SpringBoot学习体会

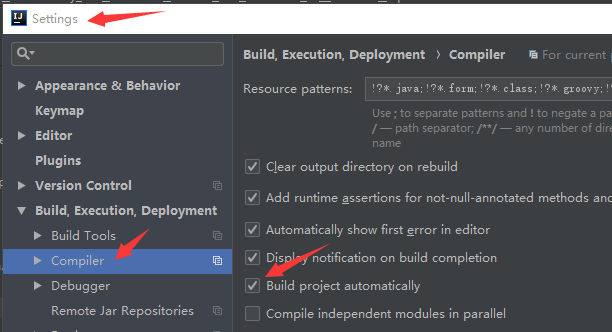
1. SpringBoot是什么

SpringBoot其实是一个启动spring项目的工具。从根本上来说，Spring Boot是对各种框架的整合，让他们集成在一起更加简单，它能够被任意项目的构建系统所使用。

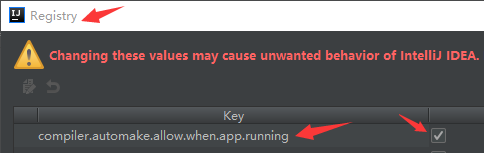
1. SpringBoot结合Idea的热发布
2. 创建SpringBoot工程时，勾选Core—DevTools 并在Pom.xml里添加

<!--支持热启动jar包-->  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <optional>true</optional>  
</dependency>

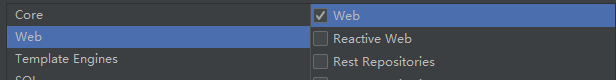
2、勾选



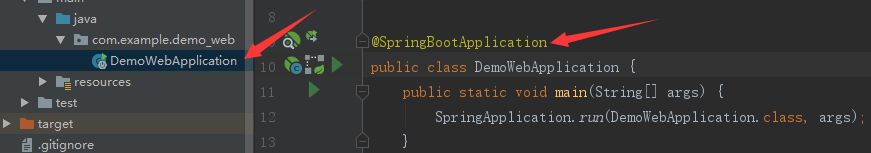
3、使用快捷键**Ctrl + shift + alt + /**，打开Maintenance操作面板，选择Registry，打开Registry操作面板，找到并勾选。完成热发布的部署。



1. SpringBoot发布web项目的
   1. 创建SpringBoot项目时勾选Web



2、在项目的根目录下找到XXApplication，DemoWebApplication类，是整个项目的入口类，这个类有一个@SpringBootApplication注解，这是整个Spring Boot的核心注解，它的目的就是开启Spring Boot的自动配置。

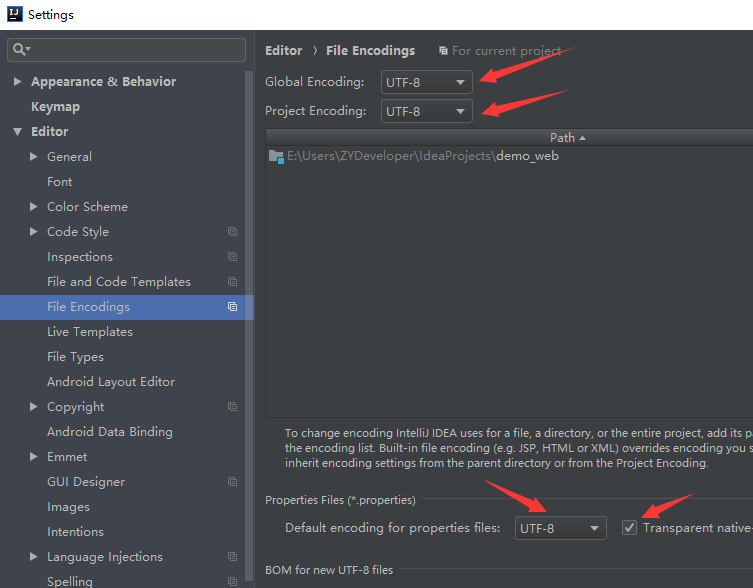


3、利用注解即可实现页面输出

@RestController  
public class Controller {  
 @RequestMapping  
 public String test(){  
 return "<h1>Hello 正元</h1>";  
 }  
}



1. SpringBoot的配置文件application.properties，相关配置见“附件1”
2. SpringBoot中乱码问题,设置以下位置，即可解决乱码问题



1. SprinfBoot属性配置操作
   1. 配置文件中的添加方式

book.author=罗贯中  
book.name=三国演义  
book.pinyin=sanguoyanyi

2、java文件中@Value方法直接获取属性

@Value(value = "${book.author}")  
private String bookAuthor;  
 @Value(value = "${book.name}")  
private String bookName;  
 @Value(value = "${book.pinyin}")  
private String bookPinyin;

附件1：SpringBoot的配置文件配置大全

#############################################################

# mvc

#############################################################

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tomcat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.

#############################################################

# messages

#############################################################

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，默认:=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

#############################################################

# mobile

#############################################################

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

#############################################################

# view

#############################################################

spring.view.prefix=设定mvc视图的前缀.

spring.view.suffix=设定mvc视图的后缀.

#############################################################

# resource

#############################################################

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context:/

#############################################################

# multipart

#############################################################

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

#############################################################

# freemarker

#############################################################

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.

#############################################################

# velocity

#############################################################

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类库使用

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.

#############################################################

# thymeleaf

#############################################################

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.

#############################################################

# mustcache

#############################################################

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.

#############################################################

# groovy模板

#############################################################

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.

#############################################################

# http

#############################################################

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

#############################################################

# json

#############################################################

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会作为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)或者指定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.

#############################################################

# jersey

#############################################################

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.server配置

server.address=指定server绑定的地址

server.compression.enabled=是否开启压缩，默认为false.

server.compression.excluded-user-agents=指定不压缩的user-agent，多个以逗号分隔，默认值为:text/html,text/xml,text/plain,text/css

server.compression.mime-types=指定要压缩的MIME type，多个以逗号分隔.

server.compression.min-response-size=执行压缩的阈值，默认为2048

server.context-parameters.[param name]=设置servlet context 参数

server.context-path=设定应用的context-path.

server.display-name=设定应用的展示名称，默认: application

server.jsp-servlet.class-name=设定编译JSP用的servlet，默认: org.apache.jasper.servlet.JspServlet)

server.jsp-servlet.init-parameters.[param name]=设置JSP servlet 初始化参数.

server.jsp-servlet.registered=设定JSP servlet是否注册到内嵌的servlet容器，默认true

server.port设定http=监听端口

server.servlet-path=设定dispatcher servlet的监听路径，默认为: /

#############################################################

# cookie、session配置

#############################################################

server.session.cookie.comment=指定session cookie的comment

server.session.cookie.domain=指定session cookie的domain

server.session.cookie.http-only=是否开启HttpOnly.

server.session.cookie.max-age=设定session cookie的最大age.

server.session.cookie.name=设定Session cookie 的名称.

server.session.cookie.path=设定session cookie的路径.

server.session.cookie.secure=设定session cookie的“Secure” flag.

server.session.persistent=重启时是否持久化session，默认false

server.session.timeoutsession=的超时时间

server.session.tracking-modes=设定Session的追踪模式(cookie, url, ssl).

#############################################################

# ssl配置

#############################################################

server.ssl.ciphers 是否支持SSL ciphers.

server.ssl.client-auth 设定client authentication是wanted 还是 needed.

server.ssl.enabled 是否开启ssl，默认: true

server.ssl.key-alias 设定key store中key的别名.

server.ssl.key-password 访问key store中key的密码.

server.ssl.key-store 设定持有SSL certificate的key store的路径，通常是一个.jks文件.

server.ssl.key-store-password 设定访问key store的密码.

server.ssl.key-store-provider 设定key store的提供者.

server.ssl.key-store-type 设定key store的类型.

server.ssl.protocol 使用的SSL协议，默认: TLS

server.ssl.trust-store 持有SSL certificates的Trust store.

server.ssl.trust-store-password 访问trust store的密码.

server.ssl.trust-store-provider 设定trust store的提供者.

server.ssl.trust-store-type 指定trust store的类型.

#############################################################

# tomcat

#############################################################

server.tomcat.access-log-enabled 是否开启access log ，默认: false)

server.tomcat.access-log-pattern 设定access logs的格式，默认: common

server.tomcat.accesslog.directory 设定log的目录，默认: logs

server.tomcat.accesslog.enabled 是否开启access log，默认: false

server.tomcat.accesslog.pattern 设定access logs的格式，默认: common

server.tomcat.accesslog.prefix 设定Log 文件的前缀，默认: access\_log

server.tomcat.accesslog.suffix 设定Log 文件的后缀，默认: .log

server.tomcat.background-processor-delay 后台线程方法的Delay大小: 30

server.tomcat.basedir 设定Tomcat的base 目录，如果没有指定则使用临时目录.

server.tomcat.internal-proxies 设定信任的正则表达式，默认:“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}| 172.1[6-9]{1}.\d{1,3}.\d{1,3}| 172.2[0-9]{1}.\d{1,3}.\d{1,3}|172.3[0-1]{1}.\d{1,3}.\d{1,3}”

server.tomcat.max-http-header-size 设定http header的最小值，默认: 0

server.tomcat.max-threads 设定tomcat的最大工作线程数，默认为: 0

server.tomcat.port-header 设定http header使用的，用来覆盖原来port的value.

server.tomcat.protocol-header 设定Header包含的协议，通常是 X-Forwarded-Proto，如果remoteIpHeader有值，则将设置为RemoteIpValve.

server.tomcat.protocol-header-https-value 设定使用SSL的header的值，默认https.

server.tomcat.remote-ip-header 设定remote IP的header，如果remoteIpHeader有值，则设置为RemoteIpValve

server.tomcat.uri-encoding 设定URI的解码字符集.

#############################################################

# undertow

#############################################################

server.undertow.access-log-dir 设定Undertow access log 的目录，默认: logs

server.undertow.access-log-enabled 是否开启access log，默认: false

server.undertow.access-log-pattern 设定access logs的格式，默认: common

server.undertow.accesslog.dir 设定access log 的目录.

server.undertow.buffer-size 设定buffer的大小.

server.undertow.buffers-per-region 设定每个region的buffer数

server.undertow.direct-buffers 设定堆外内存

server.undertow.io-threads 设定I/O线程数.

server.undertow.worker-threads 设定工作线程数

#############################################################

# datasource

#############################################################

spring.dao.exceptiontranslation.enabled 是否开启PersistenceExceptionTranslationPostProcessor，默认为true

spring.datasource.abandon-when-percentage-full 设定超时被废弃的连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 是否允许替代的用户名.

spring.datasource.auto-commit 指定updates是否自动提交.

spring.datasource.catalog 指定默认的catalog.

spring.datasource.commit-on-return 设置当连接被归还时，是否要提交所有还未完成的事务

spring.datasource.connection-init-sql 指定连接被创建，再被添加到连接池之前执行的sql.

spring.datasource.connection-init-sqls 使用DBCP connection pool时，指定初始化时要执行的sql

spring.datasource.connection-properties.[key] 在使用DBCP connection pool时指定要配置的属性

spring.datasource.connection-test-query 指定校验连接合法性执行的sql语句

spring.datasource.connection-timeout 指定连接的超时时间，毫秒单位.

spring.datasource.continue-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)脚本

spring.datasource.data-source-class-name 指定数据源的全限定名.

spring.datasource.data-source-jndi 指定jndi的地址

spring.datasource.data-source-properties.[key] 使用Hikari connection pool时，指定要设置的属性

spring.datasource.db-properties 使用Tomcat connection 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catalog.

spring.datasource.default-read-only 是否设置默认连接只读.

spring.datasource.default-transaction-isolation 指定连接的事务的默认隔离级别.

spring.datasource.driver-class-name 指定driver的类名，默认从jdbc url中自动探测.

spring.datasource.fair-queue 是否采用FIFO返回连接.

spring.datasource.health-check-properties.[key] 使用Hikari connection pool时，在心跳检查时传递的属性

spring.datasource.idle-timeout 指定连接多久没被使用时，被设置为空闲，默认为10ms

spring.datasource.ignore-exception-on-pre-load 当初始化连接池时，是否忽略异常.

spring.datasource.init-sql 当连接创建时，执行的sql

spring.datasource.initial-size 指定启动连接池时，初始建立的连接数量

spring.datasource.initialization-fail-fast 当创建连接池时，没法创建指定最小连接数量是否抛异常

spring.datasource.initialize 指定初始化数据源，是否用data.sql来初始化，默认: true

spring.datasource.isolate-internal-queries 指定内部查询是否要被隔离，默认为false

spring.datasource.jdbc-interceptors 使用Tomcat connection pool时，指定jdbc拦截器，分号分隔

spring.datasource.jdbc-url 指定JDBC URL.

spring.datasource.jmx-enabled 是否开启JMX，默认为: false

spring.datasource.jndi-name 指定jndi的名称.

spring.datasource.leak-detection-threshold 使用Hikari connection pool时，多少毫秒检测一次连接泄露.

spring.datasource.log-abandoned 使用DBCP connection pool，是否追踪废弃statement或连接，默认为: false

spring.datasource.log-validation-errors 当使用Tomcat connection pool是否打印校验错误.

spring.datasource.login-timeout 指定连接数据库的超时时间.

spring.datasource.max-active 指定连接池中最大的活跃连接数.

spring.datasource.max-age 指定连接池中连接的最大年龄

spring.datasource.max-idle 指定连接池最大的空闲连接数量.

spring.datasource.max-lifetime 指定连接池中连接的最大生存时间，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 指定连接池等待连接返回的最大等待时间，毫秒单位.

spring.datasource.maximum-pool-size 指定连接池最大的连接数，包括使用中的和空闲的连接.

spring.datasource.min-evictable-idle-time-millis 指定一个空闲连接最少空闲多久后可被清除.

spring.datasource.min-idle 指定必须保持连接的最小值(For DBCP and Tomcat connection pools)

spring.datasource.minimum-idle 指定连接维护的最小空闲连接数，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 object 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要使用的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，如果线程被中断，是否传播中断状态.

spring.datasource.read-only 当使用Hikari connection pool时，是否标记数据源只读

spring.datasource.register-mbeans 指定Hikari connection pool是否注册JMX MBeans.

spring.datasource.remove-abandoned 指定当连接超过废弃超时时间时，是否立刻删除该连接.

spring.datasource.remove-abandoned -timeout指定连接应该被废弃的时间.

spring.datasource.rollback-on-return 在归还连接时，是否回滚等待中的事务.

spring.datasource.schema 指定Schema (DDL)脚本.

spring.datasource.separator 指定初始化脚本的语句分隔符，默认: ;

spring.datasource.sql-script-encoding 指定SQL scripts编码.

spring.datasource.suspect-timeout 指定打印废弃连接前的超时时间.

spring.datasource.test-on-borrow 当从连接池借用连接时，是否测试该连接.

spring.datasource.test-on-connect 创建时，是否测试连接

spring.datasource.test-on-return 在连接归还到连接池时是否测试该连接.

spring.datasource.test-while-idle 当连接空闲时，是否执行连接测试.

spring.datasource.time-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.transaction-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 是否对连接进行包装，防止连接关闭之后被使用.

spring.datasource.use-equals 比较方法名时是否使用String.equals()替换==.

spring.datasource.use-lock 是否对连接操作加锁

spring.datasource.username 指定数据库名.

spring.datasource.validation-interval 指定多少ms执行一次连接校验.

spring.datasource.validation-query 指定获取连接时连接校验的sql查询语句.

spring.datasource.validation-query-timeout 指定连接校验查询的超时时间.

spring.datasource.validation-timeout 设定连接校验的超时时间，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.xa.data-source-class-name 指定数据源的全限定名.

spring.datasource.xa.properties 指定传递给XA data source的属性

#############################################################

# JPA

#############################################################

spring.jpa.database 指定目标数据库.

spring.jpa.database-platform 指定目标数据库的类型.

spring.jpa.generate-ddl 是否在启动时初始化schema，默认为false

spring.jpa.hibernate.ddl-auto 指定DDL mode (none, validate, update, create, create-drop). 当使用内嵌数据库时，默认是create-drop，否则为none.

spring.jpa.hibernate.naming-strategy 指定命名策略.

spring.jpa.open-in-view 是否注册OpenEntityManagerInViewInterceptor，绑定JPA EntityManager到请求线程中，默认为: true

spring.jpa.properties 添加额外的属性到JPA provider.

spring.jpa.show-sql 是否开启sql的log，默认为: false

#############################################################

# jooq

#############################################################

spring.jooq.sql-dialect 指定JOOQ使用的SQLDialect，比如POSTGRES.

#############################################################

# h2

#############################################################

spring.h2.console.enabled 是否开启控制台，默认为false

spring.h2.console.path 指定控制台路径，默认为: /h2-console

#############################################################

# JTA

#############################################################

spring.jta.allow-multiple-lrc 是否允许 multiple LRC，默认为: false

spring.jta.asynchronous2-pc 指定两阶段提交是否可以异步，默认为: false

spring.jta.background-recovery-interval 指定多少分钟跑一次recovery process，默认为: 1

spring.jta.background-recovery-interval-seconds 指定多久跑一次recovery process，默认: 60

spring.jta.current-node-only-recovery 是否过滤掉其他非本JVM的recovery，默认为: true

spring.jta.debug-zero-resource-transaction 是否追踪没有使用指定资源的事务，默认为: false

spring.jta.default-transaction-timeout 设定默认的事务超时时间，默认为60

spring.jta.disable-jmx 是否禁用jmx，默认为false

spring.jta.enabled 是否开启JTA support，默认为: true

spring.jta.exception-analyzer 设置指定的异常分析类

spring.jta.filter-log-status 使用Bitronix Transaction Manager时，是否写mandatory logs，开启的话，可以节省磁盘空间，但是调试会复杂写，默认为false

spring.jta.force-batching-enabled 使用Bitronix Transaction Manager时，是否批量写磁盘，默认为true.

spring.jta.forced-write-enabled 使用Bitronix Transaction Manager时，是否强制写日志到磁盘，默认为true

spring.jta.graceful-shutdown-interval 当使用Bitronix Transaction Manager，指定shutdown时等待事务结束的时间，超过则中断，默认为60

spring.jta.jndi-transaction-synchronization-registry-name 当使用Bitronix Transaction Manager时，在JNDI下得事务同步registry，默认为: java:comp/TransactionSynchronizationRegistry

spring.jta.jndi-user-transaction-name 指定在JNDI使用Bitronix Transaction Manager的名称，默认:java:comp/UserTransaction

spring.jta.journal 当使用Bitronix Transaction Manager，指定The journal是否disk还是null还是一个类的全限定名，默认disk

spring.jta.log-dirTransaction logs directory.

spring.jta.log-part1-filename 指定The journal fragment文件1的名字，默认: btm1.tlog

spring.jta.log-part2-filename 指定The journal fragment文件2的名字，默认: btm2.tlog

spring.jta.max-log-size-in-mb 指定journal fragments大小的最大值. 默认: 2M

spring.jta.resource-configuration-filename 指定Bitronix Transaction Manager配置文件名.

spring.jta.server-id 指定Bitronix Transaction Manager实例的id.

spring.jta.skip-corrupted-logs 是否忽略corrupted log files文件，默认为false.

spring.jta.transaction-manager-id 指定Transaction manager的唯一标识.

spring.jta.warn-about-zero-resource-transaction 当使用Bitronix Transaction Manager时，是否对没有使用指定资源的事务进行警告，默认为: true

#############################################################

# cache

#############################################################

spring.cache.cache-names 指定要创建的缓存的名称，逗号分隔(若该缓存实现支持的话)

spring.cache.ehcache.config 指定初始化EhCache时使用的配置文件的位置指定.

spring.cache.guava.spec 指定创建缓存要使用的spec，具体详见CacheBuilderSpec.

spring.cache.hazelcast.config 指定初始化Hazelcast时的配置文件位置

spring.cache.infinispan.config 指定初始化Infinispan时的配置文件位置.

spring.cache.jcache.config 指定jcache的配置文件.

spring.cache.jcache.provider 指定CachingProvider实现类的全限定名.

spring.cache.type 指定缓存类型

#############################################################

# mongodb

#############################################################

spring.mongodb.embedded.features 指定要开启的特性，逗号分隔.

spring.mongodb.embedded.version 指定要使用的版本，默认: 2.6.10

#############################################################

# redis

#############################################################

spring.redis.database 指定连接工厂使用的Database index，默认为: 0

spring.redis.host 指定Redis server host，默认为: localhost

spring.redis.password 指定Redis server的密码

spring.redis.pool.max-active 指定连接池最大的活跃连接数，-1表示无限，默认为8

spring.redis.pool.max-idle 指定连接池最大的空闲连接数，-1表示无限，默认为8

spring.redis.pool.max-wait 指定当连接池耗尽时，新获取连接需要等待的最大时间，以毫秒单位，-1表示无限等待

spring.redis.pool.min-idle 指定连接池中空闲连接的最小数量，默认为0

spring.redis.port 指定redis服务端端口，默认: 6379

spring.redis.sentinel.master 指定redis server的名称

spring.redis.sentinel.nodes 指定sentinel节点，逗号分隔，格式为host:port.

spring.redis.timeout 指定连接超时时间，毫秒单位，默认为0

#############################################################

# springdata

#############################################################

spring.data.elasticsearch.cluster-name 指定es集群名称，默认: elasticsearch

spring.data.elasticsearch.cluster-nodes 指定es的集群，逗号分隔，不指定的话，则启动client node.

spring.data.elasticsearch.properties 指定要配置的es属性.

spring.data.elasticsearch.repositories.enabled 是否开启es存储，默认为: true

spring.data.jpa.repositories.enabled 是否开启JPA支持，默认为: true

spring.data.mongodb.authentication-database 指定鉴权的数据库名

spring.data.mongodb.database 指定mongodb数据库名

spring.data.mongodb.field-naming-strategy 指定要使用的FieldNamingStrategy.

spring.data.mongodb.grid-fs-database 指定GridFS database的名称.

spring.data.mongodb.host 指定Mongo server host.

spring.data.mongodb.password 指定Mongo server的密码.

spring.data.mongodb.port 指定Mongo server port.

spring.data.mongodb.repositories.enabled 是否开启mongodb存储，默认为true

spring.data.mongodb.uri 指定Mongo database URI.默认:mongodb://localhost/test

spring.data.mongodb.username 指定登陆mongodb的用户名.

spring.data.rest.base-path 指定暴露资源的基准路径.

spring.data.rest.default-page-size 指定每页的大小，默认为: 20

spring.data.rest.limit-param-name 指定limit的参数名，默认为: size

spring.data.rest.max-page-size 指定最大的页数，默认为1000

spring.data.rest.page-param-name 指定分页的参数名，默认为: page

spring.data.rest.return-body-on-create 当创建完实体之后，是否返回body，默认为false

spring.data.rest.return-body-on-update 在更新完实体后，是否返回body，默认为false

spring.data.rest.sort-param-name 指定排序使用的key，默认为: sort

spring.data.solr.host 指定Solr host，如果有指定了zk的host的话，则忽略。默认为: http://127.0.0.1:8983/solr

spring.data.solr.repositories.enabled 是否开启Solr repositories，默认为: true

spring.data.solr.zk-host 指定zk的地址，格式为HOST:PORT.

#############################################################

# activemq

#############################################################

spring.activemq.broker-url 指定ActiveMQ broker的URL，默认自动生成.

spring.activemq.in-memory 是否是内存模式，默认为true.

spring.activemq.password 指定broker的密码.

spring.activemq.pooled 是否创建PooledConnectionFactory，而非ConnectionFactory，默认false

spring.activemq.user 指定broker的用户.

#############################################################

# artemis(HornetQ捐献给apache后的版本)

#############################################################

spring.artemis.embedded.cluster-password 指定集群的密码，默认是启动时随机生成.

spring.artemis.embedded.data-directory 指定Journal文件的目录.如果不开始持久化则不必要指定.

spring.artemis.embedded.enabled 是否开启内嵌模式，默认true

spring.artemis.embedded.persistent 是否开启persistent store，默认false.

spring.artemis.embedded.queues 指定启动时创建的队列，多个用逗号分隔，默认: []

spring.artemis.embedded.server-id 指定Server ID. 默认是一个自增的数字，从0开始.

spring.artemis.embedded.topics 指定启动时创建的topic，多个的话逗号分隔，默认: []

spring.artemis.host 指定Artemis broker 的host. 默认: localhost

spring.artemis.mode 指定Artemis 的部署模式, 默认为auto-detected(也可以为native or embedded).

spring.artemis.port 指定Artemis broker 的端口，默认为: 61616

#############################################################

# rabbitmq

#############################################################

spring.rabbitmq.addresses 指定client连接到的server的地址，多个以逗号分隔.

spring.rabbitmq.dynamic 是否创建AmqpAdmin bean. 默认为: true)

spring.rabbitmq.host 指定RabbitMQ host.默认为: localhost)

spring.rabbitmq.listener.acknowledge-mode 指定Acknowledge的模式.

spring.rabbitmq.listener.auto-startup 是否在启动时就启动mq，默认: true)

spring.rabbitmq.listener.concurrency 指定最小的消费者数量.

spring.rabbitmq.listener.max-concurrency 指定最大的消费者数量.

spring.rabbitmq.listener.prefetch 指定一个请求能处理多少个消息，如果有事务的话，必须大于等于transaction数量.

spring.rabbitmq.listener.transaction-size 指定一个事务处理的消息数量，最好是小于等于prefetch的数量.

spring.rabbitmq.password 指定broker的密码.

spring.rabbitmq.port 指定RabbitMQ 的端口，默认: 5672)

spring.rabbitmq.requested-heartbeat 指定心跳超时，0为不指定.

spring.rabbitmq.ssl.enabled 是否开始SSL，默认: false)

spring.rabbitmq.ssl.key-store 指定持有SSL certificate的key store的路径

spring.rabbitmq.ssl.key-store-password 指定访问key store的密码.

spring.rabbitmq.ssl.trust-store 指定持有SSL certificates的Trust store.

spring.rabbitmq.ssl.trust-store-password 指定访问trust store的密码.

spring.rabbitmq.username 指定登陆broker的用户名.

spring.rabbitmq.virtual-host 指定连接到broker的Virtual host.

#############################################################

# hornetq

#############################################################

spring.hornetq.embedded.cluster-password 指定集群的密码，默认启动时随机生成.

spring.hornetq.embedded.data-directory 指定Journal file 的目录. 如果不开启持久化则不必指定.

spring.hornetq.embedded.enabled 是否开启内嵌模式，默认:true

spring.hornetq.embedded.persistent 是否开启persistent store，默认: false

spring.hornetq.embedded.queues 指定启动是创建的queue，多个以逗号分隔，默认: []

spring.hornetq.embedded.server-id 指定Server ID. 默认使用自增数字，从0开始.

spring.hornetq.embedded.topics 指定启动时创建的topic，多个以逗号分隔，默认: []

spring.hornetq.host 指定HornetQ broker 的host，默认: localhost

spring.hornetq.mode 指定HornetQ 的部署模式，默认是auto-detected，也可以指定native 或者 embedded.

spring.hornetq.port 指定HornetQ broker 端口，默认: 5445

#############################################################

# jms

#############################################################

spring.jms.jndi-name 指定Connection factory JNDI 名称.

spring.jms.listener.acknowledge-mode 指定ack模式，默认自动ack.

spring.jms.listener.auto-startup 是否启动时自动启动jms，默认为: true

spring.jms.listener.concurrency 指定最小的并发消费者数量.

spring.jms.listener.max-concurrency 指定最大的并发消费者数量.

spring.jms.pub-sub-domain 是否使用默认的destination type来支持 publish/subscribe，默认: false

#############################################################

# security

#############################################################

security.basic.authorize-mode 要使用权限控制模式.

security.basic.enabled 是否开启基本的鉴权，默认为true

security.basic.path 需要鉴权的path，多个的话以逗号分隔，默认为[/\*\*]

security.basic.realm HTTP basic realm 的名字，默认为Spring

security.enable-csrf 是否开启cross-site request forgery校验，默认为false.

security.filter-order Security filter chain的order，默认为0

security.headers.cache 是否开启http头部的cache控制，默认为false.

security.headers.content-type 是否开启X-Content-Type-Options头部，默认为false.

security.headers.frame 是否开启X-Frame-Options头部，默认为false.

security.headers.hsts 指定HTTP Strict Transport Security (HSTS)模式(none, domain, all).

security.headers.xss 是否开启cross-site scripting (XSS) 保护，默认为false.

security.ignored 指定不鉴权的路径，多个的话以逗号分隔.

security.oauth2.client.access-token-uri 指定获取access token的URI.

security.oauth2.client.access-token-validity-seconds 指定access token失效时长.

security.oauth2.client.additional-information.[key] 设定要添加的额外信息.

security.oauth2.client.authentication-scheme 指定传输不记名令牌(bearer token)的方式(form, header, none,query)，默认为header

security.oauth2.client.authorities 指定授予客户端的权限.

security.oauth2.client.authorized-grant-types 指定客户端允许的grant types.

security.oauth2.client.auto-approve-scopes 对客户端自动授权的scope.

security.oauth2.client.client-authentication-scheme 传输authentication credentials的方式(form, header, none, query)，默认为header方式

security.oauth2.client.client-id 指定OAuth2 client ID.

security.oauth2.client.client-secret 指定OAuth2 client secret. 默认是一个随机的secret.

security.oauth2.client.grant-type 指定获取资源的access token的授权类型.

security.oauth2.client.id 指定应用的client ID.

security.oauth2.client.pre-established-redirect-uri 服务端pre-established的跳转URI.

security.oauth2.client.refresh-token-validity-seconds 指定refresh token的有效期.

security.oauth2.client.registered-redirect-uri 指定客户端跳转URI，多个以逗号分隔.

security.oauth2.client.resource-ids 指定客户端相关的资源id，多个以逗号分隔.

security.oauth2.client.scopeclient 的scope

security.oauth2.client.token-name 指定token的名称

security.oauth2.client.use-current-uri 是否优先使用请求中URI，再使用pre-established的跳转URI. 默认为true

security.oauth2.client.user-authorization-uri 用户跳转去获取access token的URI.

security.oauth2.resource.id 指定resource的唯一标识.

security.oauth2.resource.jwt.key-uri JWT token的URI. 当key为公钥时，或者value不指定时指定.

security.oauth2.resource.jwt.key-value JWT token验证的value. 可以是对称加密或者PEMencoded RSA公钥. 可以使用URI作为value.

security.oauth2.resource.prefer-token-info 是否使用token info，默认为true

security.oauth2.resource.service-id 指定service ID，默认为resource.

security.oauth2.resource.token-info-uritoken 解码的URI.

security.oauth2.resource.token-type 指定当使用userInfoUri时，发送的token类型.

security.oauth2.resource.user-info-uri 指定user info的URI

security.oauth2.sso.filter-order 如果没有显示提供WebSecurityConfigurerAdapter时指定的Filter order.

security.oauth2.sso.login-path 跳转到SSO的登录路径默认为/login.

security.require-ssl 是否对所有请求开启SSL，默认为false.

security.sessions 指定Session的创建策略(always, never, if\_required, stateless).

security.user.name 指定默认的用户名，默认为user.

security.user.password 默认的用户密码.

security.user.role 默认用户的授权角色.

#############################################################

# SpringBoot支持了两种数据库迁移工具，一个是flyway，一个是liquibase。其本身也支持sql script，在初始化数据源之后执行指定的脚本。

# flyway

#############################################################

flyway.baseline-description 对执行迁移时基准版本的描述.

flyway.baseline-on-migrate 当迁移时发现目标schema非空，而且带有没有元数据的表时，是否自动执行基准迁移，默认false.

flyway.baseline-version 开始执行基准迁移时对现有的schema的版本打标签，默认值为1.

flyway.check-location 检查迁移脚本的位置是否存在，默认false.

flyway.clean-on-validation-error 当发现校验错误时是否自动调用clean，默认false.

flyway.enabled 是否开启flywary，默认true.

flyway.encoding 设置迁移时的编码，默认UTF-8.

flyway.ignore-failed-future-migration 当读取元数据表时是否忽略错误的迁移，默认false.

flyway.init-sqls 当初始化好连接时要执行的SQL.

flyway.locations 迁移脚本的位置，默认db/migration.

flyway.out-of-order 是否允许无序的迁移，默认false.

flyway.password 目标数据库的密码.

flyway.placeholder-prefix 设置每个placeholder的前缀，默认${.

flyway.placeholder-replacementplaceholders 是否要被替换，默认true.

flyway.placeholder-suffix 设置每个placeholder的后缀，默认}.

flyway.placeholders.[placeholder name] 设置placeholder的value

flyway.schemas 设定需要flywary迁移的schema，大小写敏感，默认为连接默认的schema.

flyway.sql-migration-prefix 迁移文件的前缀，默认为V.

flyway.sql-migration-separator 迁移脚本的文件名分隔符，默认\_\_

flyway.sql-migration-suffix 迁移脚本的后缀，默认为.sql

flyway.tableflyway 使用的元数据表名，默认为schema\_version

flyway.target 迁移时使用的目标版本，默认为latest version

flyway.url 迁移时使用的JDBC URL，如果没有指定的话，将使用配置的主数据源

flyway.user 迁移数据库的用户名

flyway.validate-on-migrate 迁移时是否校验，默认为true.

#############################################################

# liquibase

#############################################################

liquibase.change-logChange log 配置文件的路径，默认值为classpath:/db/changelog/db.changelog-master.yaml

liquibase.check-change-log-location 是否坚持change log的位置是否存在，默认为true.

liquibase.contexts 逗号分隔的运行时context列表.

liquibase.default-schema 默认的schema.

liquibase.drop-first 是否首先drop schema，默认为false

liquibase.enabled 是否开启liquibase，默认为true.

liquibase.password 目标数据库密码

liquibase.url 要迁移的JDBC URL，如果没有指定的话，将使用配置的主数据源.

liquibase.user 目标数据用户名

#############################################################

# aop

#############################################################

spring.aop.auto 是否支持@EnableAspectJAutoProxy，默认为: true

spring.aop.proxy-target-classtrue 为使用CGLIB代理，false为JDK代理，默认为false

#############################################################

# application

#############################################################

spring.application.admin.enabled 是否启用admin特性，默认为: false

spring.application.admin.jmx-name 指定admin MBean的名称，默认为: org.springframework.boot:type=Admin,name=SpringApplication

#############################################################

# autoconfig

#############################################################

spring.autoconfigure.exclude 配置要排除的Auto-configuration classes.

#############################################################

# batch

#############################################################

spring.batch.initializer.enabled 是否在必要时创建batch表，默认为true

spring.batch.job.enabled 是否在启动时开启batch job，默认为true

spring.batch.job.names 指定启动时要执行的job的名称，逗号分隔，默认所有job都会被执行

spring.batch.schema 指定要初始化的sql语句路径，默认:classpath:org/springframework/batch/core/schema-@@platform@@.sql)

spring.batch.table-prefix 指定批量处理的表的前缀.

#############################################################

# jmx

#############################################################

spring.jmx.default-domain 指定JMX domain name.

spring.jmx.enabled 是否暴露jmx，默认为true

spring.jmx.server 指定MBeanServer bean name. 默认为: mbeanServer)

#############################################################

# mail

#############################################################

spring.mail.default-encoding 指定默认MimeMessage的编码，默认为: UTF-8

spring.mail.hos t指定SMTP server host.

spring.mail.jndi-name 指定mail的jndi名称

spring.mail.password 指定SMTP server登陆密码.

spring.mail.port 指定SMTP server port.

spring.mail.properties 指定JavaMail session属性.

spring.mail.protocol 指定SMTP server使用的协议，默认为: smtp

spring.mail.test-connection 指定是否在启动时测试邮件服务器连接，默认为false

spring.mail.username 指定SMTP server的用户名.

#############################################################

# sendgrid

#############################################################

spring.sendgrid.password 指定SendGrid password.

spring.sendgrid.proxy.host 指定SendGrid proxy host.

spring.sendgrid.proxy.port 指定SendGrid proxy port.

spring.sendgrid.username 指定SendGrid username.

#############################################################

# social

#############################################################

spring.social.auto-connection-views 是否开启连接状态的视图，默认为false

spring.social.facebook.app-id 指定应用id

spring.social.facebook.app-secret 指定应用密码

spring.social.linkedin.app-id 指定应用id

spring.social.linkedin.app-secret 指定应用密码

spring.social.twitter.app-id 指定应用ID.

spring.social.twitter.app-secret 指定应用密码