# PandasとJavaを用いたXML解析結果の統合とSQLテーブルの紐づけ

## 1. 目的

本資料では、Pandasを用いたXML解析結果とJavaを用いた解析結果を統合し、Business Logic（ビジネスロジック）からSQLのテーブルまでの関連性を可視化する方法を解説する。

## 2. XML解析の概要

SpringやMyBatisを利用したシステムでは、XMLファイルを通じてBeanの定義やSQLマッピングが行われる。Pandasを用いてXML解析を行い、Beanの依存関係やMyBatisのSQL定義を抽出する。

## 3. Pandasを用いた解析（Python）

以下の手順でPandasを利用し、XMLファイルを解析する。

1. lxmlを使用してXMLファイルを読み込み  
2. XPathを用いてBean定義やSQLマッピングを抽出  
3. PandasのDataFrameにデータを格納し、構造化  
4. MyBatisのnamespace, SQL ID, SQL内容をDataFrameに整理

サンプルコード（PythonによるXML解析）:  
```python  
import pandas as pd  
from lxml import etree  
  
# XMLの読み込み  
tree = etree.parse("beans.xml")  
  
# Bean定義の取得  
beans = []  
for bean in tree.xpath("//bean"):  
 beans.append({"Bean ID": bean.get("id"), "Class": bean.get("class")})  
  
# DataFrameに変換  
df\_beans = pd.DataFrame(beans)  
print(df\_beans)  
```

## 4. Javaを用いた解析

JavaのDOMパーサを利用し、MyBatisのSQL IDと関連テーブルを取得する。

サンプルコード（JavaによるSQL ID取得）:  
```java  
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
DocumentBuilder builder = factory.newDocumentBuilder();  
Document doc = builder.parse(new File("mapper.xml"));  
NodeList sqlNodes = doc.getElementsByTagName("select");  
for (int i = 0; i < sqlNodes.getLength(); i++) {  
 Element sqlElement = (Element) sqlNodes.item(i);  
 System.out.println("SQL ID: " + sqlElement.getAttribute("id"));  
}  
```

## 5. DataFrameとJava解析結果の統合

PythonのPandasを用いて取得したBean情報と、Javaで解析したSQL情報を統合することで、Business LogicからSQLテーブルまでの関連性を明確にする。

統合のポイント:  
・PandasのDataFrameを用いてBeanとSQLの対応関係を管理  
・namespace, SQL ID, SQL内容を結びつける  
・Javaの解析結果をDataFrameに追加し、SQLの実行対象テーブルを明確化

サンプルコード（Pythonによる統合処理）:  
```python  
# Javaの解析結果をDataFrameに追加  
df\_sql = pd.DataFrame([  
 {"Namespace": "com.example.mapper.UserMapper", "SQL ID": "getUserById", "SQL": "SELECT \* FROM users WHERE id = #{id}"}  
])  
  
# 統合（BeanとSQLの対応関係をマージ）  
df\_merged = df\_beans.merge(df\_sql, left\_on="Class", right\_on="Namespace", how="left")  
print(df\_merged)  
```

## 6. Business Logic から SQLテーブルの紐づけ

最終的に、Business Logic（Javaのクラス）とMyBatisのSQL、データベースのテーブルを紐づける。これにより、システム内の処理の流れを可視化できる。

可視化例:  
```plaintext  
[Business Logic] → [Bean] → [MyBatis SQL] → [Database Table]  
UserService → userService Bean → getUserById() → usersテーブル  
```