

# Spring Cloud 配置

---

## Java 中配置

---

### 字符类型配置

- 通用（内置）
  - Java 系统属性（`System#getProperties()`）
  - OS 环境变量（`System#getenv()`）
- 特别（应用）
  - XML（JDK API 可处理）
    - Spring DOM API
      - `org.w3c.dom.Document`
    - JAXB (Spring XML 序列化)
    - SAX (Simple API for XML)
    - XML Stream
    - XStream
  - Properties（JDK API 可处理）
    - Key Value 配置模式
    - XML 配置模式
  - .ini（JDK Properties API 可处理）
  - JSON（第三方）
  - YAML（第三方）
  - CSV（自定义、第三方）

- Hard Code (硬编码)

## 非字符类型配置扩展

- 通用
  - Integer
  - Boolean
  - Long

结论：JDK 提供底层配置源的存储方式，没有具体抽象配置 API，仅提供了一些零散的配置类型转换

## Apache Commons Configuration API

---

Apache Commons 通用工程集合，它对 JDK API 做出一定的补充，commons-lang StringUtils, commons-collection CollectionUtils, commons-dbcp BasicDataSource

commons-configuration 提供了统一配置 API，提供通用数据类型转换、Integer、Long、BigDecimal

## 核心接口

- org.apache.commons.configuration.Configuration
  - 关系型数据库 -  
org.apache.commons.configuration.DatabaseConfiguration

- Properties 配置文件 -  
org.apache.commons.configuration.PropertiesConfiguration
- XML 配置文件 -  
org.apache.commons.configuration.XMLConfiguration
- OS 环境变量 -  
org.apache.commons.configuration.EnvironmentConfiguration
- Java System Properties -  
org.apache.commons.configuration.SystemConfiguration
- 特殊组合实现 -  
org.apache.commons.configuration.CompositeConfiguration
  - 自身不提供数据来源，组合其他子来源，并且控制优先级顺序

以上实现均有外部提供配置的来源

## 举例说明

- CompositeConfiguration ( List 有序)
  - DatabaseConfiguration
    - user.age = 34
  - PropertiesConfiguration
    - user.age = 30
  - XMLConfiguration
  - EnvironmentConfiguration
    - user.age = 19
  - SystemConfiguration

CompositeConfiguration -> Configuration.getInteger("user.age") ->

- 34
- 19

```
String key = "user.age";
for(Configuration configuration : getConfigurations()){
    //
    Integer value = configuration.getInteger(key);
    if(value != null){
        return value;
    }
}
```

## 通用需求

- 读取配置源
- 合并配置源
- 定义顺序

结论：Apache commons-configuration 提供一套完整配置 API，并且能够通过组合模式组合多种不同的配置来源，来实现统一的配置读取，不支持动态更新和写入

## Netflix Archaius

Netflix Archaius 扩展 Apache Commons Configuration，实现动态更新和写入等操作

## 核心 API

- `com.netflix.config.DynamicConfiguration`

结论：Netflix Archaius 弥补了 Apache Commons Configuration 不足，不过同时它尚未提供类型转换的 API 实现

# Spring Framework Environment 抽象

---

## Environment 抽象接口

### 语义

- 存储配置
- 存储Profile
- 转换类型

### API

- 只读 API - `org.springframework.core.env.Environment`
  - 配置 - `org.springframework.core.env.PropertyResolver`
  - Profile
- 写入 API - `org.springframework.core.env.ConfigurableEnvironment`
  - 配置 - 获取 `MutablePropertySources`
  - Profile - `setProfile*`

[0] = {}

[0] addLast SystemProperties

[1] addLast OS Environment

[2]

Spring Boot application.properties / application.xml or  
application.yml / application.yaml

## 配置源

### API

- 单配置源 - org.springframework.core.env.PropertySource
- 多配置源 - org.springframework.core.env.PropertySources

### 注解

- 单配置源 -  
@org.springframework.context.annotation.PropertySource
- 多配置源 -  
@org.springframework.context.annotation.PropertySources

## Spring Profile

### API

- 读
  - org.springframework.core.env.AbstractEnvironment#getActiveProfiles
  - org.springframework.core.env.AbstractEnvironment#getDefaultProfiles
- 写
  - org.springframework.core.env.AbstractEnvironment#setActiveProfiles
  - org.springframework.core.env.AbstractEnvironment#setDefaultProfiles

# 类型转换

## API

- org.springframework.core.convert.ConversionService
- org.springframework.core.convert.converter.Converter

```
@Value("${user.age}")
```

```
private Integer age;
```

## Spring Boot 配置实现

---

### 配置来源

核心实现 -

```
org.springframework.boot.context.config.ConfigFileApplicationList  
ener
```

## Spring Cloud 中的配置

---

### 配置来源

核心实现 -

```
org.springframework.cloud.bootstrap.BootstrapApplicationListene  
r
```

问题

如何控制读取远程配置在本地配置读取之前

## 相关工程

---

- Spring Cloud Alibaba - <https://github.com/alibaba/spring-cloud-alibaba>
- Nacos Spring Project - <https://github.com/nacos-group/nacos-spring-project>