

# 第十六章：Spring 泛型处理

小马哥 • mercyblitz

# Spring 泛型处理

---

1. Java 泛型基础
2. Java 5 类型接口
3. Spring 泛型类型辅助类
4. Spring 泛型集合类型辅助类
5. Spring 方法参数封装 - MethodParameter
6. Spring 4.0 泛型优化实现 - ResolvableType
7. ResolvableType 的局限性
8. 面试题精选



# Java 泛型基础

- 泛型类型
  - 泛型类型是在类型上参数化的泛型类或接口
- 泛型使用场景
  - 编译时强类型检查
  - 避免类型强转
  - 实现通用算法

# Java 泛型基础

- 泛型类型擦写
  - 泛型被引入到 Java 语言中，以便在编译时提供更严格的类型检查并支持泛型编程。类型擦除确保不会为参数化类型创建新类；因此，泛型不会产生运行时开销。为了实现泛型，编译器将类型擦除应用于：
    - 将泛型类型中的所有类型参数替换为其边界，如果类型参数是无边界的，则将其替换为“Object”。因此，生成的字节码只包含普通类、接口和方法
    - 必要时插入类型转换以保持类型安全
    - 生成桥方法以保留扩展泛型类型中的多态性

## Java 5 类型接口

- Java 5 类型接口 – `java.lang.reflect.Type`

派生类或接口	说明
<code>java.lang.Class</code>	Java 类 API, 如 <code>java.lang.String</code>
<code>java.lang.reflect.GenericArrayType</code>	泛型数组类型
<code>java.lang.reflect.ParameterizedType</code>	泛型参数类型
<code>java.lang.reflect.TypeVariable</code>	泛型类型变量, 如 <code>Collection&lt;E&gt;</code> 中的 <code>E</code>
<code>java.lang.reflect.WildcardType</code>	泛型通配类型

## Java 5 类型接口

- Java 泛型反射 API

类型	API
泛型信息 (Generics Info)	<code>java.lang.Class#getGenericInfo()</code>
泛型参数 (Parameters)	<code>java.lang.reflect.ParameterizedType</code>
泛型父类 (Super Classes)	<code>java.lang.Class#getGenericSuperclass()</code>
泛型接口 (Interfaces)	<code>java.lang.Class#getGenericInterfaces()</code>
泛型声明 (Generics Declaration)	<code>java.lang.reflect.GenericDeclaration</code>

# Spring 泛型类型辅助类

- 核心 API – `org.springframework.core.GenericTypeResolver`
  - 版本支持: [2.5.2 , )
  - 处理类型相关 (Type) 相关方法
    - `resolveReturnType`
    - `resolveType`
  - 处理泛型参数类型 (ParameterizedType) 相关方法
    - `resolveReturnTypeArgument`
    - `resolveTypeArgument`
    - `resolveTypeArguments`
  - 处理泛型类型变量 (TypeVariable) 相关方法
    - `getTypeVariableMap`

# Spring 泛型集合类型辅助类

- 核心 API – `org.springframework.core.GenericCollectionTypeResolver`
  - 版本支持: [2.0 , 4.3]
  - 替换实现: `org.springframework.core.ResolvableType`
  - 处理 Collection 相关
    - `getCollection*Type`
  - 处理 Map 相关
    - `getMapKey*Type`
    - `getMapValue*Type`



# Spring 方法参数封装

- 核心 API – `org.springframework.core.MethodParameter`
  - 起始版本: [2.0 , )
  - 元信息
    - 关联的方法 – `Method`
    - 关联的构造器 – `Constructor`
    - 构造器或方法参数索引 – `parameterIndex`
    - 构造器或方法参数类型 – `parameterType`
    - 构造器或方法参数泛型类型 – `genericParameterType`
    - 构造器或方法参数参数名称 – `parameterName`
    - 所在的类 – `containingClass`

# Spring 4.0 泛型优化实现 - ResolvableType

- 核心 API - `org.springframework.core.ResolvableType`
  - 起始版本: [4.0 , )
  - 扮演角色: `GenericTypeResolver` 和 `GenericCollectionTypeResolver` 替代者
  - 工厂方法: `for*` 方法
  - 转换方法: `as*` 方法
  - 处理方法: `resolve*` 方法

# ResolvableType 的局限性

- 局限一：ResolvableType 无法处理泛型擦写
- 局限二：ResolvableType 无法处理非具体化的 ParameterizedType

# 面试题

沙雕面试题 - Java 泛型擦写发生在编译时还是运行时?

答：运行时



我真的没笑

# 面试题

996 面试题 - 请介绍 Java 5 Type 类型的派生类或接口‘



答：

- `java.lang.Class`
- `java.lang.reflect.GenericArrayType`
- `java.lang.reflect.ParameterizedType`
- `java.lang.reflect.TypeVariable`
- `java.lang.reflect.WildcardType`

# 面试题

**劝退面试题** – 请说明 ResolvableType 的设计优势？



答：

- 简化 Java 5 Type API 开发，屏蔽复杂 API 的运用，如 ParameterizedType
- 不变性设计（Immutability）
- Fluent API 设计（Builder 模式），链式（流式）编程