false</span>
spring.velocity.exposeSpringMacroHelpers=<span class="keyword">false</span>
spring.velocity.numberToolAttribute=
spring.velocity.prefix=
spring.velocity.properties.*=
spring.velocity.requestContextAttribute=
spring.velocity.resourceLoaderPath=classpath:/templates/
spring.velocity.suffix=.vm
spring.velocity.templateEncoding=UTF-8<span class="number">8</span>
spring.velocity.viewNames=<span class="preprocessor"># whitelist of view names that can be resolved</span>
```

```
<span class="preprocessor"># INTERNATIONALIZATION (MessageSourceAutoConfiguration)</span>
```

```
spring.messages.basename=messages
spring.messages.cacheSeconds=-<span class="number">1</span>
spring.messages.encoding=UTF-8<span class="number">8</span>
```

```
<span class="preprocessor"># SECURITY (SecurityProperties)</span>
```

```
security.user.name=user <span class="preprocessor"># login username</span>
security.user.password=<span class="preprocessor"># login password</span>
security.user.role=USER <span class="preprocessor"># role assigned to the user</span>
security.require-ssl=<span class="keyword">false</span> <span class="preprocessor"># advanced settings ...</span>
security.enable-csrf=<span class="keyword">false</span>
security.basic.enabled=<span class="keyword">true</span>
security.basic.realm=Spring
security.basic.path=<span class="preprocessor"># /**</span>
security.headers.xss=<span class="keyword">false</span>
security.headers.cache=<span class="keyword">false</span>
security.headers.frame=<span class="keyword">false</span>
security.headers.contentType=<span class="keyword">false</span>
security.headers.hsts=all <span class="preprocessor"># none / domain / all</span>
security.sessions=stateless <span class="preprocessor"># always / never / if_required / statel
```

```

ss</span>
security.ignored= <span class="keyword">false</span>

<span class="preprocessor"># DATASOURCE (DataSourceAutoConfiguration & DataSourceP
operties)</span>
spring.datasource.name= <span class="preprocessor"># name of the data source</span>
spring.datasource.initialize= <span class="keyword">>true</span> <span class="preprocessor"
># populate using data.sql</span>
spring.datasource.schema= <span class="preprocessor"># a schema (DDL) script resource re
ference</span>
spring.datasource.data= <span class="preprocessor"># a data (DML) script resource referenc
</span>
spring.datasource.platform= <span class="preprocessor"># the platform to use in the schem
resource (schema-${platform}.sql)</span>
spring.datasource.continueOnError= <span class="keyword">>false</span> <span class="pre
rocessor"># continue even <span class="keyword">if</span> can't be initialized</span>
spring.datasource.separator=; <span class="preprocessor"># statement separator in SQL initi
lization scripts</span>
spring.datasource.driverClassName= <span class="preprocessor"># JDBC Settings...</span>
spring.datasource.url=
spring.datasource.username=
spring.datasource.password=
spring.datasource.max-active= <span class="number">100</span> <span class="preprocess
r"># Advanced configuration...</span>
spring.datasource.max-idle= <span class="number">8</span>
spring.datasource.min-idle= <span class="number">8</span>
spring.datasource.initial-size= <span class="number">10</span>
spring.datasource.validation-query=
spring.datasource.test-on-borrow= <span class="keyword">>false</span>
spring.datasource.test-on- <span class="keyword">return</span> = <span class="keyword">f
lse</span>
spring.datasource.test- <span class="keyword">while</span> -idle=
spring.datasource.time-between- eviction-runs-millis=
spring.datasource.min- evictable-idle-time-millis=
spring.datasource.max-wait-millis=

<span class="preprocessor"># MONGODB (MongoProperties)</span>
spring.data.mongodb.host= <span class="preprocessor"># the db host</span>
spring.data.mongodb.port= <span class="number">27017</span> <span class="preprocess
r"># the connection port (defaults to 27107)</span>
spring.data.mongodb.uri=mongodb: <span class="comment">//localhost/test # connection
RL</span>
spring.data.mongo.repositories.enabled= <span class="keyword">>true</span> <span class=
preprocessor"># <span class="keyword">if</span> spring data repository support is enable
</span>

<span class="preprocessor"># JPA (JpaBaseConfiguration, HibernateJpaAutoConfiguration)<
span>
spring.jpa.properties.*= <span class="preprocessor"># properties to set on the JPA connecti
n</span>
spring.jpa.openInView= <span class="keyword">>true</span>
spring.jpa.show-sql= <span class="keyword">>true</span>
spring.jpa.database-platform=
spring.jpa.database=

```

```
spring.jpa.generate-ddl=<span class="keyword">false</span> <span class="preprocessor">
  ignored by Hibernate, might be useful for other vendors</span>
spring.jpa.hibernate.naming-strategy= <span class="preprocessor"># naming classname</s
an>
spring.jpa.hibernate.ddl-auto= <span class="preprocessor"># defaults to create-drop for em
edded dbs</span>
spring.data.jpa.repositories.enabled=<span class="keyword">>true</span> <span class="pre
rocessor"># <span class="keyword">if</span> spring data repository support is enabled</s
an>
```

```
<span class="preprocessor"># SOLR (SolrProperties)</span>
spring.data.solr.host=http:<span class="comment">//127.0.0.1:8983/solr</span>
spring.data.solr.zkHost=
spring.data.solr.repositories.enabled=<span class="keyword">>true</span> <span class="pr
eprocessor"># <span class="keyword">if</span> spring data repository support is enabled</
pan>
```

```
<span class="preprocessor"># ELASTICSEARCH (ElasticsearchProperties)</span>
spring.data.elasticsearch.cluster-name= <span class="preprocessor"># The cluster name (def
ults to elasticsearch)</span>
spring.data.elasticsearch.cluster-nodes= <span class="preprocessor"># The address(es) of th
e server node (comma-separated; <span class="keyword">if</span> not specified starts a clie
nt node)</span>
spring.data.elasticsearch.local=<span class="keyword">>true</span> <span class="preproces
or"># <span class="keyword">if</span> local mode should be used with client nodes</spa
>
spring.data.elasticsearch.repositories.enabled=<span class="keyword">>true</span> <span c
lass="preprocessor"># <span class="keyword">if</span> spring data repository support is e
abled</span>
```

```
<span class="preprocessor"># FLYWAY (FlywayProperties)</span>
flyway.locations=classpath:db/migrations <span class="preprocessor"># locations of migrati
ns scripts</span>
flyway.schemas= <span class="preprocessor"># schemas to update</span>
flyway.initVersion= <span class="number">1</span> <span class="preprocessor"># version
to start migration</span>
flyway.prefix=V
flyway.suffix=.sql
flyway.enabled=<span class="keyword">>true</span>
flyway.url= <span class="preprocessor"># JDBC url <span class="keyword">if</span> you
want Flyway to create its own DataSource</span>
flyway.user= <span class="preprocessor"># JDBC username <span class="keyword">if</spa
> you want Flyway to create its own DataSource</span>
flyway.password= <span class="preprocessor"># JDBC password <span class="keyword">if<
span> you want Flyway to create its own DataSource</span>
```

```
<span class="preprocessor"># LIQUIBASE (LiquibaseProperties)</span>
liquibase.change-log=classpath:/db/changelog/db.changelog-master.yaml
liquibase.contexts= <span class="preprocessor"># runtime contexts to use</span>
liquibase.<span class="keyword">default</span>-schema= <span class="preprocessor">#
efault database schema to use</span>
liquibase.drop-first=<span class="keyword">>false</span>
```


liquibase.enabled= true

JMX

spring.jmx.enabled= true # Expose MBeans from Spring

RABBIT (RabbitProperties)

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.virtualhost=

spring.rabbitmq.dynamic=

REDIS (RedisProperties)

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

ACTIVEMQ (ActiveMQProperties)

spring.activemq.broker-url=tcp://localhost:61616 # 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

HornetQ (HornetQProperties)

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-jms-server.jar)

spring.hornetq.embedded.serverId= # auto-generated id of the embedded server (integer)

spring.hornetq.embedded.persistent= >false # message persistence

spring.hornetq.embedded.data-directory= # location of data content (when persistence is enabled)

spring.hornetq.embedded.queues= # comma separate queues

```
create on startup</span>
spring.hornetq.embedded.topics= <span class="preprocessor"># comma separate topics to create on startup</span>
spring.hornetq.embedded.cluster-password= <span class="preprocessor"># customer password (randomly generated by default)</span>

<span class="preprocessor"># JMS (JmsProperties)</span>
spring.jms.pub-sub-domain= <span class="preprocessor"># false for queue (default), true for topic</span>

<span class="preprocessor"># SPRING BATCH (BatchDatabaseInitializer)</span>
spring.batch.job.names=job1,job2
spring.batch.job.enabled=<span class="keyword">true</span>
spring.batch.initializer.enabled=<span class="keyword">true</span>
spring.batch.schema= <span class="preprocessor"># batch schema to load</span>

<span class="preprocessor"># AOP</span>
spring.aop.auto=
spring.aop.proxy-target-<span class="keyword">class</span>=

<span class="preprocessor"># FILE ENCODING (FileEncodingApplicationListener)</span>
spring.mandatory-file-encoding=<span class="keyword">>false</span>

<span class="preprocessor"># SPRING SOCIAL (SocialWebAutoConfiguration)</span>
spring.social.auto-connection-views=<span class="keyword">true</span> <span class="preprocessor"># Set to true for default connection views or false <span class="keyword">if</span>
</span> you provide your own</span>

<span class="preprocessor"># SPRING SOCIAL FACEBOOK (FacebookAutoConfiguration)</span>
spring.social.facebook.app-id= <span class="preprocessor"># your application's Facebook App ID</span>
spring.social.facebook.app-secret= <span class="preprocessor"># your application's Facebook App Secret</span>

<span class="preprocessor"># SPRING SOCIAL LINKEDIN (LinkedInAutoConfiguration)</span>
spring.social.linkedin.app-id= <span class="preprocessor"># your application's LinkedIn App ID</span>
spring.social.linkedin.app-secret= <span class="preprocessor"># your application's LinkedIn App Secret</span>

<span class="preprocessor"># SPRING SOCIAL TWITTER (TwitterAutoConfiguration)</span>
spring.social.twitter.app-id= <span class="preprocessor"># your application's Twitter App ID</span>
spring.social.twitter.app-secret= <span class="preprocessor"># your application's Twitter App Secret</span>

<span class="preprocessor"># SPRING MOBILE SITE PREFERENCE (SitePreferenceAutoConfiguration)</span>
spring.mobile.sitepreference.enabled=<span class="keyword">true</span> <span class="preprocessor"># enabled by default</span>

<span class="preprocessor"># SPRING MOBILE DEVICE VIEWS (DeviceDelegatingViewResolver)</span>
```



```
rAutoConfiguration)</span>
spring.mobile.devicedelegatingviewresolver.enabled= <span class="keyword">true</span> <
span class="preprocessor"># disabled by default</span>
spring.mobile.devicedelegatingviewresolver.normalPrefix=
spring.mobile.devicedelegatingviewresolver.normalSuffix=
spring.mobile.devicedelegatingviewresolver.mobilePrefix=mobile/
spring.mobile.devicedelegatingviewresolver.mobileSuffix=
spring.mobile.devicedelegatingviewresolver.tabletPrefix=tablet/
spring.mobile.devicedelegatingviewresolver.tabletSuffix=

<span class="preprocessor"># -----</span>
<span class="preprocessor"># ACTUATOR PROPERTIES</span>
<span class="preprocessor"># -----</span>

<span class="preprocessor"># MANAGEMENT HTTP SERVER (ManagementServerProperties)
/span>
management.port= <span class="preprocessor"># defaults to 'server.port'</span>
management.address= <span class="preprocessor"># bind to a specific NIC</span>
management.contextPath= <span class="preprocessor"># default to '/'</span>

<span class="preprocessor"># ENDPOINTS (AbstractEndpoint subclasses)</span>
endpoints.autoconfig.id=autoconfig
endpoints.autoconfig.sensitive= <span class="keyword">true</span>
endpoints.autoconfig.enabled= <span class="keyword">true</span>
endpoints.beans.id=beans
endpoints.beans.sensitive= <span class="keyword">true</span>
endpoints.beans.enabled= <span class="keyword">true</span>
endpoints.configprops.id=configprops
endpoints.configprops.sensitive= <span class="keyword">true</span>
endpoints.configprops.enabled= <span class="keyword">true</span>
endpoints.configprops.keys-to-sanitize=password,secret
endpoints.dump.id=dump
endpoints.dump.sensitive= <span class="keyword">true</span>
endpoints.dump.enabled= <span class="keyword">true</span>
endpoints.env.id=env
endpoints.env.sensitive= <span class="keyword">true</span>
endpoints.env.enabled= <span class="keyword">true</span>
endpoints.health.id=health
endpoints.health.sensitive= <span class="keyword">false</span>
endpoints.health.enabled= <span class="keyword">true</span>
endpoints.info.id=info
endpoints.info.sensitive= <span class="keyword">false</span>
endpoints.info.enabled= <span class="keyword">true</span>
endpoints.metrics.id=metrics
endpoints.metrics.sensitive= <span class="keyword">true</span>
endpoints.metrics.enabled= <span class="keyword">true</span>
endpoints.shutdown.id=shutdown
endpoints.shutdown.sensitive= <span class="keyword">true</span>
endpoints.shutdown.enabled= <span class="keyword">false</span>
endpoints.trace.id=trace
endpoints.trace.sensitive= <span class="keyword">true</span>
endpoints.trace.enabled= <span class="keyword">true</span>

<span class="preprocessor"># MVC ONLY ENDPOINTS</span>
```

```
endpoints.jolokia.path=jolokia
endpoints.jolokia.sensitive= <span class="keyword">true</span>
endpoints.jolokia.enabled= <span class="keyword">true</span> <span class="preprocessor"
># when using Jolokia</span>
endpoints.error.path=/error

<span class="preprocessor"># JMX ENDPOINT (EndpointMBeanExportProperties)</span>
endpoints.jmx.enabled= <span class="keyword">true</span>
endpoints.jmx.domain= <span class="preprocessor"># the JMX domain, defaults to 'org.spr
igboot'</span>
endpoints.jmx.unique-names= <span class="keyword">>false</span>
endpoints.jmx.enabled= <span class="keyword">true</span>
endpoints.jmx.staticNames=

<span class="preprocessor"># JOLOKIA (JolokiaProperties)</span>
jolokia.config.*= <span class="preprocessor"># See Jolokia manual</span>

<span class="preprocessor"># REMOTE SHELL</span>
shell.auth=simple <span class="preprocessor"># jaas, key, simple, spring</span>
shell.command-refresh-interval=-<span class="number">1</span>
shell.command-path-pattern= <span class="preprocessor"># classpath*/commands/**, clas
path*/crash/commands/**</span>
shell.config-path-patterns= <span class="preprocessor"># classpath*/crash/*</span>
shell.disabled-plugins= <span class="keyword">>false</span> <span class="preprocessor">#
don't expose plugins</span>
shell.ssh.enabled= <span class="preprocessor"># ssh settings ...</span>
shell.ssh.keyPath=
shell.ssh.port=
shell.telnet.enabled= <span class="preprocessor"># telnet settings ...</span>
shell.telnet.port=
shell.auth.jaas.domain= <span class="preprocessor"># authentication settings ...</span>
shell.auth.key.path=
shell.auth.simple.user.name=
shell.auth.simple.user.password=
shell.auth.spring.roles=

<span class="preprocessor"># GIT INFO</span>
spring.git.properties= <span class="preprocessor"># resource ref to generated git info prope
ties file</span></pre>
```