

```

package com.sxt.server.basic;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

/**
 * 熟悉SAX解析流程
 * @author 江
 *
 */

public class XmlTest04 {
    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        //SAX解析
        //1. 获取解析工厂
        SAXParserFactory factory=SAXParserFactory.newInstance();
        //2. 从解析工厂获取解析器
        SAXParser parse=factory.newSAXParser();
        //3. 编写处理器
        //4. 加载文档Document注册处理器
        PersonHandler handler=new PersonHandler();
        //5. 解析

        parse.parse(Thread.currentThread().getContextClassLoader().getResourceAsStream("com/sxt/server/basic/p.xml"), handler);

        List<Person03> persons=handler.getPersons();
        for(Person03 p:persons){
            System.err.println(p.getName()+"-->"+p.getAge());
        }
    }
}

class PersonHandler extends DefaultHandler{
    private List<Person03> persons;
    private Person03 person;
    private String tag; //存储操作标签

    public void startDocument() throws SAXException{
        persons=new ArrayList<>();
    }
}

```

```

    }

    @Override
    public void startElement(String uri, String localName, String qName, Attributes attributes) throws
SAXException {
        if(null!=qName) {
            tag=qName; //存储标签名
            if(tag.equals("person")) {
                person=new Person03();
            }
        }
    }

    @Override
    public void characters(char[] ch, int start, int length) throws SAXException {
        String contends=new String(ch,start,length).trim();
        if(null!=tag){ //处理了空
            if(tag.equals("name")){
                person.setName(contends);
            }else if(tag.equals("age")){
                if(contends.length()>0){
                    person.setAge(Integer.valueOf(contends));
                }
            }
        }
    }

    @Override
    public void endElement(String uri, String localName, String qName) throws SAXException {
        if(null!=qName) {
            if(qName.equals("person")) {
                persons.add(person);
            }
        }
        tag=null;
    }

    @Override
    public void endDocument() throws SAXException {
    }

    public List<Person03> getPersons() {
        return persons;
    }

}

```

