

# Bigdata-配置加解密

Bigdata

加密

设计

架构

## jasypt 的解决方案

### 1. Maven依赖

```
<dependency>
  <groupId>com.github.ulisesbocchio</groupId>
  <artifactId>jasypt-spring-boot-starter</artifactId>
  <version>2.1.0</version>
</dependency>
```

### 2. 配置

```
jasypt:
  encryptor:
    password: boss #根密码
```

### 3. 调用JAVA API 生成密文

```
public class BosxAdminApplicationTest {
    /**
     * jasypt.encryptor.password 对应 配置中心 application-dev.yml 中的密码
     */
    @Test
    public void testEnvironmentProperties() {
        System.setProperty("jasypt.encryptor.password", "boss");
        StringEncryptor stringEncryptor = new DefaultLazyEncryptor(new StandardEnvironment());
        //加密方法
        System.out.println(stringEncryptor.encrypt("123456"));
        //解密方法
        System.out.println(stringEncryptor.decrypt("VZ5XhsD3fxa+1NvkSmwp4w=="));
    }
}
```

### 4. 配置文件中使使用密文

```
spring:
```

```
datasource:  
  password: ENC(密文)  
xxx: ENC(密文)
```