

# 第十九节 数据架构升级

---

## MyBatis

---

### 配置

### XML 配置

### XML 结构定义

- XML Schema
- DTD

Schema - JAX

JAXB 能够解析 Schema，也能解析 DTD，Java 6 开始引入到 JDK

# 支持特性

## 占位符

## Profile

# 拦截器 (Interceptor)

## 缓存连接器 -

**org.apache.ibatis.executor.CachingExecutor**

MyBatis Cache 适配 Spring Caching 接口，桥接 Java Caching。

Java Caching -> Spring Caching -> MyBatis Cache

# TypeAliasRegistry

提供 Alias 与实现扩展

# SqlSessionFactoryBuilder

生成 SqlSessionFactory，并且仅支持 XML 配置方式

- XmlConfigBuilder

- Configuration
- Environment

## SqlSessionFactory

SqlSessionFactory 相当于 java.sql.Connection 封装

## 相关技术

---

### JDK 动态代理

字节码提升实现

### MapStruct

<https://mapstruct.org/>

基于 APT 实现

### W3C School

DTD

XML Schema

# 面试题

---

**T1、T2、T3 线程，按照这个次序输出  
Hello,World**

不能用 join

wait(0)

POSIX Thread

JOINABLE

T1.setPriority

T2.setPriority

T3.setPriority

T1. started

# 其他

---

Antx : XML.vm -> antx.properties -> XML

## 思考题

org.apache.ibatis.reflection.factory.ObjectFactory 方法实现:

```
<T> T create(Class<T> type, List<Class<?>>  
    constructorArgTypes, List<Object>  
    constructorArgs)
```

为什么不这样设计?

```
<T> T create(Class<T> type, Object...  
    constructorArgs);
```

```
public class User {  
  
    public User(CharSequence value) { // #1  
  
    }  
  
    public User(String value) { // #2  
  
    }  
  
    public User(char[] value) { // #3  
  
    }  
}
```

new User("Hello,World");

Constructor.newInstance("Hello,World"); // 运行时  
java.lang.String 的确是 CharSequence

"Hello,World".getClass() == java.lang.String !=  
java.lang.CharSequence

```
Class<?>[] constructorArgTypes; // [0] ==  
CharSequence.class  
  
constructorArgTypes[0] = String.class;  
  
Object[] parameters;
```