# spring boot

## spring boot 源码

### 重点单词或短语

|  |  |  |
| --- | --- | --- |
| bootstrap:引导  launch: 启动  by default: 默认情况下  perform: 执行  performance: 性能  additional: 额外的,其他的…一样  a number of: 多个,一些  verify: 验证  obvious: 明显,变得明显  further: 进一步的,更多的  kick in: 开始生效  accordingly: 相应地  and then: 然后  enforce: 强制  halt: 停止  explicit: 明确的  regardless: 无论如何,不管  use: 使用  prefer to: 宁愿  background: 后台  foreground: 前台  as soon as: 一…就,一旦…  since: 因为  derived: 派生的,衍生的  ultimately: 最终  precise: 精确的,准确的  efficient: 有效的  potentially: 潜在的  populate: 填充  check against: 检测(against不翻译)  as of: 从…开始 | appropriate: 适当的,恰当的  In most circumstances: 在大多数情况下  for more advanced: 对于更高级的…  generally: 通常的  generically: 一般的  generic:通用的,泛型  stop watch: 秒表  likelihood: 可能性  rather than: 而不是  in addition to: [除了…之外,还…]  encapsulate: 封装  regular: 常规的  in turn: 反过来,继而  raise: 引发,触发  from there: 从哪里  weak: 弱  along with: 随着…,与…一起,以及  logging: 记录  entry: 条目,进入  useful: 有用的  preinitializer: 预初始化器  time consuming: 耗时  conceivably: 尽…, 想得到地  beware: 要注意,当心  later: 后期,在..后期,在…后  navigation: 导航  propagated: 传播的  retrieval: 检索  suppose: 假设  candidate: 候选者  supersede: 取代 | recommended: 推荐,建议  bound: 绑定  indicate: 表明,指示  flag: 标识,标志  timing: 时间,计时  normally: 通常地  proof: 证明,验证  facility: 工具,设施  represent: 代表  detection: 检测  across: 在…之间  grab:抓取,获得  listen: 监听  reference: 引用  as well as: 以及  reacts: 响应  side effect: 副作用  if any: 如果有  mechanism: 机制  as early as conceivably possible: 尽可能早的…  as early as: 早的,早在  stage: 阶段  make sense: 有道理,有意义  encapsulates: 封装  convey: 传达  a set of: 一组  a specific set of: 一组指定的  purpose: 目的  introspect: 内省,反思  variant: 变形,变体 |

### 重点介词的可能的翻译

|  |
| --- |
| to: 到,以,为了 |
| for: 为了 |
| from: 从… |
|  |
|  |

### 难翻译句笔记

1. class#SpringApplication

|  |
| --- |
| In most circumstances the static run(Class, String []) method can be called directly from your main method to bootstrap your application  大多数情况下,从你的main方法引导你的应用可以直接调用静态的run方法 |
| 翻译解析: can be called : 可知该语句为一个被动语句,因此翻译是可以将一个被动语法翻译为陈述句,断句方式如下  状语(介词短语部分)🡪谓语🡪主语 |

|  |
| --- |
| for more advanced configuration a SpringApplication instance can be created and customized before being run:  对于更高级的配置,在运行之前可以创建并且自定义一个 SpringApplication 实例 |
| 翻译解析: can be created: 可知该语句为一个被动语句,因此翻译是可以将一个被动语法翻译为陈述句,断句方式如下  状语(介词短语部分)🡪谓语🡪主语 |

|  |
| --- |
| This makes it possible to set SpringApplication properties dynamically, like additional sources ("spring.main.sources" - a CSV list) the flag to indicate a web environment ("spring.main.web-application-type=none") or the flag to switch off the banner ("spring.main.banner-mode=off").  这使得它可以动态设置SpringApplication属性,像其他源(“spring.main.sources” – 一个csv列表)一样,这个标识表明一个web环境(spring.main.web-application-type=none)或者 这个标识关闭横幅(“spring.main.banner-mode=off”) |

1. class#StopWatch

|  |
| --- |
| This class is normally used to verify performance during proof-of-concepts and in development, rather than as part of production applications.  这个类通常被用于在开发过程中性能验证和概念验证期间,而不是作为生产应用程序的一部分. |
| 翻译解析: 一个句子,再有介词部分时,在翻译完主谓语后就应该接着翻译介词部分. proof-of-concepts : 单词之间使用 – 分隔的句子,按照正常句子翻译即可. |

class#StopWatch#starting()

|  |
| --- |
| Start an unnamed task. The results are undefined if stop() or timing methods are called without invoking this method.  开始一个没有命名的任务,如果没有调用这个方法的情况下stop() 或者计时方法被调用,这个结果是没有被定义的 |
| 翻译解析: 一个句子中如果有 if 从句,要先翻译 if 从句,if句子中如果有介词部分,if翻译后接着翻译介词部分 |

1. class#ConfigurableApplicationContext

|  |
| --- |
| Provides facilities to configure an application context in addition to the application context client methods in the ApplicationContext interface.  除了在ApplicationContext接口中的应用程序上下文客户端方法之外,还提供配置一个应用上下文的工具 |

1. class#BeanDefinitionRegistryPostProcessor

|  |
| --- |
| allowing for the registration of further bean definitions before regular BeanFactoryPostProcessor detection kicks in.  允许在常规BeanFactoryPostProcessor检测开始生效之前进一步的注册bean定义 |

1. class#ApplicationListener

|  |
| --- |
| an ApplicationListener can generically declare the event type that it is interested in.  ApplicationListener一般可以声明他感兴趣的事件类型 |
| 翻译解析: that it is interested in 是修饰the event type的定语从句 |

|  |
| --- |
| When registered with a Spring ApplicationContext, events will be filtered accordingly, with the listener getting invoked for matching event objects only.  当用一个spring applicationContext注册时,相应的事件将被过滤,且仅仅调用匹配的事件对象监听器 |
| 翻译解析: 形容词修饰名词,副词修饰动词, accordingly:相应的(形容词),only:仅仅(副词) |

1. class#ParentContextCloserApplicationListener

|  |
| --- |
| Listener that closes the application context if its parent is closed. It listens for refresh events and grabs the current context from there, and then listens for closed events and propagates it down the hierarchy.  如果父亲被关闭,监听器则关闭应用上下文,它监听刷新事件并且从哪里获得当前上下文,然后监听关闭事件并且传播它到下面的层次结构中. |

1. class#ClasspathLoggingApplicationListener

|  |
| --- |
| A SmartApplicationListener that reacts to environment prepared events and to failed events by logging the classpath of the thread context class loader (TCCL) at DEBUG level.  一个SmartApplicationListener,它通过在debug级别记录线程上下问题类加载器的类路径来响应环境准备事件和失败事件 |
| 翻译解析: A SmartApplicationListener that : 单独进行翻译,后面的定语从句用 ‘它’ 作为翻译主语,介词从句若还有介词语句,要先翻译介词语句在翻译介词从句 |

1. class#LoggingApplicationListener

|  |
| --- |
| LOG\_FILE is set to the value of path of the log file that should be written (if any).  LOG\_FILE 设置应该写入的日志文件路径的值(如果有) |
| 翻译解析: 因为 log file 后的 that定语从句是修饰 log file的,所以要先翻译该从句 |

1. class#

|  |
| --- |
| The source of the event is the SpringApplication itself, but beware of using its internal state too much at this early stage since it might be modified later in the lifecycle.  事件源是SpringApplication自己,但要注意在早期阶段使用它的内部状态太多,因为在它的生命周期后期它可以被修改 |

1. class#ResolvableType

|  |
| --- |
| Encapsulates a Java java.lang.reflect.Type, providing access to supertypes, interfaces, and generic parameters along with the ability to ultimately resolve to a java.lang.Class.  封装一个java Type类型,提供超类型,接口和泛型参数的访问,以及最终解析为一个Class能力 |
| 翻译解析: 翻译该句子时要注意对介词的翻译 |

## spring boot 官方文档