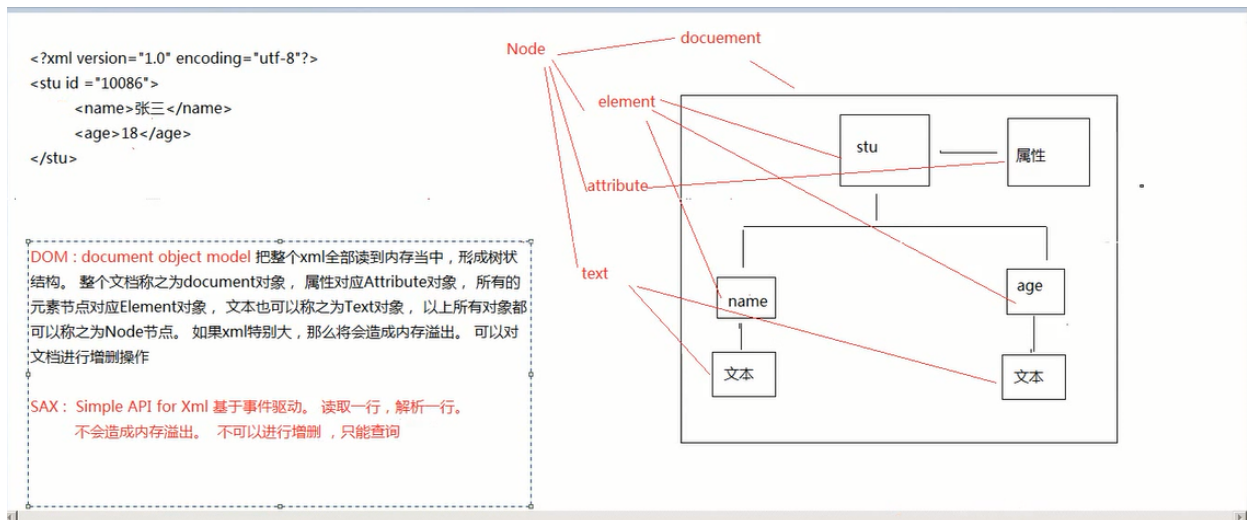


## 回顾DOM解析



DOM解析将一个xml拆分为多个不同的分段

## 解析XML文件

需求: dom4j-1.6.1.jar、jaxen-1.1-beta-6.jar

### XML中的文件内容

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <c3p0-config>
3   <default-config>
4     <property name="driverClass">com.mysql.jdbc.Driver</property>
5     <property name="jdbcUrl">jdbc:mysql://localhost:3306/testBase?useSSL=false</property>
6     <property name="user">root</property>
7     <property name="password">123</property>
8   </default-config>
9 </c3p0-config>
10
11
```

首先想解析得先了解dom4j中的一个类 **SAXReader**

构造方法:

```
public SAXReader() //有很多构造方法, 但我们有一个无参就够了
```

常用方法:

```
public Document read(File file) //通过read获取到整个文档内容/document对象
```

容/document对象

### 第二个类 **Document**

通过以上类获取到Document对象实例

常用方法:

```
public Element getRootElement() //获取文档头 也就是c3p0-config
```

```
public String getXMLEncoding() //获取该xml的字符编码
```

### 第三个类 `Element`

```
public Attribute attribute(String name) //获取当前Element中的属性值
public Element element(String name) //获取当前标签下子标签name的element对象
public List elements() //获取当前标签下全部子标签的list<Element>对象
public List elements(String name) //获取一组name元素
public String getText() //返回该标签中的文本内容           如: <property>内容
</property>
public void setXMLEncoding(String encoding) //设置编码
```

### 一个小的案例(获取以上xml中的文本内数据):

```
public void test(){
    SAXReader reader = new SAXReader();
    Document doc = reader.read(new File("xml地址"));
    Element root = doc.getRootElement();
    Element root_child = root.element("default-config");
    List<Element> list = root_child.elements();
    for(Element value: list){
        System.out.println(value.getText());
    }
}
```

结果:

```
/*
com.mysql.jdbc.Driver
jdbc:mysql://localhost:3306/testBase?useSSL=false
root
123
*/
```

以上属于一层一层找,但也可以不必那么麻烦一层一层找下去

### 升级案例:

```
public void test(){
    SAXReader reader = new SAXReader();
    Document doc = reader.read(new File("xml地址"));
    Element root = doc.getRootElement();
    List<Element> list = root.selectNodes("//default-config//property");
    for(Element value:list){
```

```
        System.out.println(value.getText());
    }
}
```

解析:

目标: 获取property层的标签 执行过程: 我是先获取头标签, 然后想拿到property标签的话得先经过default-config, 它下面才是property 所以路径是//default-config//property, 然后就调用selectNodes获取该层property的全部标签, 遍历打印property中的内容

使用方法:

public Node selectSingleNode(String xpathExpression) 获取查找到的第一个元素

public List selectNodes(String xpathExpression) 获取查找到的全部元素