SpringBoot应用配置项加密

2017/12/11 0:00 - Posted By 起衣 - Tags: java,spring - Category: Code

2 条评论

在实际的应用当中，我们可能会存在一些需要加密存储的配置项，最典型的就是数据库的密码。推荐的一种方案是 jasypt-spring-boot 项目，不过 Spring Cloud 本身也提供了对加密配置项的支持。

首先引入 spring-cloud-context 的依赖 org.springframework.cloud:spring-cloud-context:1.2.0.RELEASE, 然后在 resources 目录下新建 bootstrap.properties, 里面写上加密数据库的密钥内容, 由于默认情况下启用的是 AES128 的算法，因此密钥长度为 16，

```
encrypt.key=nxN8cE/CoUxxexPh
```

接下来，就可以在 application.properties 文件里直接写上加密后的密文了，默认情况下加解密算法为 AES/CBC/PKCS5Padding, 使用的 salt 为 deadbeef(16进制字符串格式), 由于 AES 加密结果为字节数组，并不一定是可见字符，所以结果需要再次转换成16进制的字符串。配置格式为 key={cipher}加密后的hex字符串，由 {cipher} 开头，表明后面是加密的配置值，样例如:

```
spring.datasource.password={cipher}fb748dce88b94fb7d84a9f32e6b5d51729049792d1ee38e2240b176ec5db91cb
```

Spring Boot 应用启动时，spring-cloud-context 模块会注册一个 BootstrapConfiguration，BootstrapConfiguration 会在 SpringBoot 应用的很早期进行加载，加载时会检测 Environment 里是否存在 encrypt.key 的配置项，如果存在就将其作为加解密的密钥，然后再次注册一个 ApplicationInitializer，由这个 ApplicationInitializer 在 SpringContext 初始化过程当中，对所有配置项值为 {cipher} 开头的配置项值进行解密，解密的结果添加到一个新的 decrypted 的 PropertySource，插入到 Environment 的最前面，于是后面的应用、Bean 初始化过程当中，直接可以根据配置项名称获取到解密的结果了。

用起来特别简单，spring-cloud-context 也基于 Spring 已有的机制，几乎无缝、透明地提供了这个能力，不过直接把密钥的明文写到配置文件里也不太安全，推荐的做法是将密钥分开存储到配置文件和代码里，应用启动时再把两部分呢拼接起来，以便提高安全性，可以参考 PinkApp 的做法。

相关日志 Relate Posts

- SpringBoot应用集成etcd配置源
- SpringBoot 集成 Flyway 自动创建数据库表
- 根据条件过滤日志输出
- retrofit 服务自定义签名认证

- retrofit 发起form data请求时使用自定义对象参数

“SpringBoot应用配置项加密”2条留言

hobey: December 11, 2017 at 8:31 pm

Reply

起衣: December 11, 2017 at 10:47 pm

@hobey 卧槽线程

Reply

发表留言（Ctrl+Enter提交）

昵称 (必填)

EMail (不公开，必填)

网站 (选填)

发表