```
package com.sxt.server.basic.servlet;

import java.io.IOException;
import java.lang.reflect.InvocationTargetException;
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;

import com.sxt.server.basic.Person03;

/**
 * 熟悉SAX解析流程
 * @author 江
 *
 */

public class XmlTest {
  public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException,
InstantiationException, IllegalAccessException, IllegalArgumentException, InvocationTargetException,
NoSuchMethodException, SecurityException, ClassNotFoundException {
        //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
          WebHandler handler=new WebHandler();
          //5.解析

parse.parse(Thread.currentThread().getContextClassLoader().getResourceAsStream("com/sxt/server/basic/servlet/web.xml"),han

          List<Entity> entitys=handler.getEntitys();
          for(Entity entity:entitys){
                  System.err.println(entity.getName()+"-->"+entity.getClz());
          }

          List<Mapping> mappings=handler.getMappings();
          for(Mapping mapping:mappings) {
                  System.err.println(mapping.getName()+"-->"+mapping.getPatterns());
          }

          //获取数据
          WebContext context=new WebContext(handler.getEntitys(),handler.getMappings());
          //假设你输入/login
```

```
            String className=context.getClz("/login");
            Class clz=Class.forName(className);
            Servlet servlet=(Servlet)clz.getConstructor().newInstance();
            System.err.println(servlet);
            servlet.service();
}
}

class WebHandler extends DefaultHandler{
        private List<Entity> entitys;
        private List<Mapping> mappings;
        private Entity entity;
        private Mapping mapping;
        private String tag;  //存储操作标签
        private boolean isMapping=false;

        public void startDocument() throws SAXException{
            entitys=new ArrayList<Entity>();
            mappings=new ArrayList<Mapping>();


        }

        @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("servlet")) {
                                entity=new Entity();
                                isMapping=false;
                        }if(tag.equals("servlet-mapping")) {
                                mapping=new Mapping();
                                isMapping=true;
                        }
                }
        }

        @Override
        public void characters(char[] ch, int start, int length) throws SAXException {
                String contents=new String(ch,start,length).trim();
                if(null!=tag){  //处理了空
                        if(isMapping) {//操作servlet-mapping
                                if(tag.equals("url-pattern")) {
                                        mapping.addPattern(contents);
                                }else if(tag.equals("servlet-name")){
                                        mapping.setName(contents);
                                }
                        }else {//操作servlet
                                if(tag.equals("servlet-name")) {
                                        entity.setName(contents);
                                }else if(tag.equals("servlet-class")) {
                                        entity.setClz(contents);
```

```
                        }
                    }
                }
        }

        @Override
        public void endElement(String uri, String localName, String qName) throws SAXException {
                if(null!=qName) {
                  if(qName.equals("servlet")) {
                          entitys.add(entity);
                          entity=new Entity();
                  }else if(qName.equals("servlet-mapping")) {
                          mappings.add(mapping);
                          mapping=new Mapping();
                  }
                }
                tag=null;
        }

        public List<Entity> getEntitys() {
                return entitys;
        }


        public List<Mapping> getMappings() {
                return mappings;
        }



        @Override
        public void endDocument() throws SAXException {
        }




}
```