

第 3 章



Web数据抽取

本章学习目标

- 了解数据抽取的概念
- 了解 Web 数据抽取的原理
- 掌握使用 Kettle 实现 Web 数据抽取的应用

3.1 Web 数据抽取基础

1. 数据抽取概述

数据抽取是指把数据从数据源读出来,一般用于从源文件和源数据库中获取相关的数据,也可以从 Web 数据库中获取相关数据。

数据抽取的两个常用方式是全量抽取和增量抽取。全量抽取类似于数据迁移或数据复制,它将数据源中的表或视图的数据原封不动地从数据库中抽取出来,并转换成自己的 ETL 工具可以识别的格式,全量抽取通常比较简单。增量抽取只抽取自上次抽取以来数据库中要抽取的表中新增或修改的数据。在 ETL 使用过程中,增量抽取较全量抽取应用更广。

图 3-1 显示了从数据库中抽取数据的过程。一般来说,在进行数据抽取的时候,需要根据不同情况制定不同的抽取规则,再将抽取规则转换为对应的查询语句实现。

目前增量数据抽取中常用的捕获变化数据的方法主要有以下 5 种。

1) 触发器方式

触发器方式是指在要抽取的表上建立需要的触发器,一般要建立插入、修改、删除 3 个触发器,每当源表中的数据发生变化,变化的数据就被相应的触发器写入一个临时表,

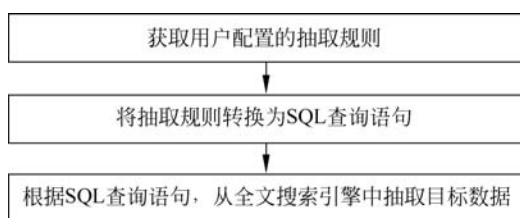


图 3-1 从数据库中抽取数据的过程

抽取线程从临时表中抽取数据,临时表中抽取过的数据被标记或删除。

优点:数据抽取的性能高,ETL 加载规则简单,速度快,不需要修改业务系统表结构,可以实现数据的递增加载。

缺点:要求业务表建立触发器,对业务系统有一定的影响。

2) 时间戳方式

时间戳方式是一种基于快照比较的变化数据捕获方式,在源表上增加一个时间戳字段,系统中更新修改表数据的时候,同时修改时间戳字段的值。当进行数据抽取时,通过比较系统时间与时间戳字段的值决定抽取哪些数据。有的数据库的时间戳支持自动更新,即表的其他字段的数据发生改变时,自动更新时间戳字段的值。有的数据库不支持时间戳的自动更新,这就要求业务系统在更新业务数据时手工更新时间戳字段。

优点:与触发器方式一样,时间戳方式的性能也比较好,ETL 系统设计清晰,源数据抽取相对清楚简单,可以实现数据的递增加载。

缺点:时间戳维护需要由业务系统完成,对业务系统也有很大的侵入性(加入额外的时间戳字段),特别是对不支持时间戳的自动更新的数据库,还要求业务系统进行额外的更新时间戳操作,工作量大,改动大,风险大;另外,无法捕获对时间戳以前数据的删除和更新操作,在数据准确性上受到了一定的限制。

3) 全表删除插入方式

全表删除插入方式是指每次 ETL 操作均删除目标表数据,由 ETL 全新加载数据。

优点:ETL 加载规则简单,速度快。

缺点:对于数据的管理不方便,当业务系统产生删除数据操作时,综合数据库不会记录到所删除的历史数据,不可以实现数据的递增加载;同时对于目标表所建立的关联关系,需要重新进行创建。

4) 全表比对方式

全表比对方式是采用 MD5 校验码,ETL 工具事先为要抽取的表建立一个结构类似的 MD5 临时表,该临时表记录源表主键以及根据所有字段的数据计算出来的 MD5 校验码,每次进行数据抽取时,对源表和 MD5 临时表进行 MD5 校验码的比对,如有不同,进行更新操作,如目标表中没有该主键值,表示还没有该记录,即进行插入操作。

优点:对已有系统表结构不产生影响,不需要修改业务操作程序,所有抽取规则由 ETL 完成,管理维护统一,可以实现数据的递增加载,没有风险。

缺点:ETL 比对较复杂,设计较复杂,速度较慢。与触发器和时间戳方式中的主动通知不同,全表比对方式是被动地进行全表数据的比对,性能较差。当表中没有主键或唯

一列且含有重复记录时,全表比对方式的准确性较差。

5) 日志表方式

日志表方式是指在业务系统中添加系统日志表,当业务数据发生变化时,更新维护日志表内容,当进行 ETL 加载时,通过读日志表数据决定加载哪些数据及如何加载。

优点:不需要修改业务系统表结构,源数据抽取清楚,速度较快,可以实现数据的递增加载。

缺点:日志表维护需要由业务系统完成,需要对业务系统的业务操作程序进行修改,记录日志信息。日志表维护较为麻烦,对原有系统有较大影响,工作量较大,改动较大,有一定风险。

2. Web 数据抽取

随着信息技术的不断发展,Web 上的信息内容和数据呈现出爆炸式的增长,从而使 Web 逐渐成为一个巨大、丰富、分布广泛的数据源。因此,有效地在 Web 上实现数据的抽取技术为进一步分析和挖掘提供了数据支持,具有十分重要的应用价值和现实意义。通过 Web 数据集成可以实现对 Web 数据的有效整合,为大数据分析提供信息源支持。Web 数据抽取技术随着互联网技术的发展、网页信息的扩充而产生,从手工到半自动,再到全自动的技术完善,Web 数据抽取技术成为大数据分析的主要技术,并在此基础上形成 Web 数据集成系统。Web 数据集成系统中的数据,不仅可以为各类大数据分析提供信息支持,还可以为 Web 数据集成系统自身集成提供帮助。

Web 数据抽取技术的作用和意义主要体现在:首先,Web 数据抽取是实现 Web 数据集成的基础和保证,Web 数据抽取可以完成对 Web 页面中广泛存在的结构化数据和半结构化数据的抽取工作,为 Web 数据集成奠定数据基础;其次,Web 数据抽取可以实现对 Web 数据的理解,Web 网页中的数据大部分是半结构化数据,通过 Web 数据抽取技术的实现,可以对抽取到的 Web 数据元素进行语言标注;最后,Web 数据抽取为 Web 数据集成中的其他环节提供数据服务,Web 数据抽取可以利用已抽取的 Web 数据对象间的联系,发现 Web 实体间的潜在联系。例如,在 Web 数据集成系统中,利用 Web 实体间的联系,可以形成一个基于这些联系的实施知识库,为进一步实施 Web 数据集成的数据整合、数据分析等服务提供数据支持。

一般来讲,目前 Web 数据抽取的实现有以下几步。

- (1) 确立采集目标,即由用户选择目标网站。
- (2) 提取特征信息,即根据目标网站的网页格式,提取出采集目标数据的共性。
- (3) 网络信息获取,即利用工具自动地把页面数据存到数据库。

值得注意的是,由于 Web 上的信息大多以 HTML 文档的形式出现,且 HTML 文档主要是用于浏览,而不是用于数据操作和应用的。因此,Web 信息抽取在传统的信息提取研究的基础上,将重点放在如何将分布在 Internet 上的半结构化 HTML 页面中的某些特定信息抽取出来,转化为结构化的形式保存在数据库中供用户查询、分析使用。

所谓半结构化数据,其实是结构化数据的一种形式,它并不符合关系型数据库或其他

数据表的形式关联起来的数据模型结构,但包含相关标记,用来分隔语义元素以及对记录 and 字段进行分层。因此,它也称为自描述的结构。这种灵活性可能使查询处理更加困难,但它也具有显著的优势。

图 3-2 所示为 JSON 在 Web 中的存储形式。

```
{
  "status": "OK",
  "result": {
    "location": {
      "lng": 116.512885,
      "lat": 39.847469
    },
    "precise": 1,
    "confidence": 80,
    "level": "\u9910\u996e"
  }
}
```

图 3-2 JSON 在 Web 中的存储形式

3.2 Web 数据抽取的实现

3.2.1 Kettle 数据抽取原理

Web 数据抽取可以依靠各种编程或开源软件实现。例如,可以通过 Kettle 抽取 Web 数据。通过 Kettle 获取的网页数据以结构化数据和半结构化数据为主,如人们熟悉的 XML 格式、JSON 格式等。本节主要讲解使用 Kettle 实现 Web 页面中的数据抽取。

在 Kettle 中要实现 Web 数据抽取,可以在“查询”列表中利用 HTTP client、HTTP post、REST client 以及 Web 服务查询等多个步骤完成,如图 3-3 所示。

1. HTTP client

Kettle 抽取网页数据的原理主要在于 Kettle 使用了 Apache VFS 系统,VFS 系统可以像处理文本文件一样处理 HTTP 文件,所以用户可以在 Kettle 中使用文本文件输入等步骤并直接将统一资源定位符(Uniform Resource Locator,URL)作为文件名使用。

HTTP client 是 Apache Jakarta Common 下的子项目,用来提供高效的、最新的、功能丰富的、支持 HTTP 协议的客户端编程工具包,并且支持 HTTP 最新的版本和建议。HTTP client 相比传统 jdk 自带的 URLConnection,增强了易用性和灵活性,它不仅使客户端发送 HTTP 请求变得容易,而且也方便开发人员测试接口(基于 HTTP),既提高了开发的效率,也方便提高代码的健壮性。现在,HTTP client 已经应用在很多项目中,如 Apache Jakarta 上很著名的两个开源项目 Cactus 和

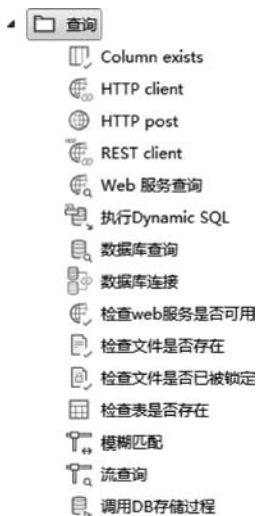


图 3-3 查询组件

HTML Unit,都使用了 HTTP client。

HTTP client 有以下几个主要特点。

- (1) 基于标准、纯净的 Java 语言,实现了 HTTP 1.0 和 HTTP 1.1。
- (2) 以可扩展的面向对象的结构实现了 HTTP 全部的方法(GET、POST、PUT、DELETE、HEAD、OPTIONS 以及 TRACE)。
- (3) 支持 HTTPS 协议。
- (4) 通过 HTTP 代理建立透明的连接。
- (5) 便携可靠的套接字工厂使它更容易使用第三方解决方案。
- (6) 直接获取服务器发送的 Response Code 和 Headers。
- (7) 在 HTTP 1.0 和 HTTP 1.1 中利用 Keep Alive 保持持久连接。
- (8) 设置连接超时的能力。

2. Web 服务查询

Web 服务查询也称为 Web Service,它是一个平台独立的、低耦合的、自包含的、基于可编程的 Web 应用程序,同时也是一种跨编程语言和跨操作系统平台的远程调用技术。Web Service 以 HTTP 为基础,通过 XML 进行客户端和服务端通信。通过 Web Service 可以使运行在不同机器上的不同应用程序不借助附加的、专门的第三方软件或硬件,就可相互交换数据。

其实可以从多个角度理解 Web Service。从表面上看,Web Service 就是一个应用程序向外界暴露出一个能通过 Web 进行调用的 API(属于一种操作系统或程序接口),也就是说,能用编程的方法通过 Web 调用这个应用程序。我们把调用这个 Web Service 的应用程序叫作客户端,把提供这个 Web Service 的应用程序叫作服务端。从深层次看,Web Service 是建立可互操作的分布式应用程序的新平台,是一套标准,它定义了应用程序如何在 Web 上实现互操作性。

Web Service 关键技术与协议如下。

(1) 可扩展标记语言(eXtensible Markup Language,XML): XML 是在 Web 上传输结构化数据的伟大方式,Web Service 要以一种可靠的、自动的方式操作数据,HTML 满足不了其要求,而 XML 可以使 Web Service 以十分方便的方式处理数据。

(2) 简易对象访问协议(Simple Object Access Protocol,SOAP): 用来描述传递信息的格式,是一种简单的基于 XML 的协议,它被设计成在 Web 上交换结构化和固化的信息。SOAP 采用了已经广泛使用的两个协议: HTTP 和 XML,其中 HTTP 用于实现 SOAP 的远程过程调用(Remote Procedure Call,RPC)风格的传输,而 XML 是它的编码模式。

(3) Web 服务描述语言(Web Service Description Language,WSDL): 用来描述 Web Service 以及如何访问 Web Service。同时,WSDL 文档是一个遵循 WSDL-XML 模式的 XML 文档,WSDL 文档将 Web 服务定义为服务访问点或端口的集合。在 WSDL

中,由于服务访问点和消息的抽象定义已从具体的服务部署或数据格式绑定中分离出来,因此可以对抽象定义进行再次使用。

(4) UDDI (Universal Description Discovery and Integration): 一种用于描述、发现、集成 Web Service 的技术,它是 Web Service 协议栈的一个重要部分。通过 UDDI,企业可以根据自己的需要动态查找并使用 Web 服务,也可以将自己的 Web 服务动态地发布到 UDDI 注册中心,供其他用户使用。

在具体实现中,SOAP 用来描述传递信息的格式;WSDL 用来描述如何访问具体的接口;UDDI 用来管理、分发、查询 Web Service。

Web Service 原理如图 3-4 所示。

Web Service 常见的开发实例如下。

- (1) 利用 Web Service 实现数据添加。
- (2) 利用 Web Service 实现数据删除。
- (3) 利用 Web Service 实现给手机发短信。

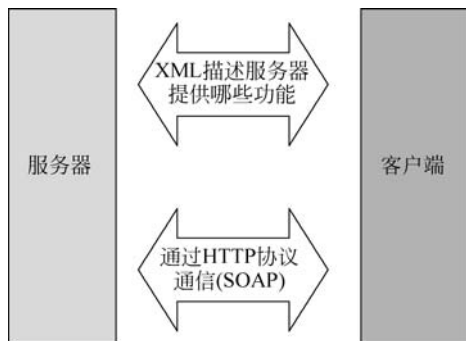


图 3-4 Web Service 原理

3.2.2 Kettle 数据抽取实现

1. HTTP client 数据抽取

【例 3-1】 从生成记录中抽取数据。

(1) 启动 Kettle 后,新建转换,在“输入”列表中选择“生成记录”步骤,在“查询”列表中选择 HTTP client 步骤,拖动至右侧工作区并建立节点连接,如图 3-5 所示。

(2) 双击“生成记录”图标,在“限制”输入框中输入 1;设置字段名称为 string,类型为 String,值为 `http://services.odata.org/V3/Northwind/Northwind.svc/Products/`,如图 3-6 所示。

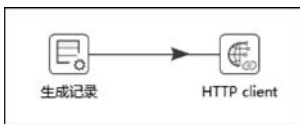


图 3-5 HTTP client 数据抽取工作流程

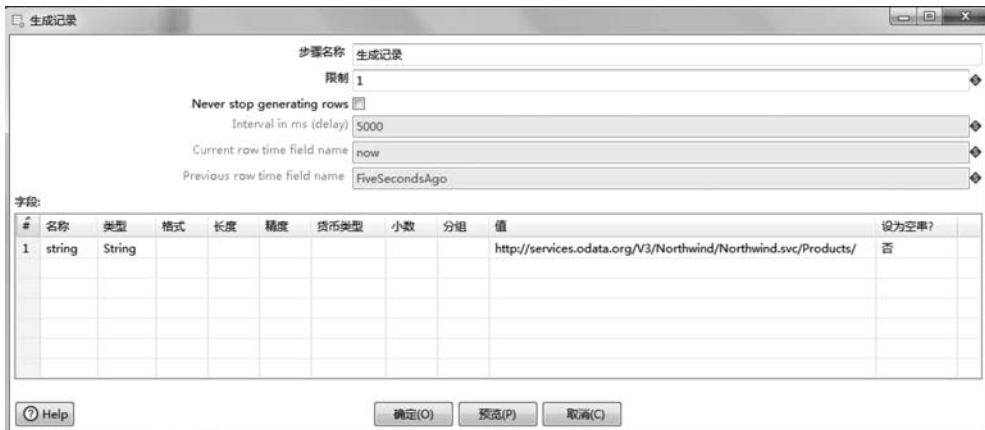


图 3-6 设置生成记录(1)



视频讲解

该网址中存储的数据格式为 XML 类型,网站部分数据如图 3-7 所示。

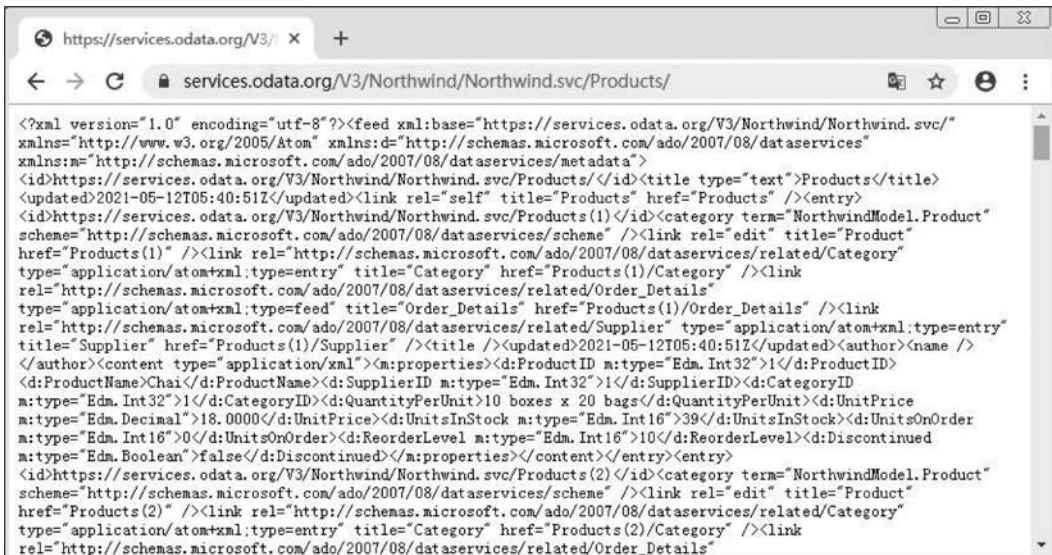


图 3-7 网站部分数据

(3) 双击 HTTP client 图标,在 General 选项卡中勾选“从字段中获取 URL?”,并输入 URL 字段名为 string,结果字段名为 result,并在 Fields 选项卡中获取字段,如图 3-8 和图 3-9 所示。

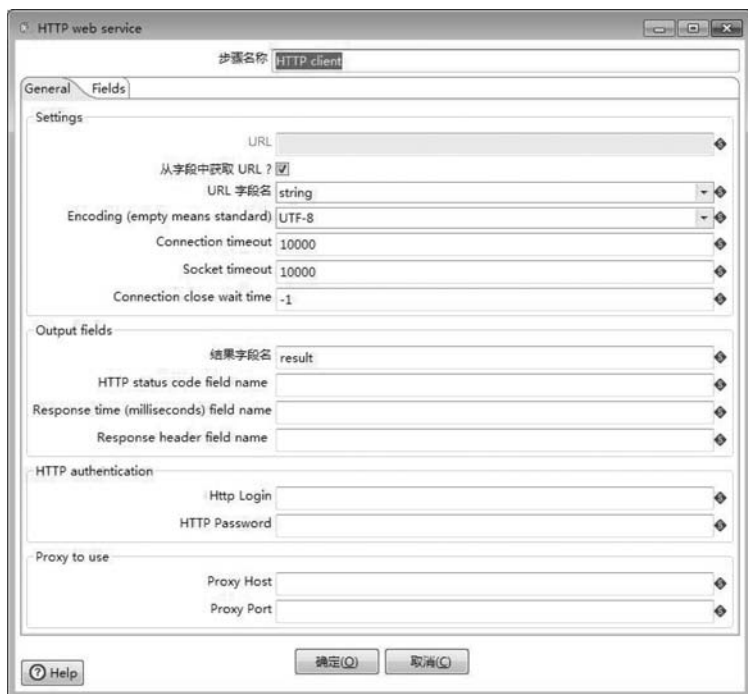


图 3-8 设置 General 选项卡

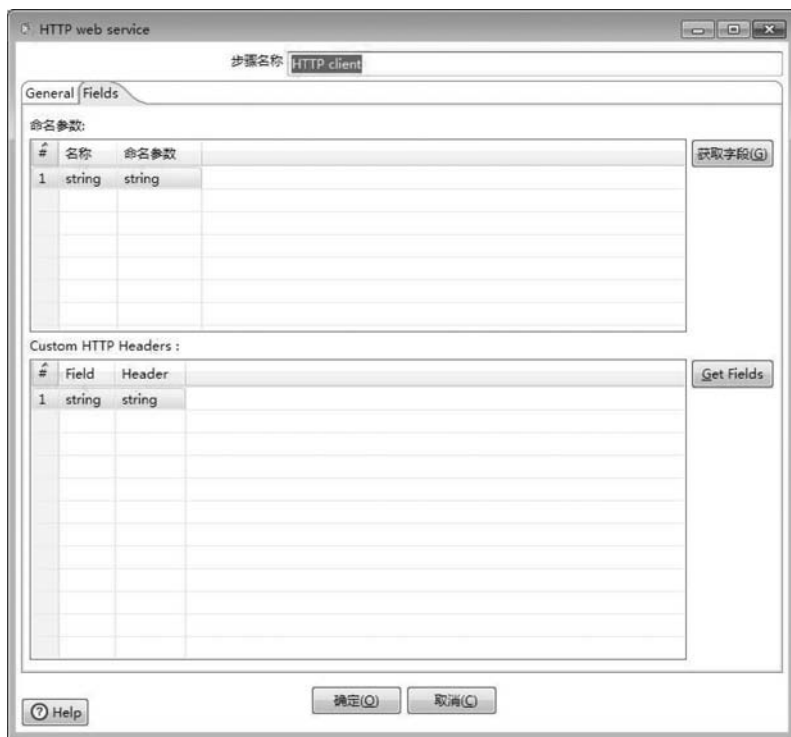


图 3-9 获取字段

(4) 保存该文件,运行转换,执行结果如图 3-10 所示。

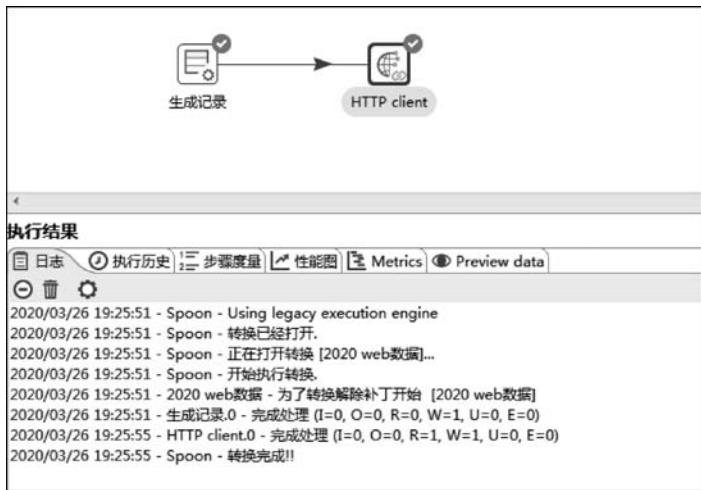


图 3-10 运行转换

(5) 可以清楚地看到,已经完成了数据的抽取,如图 3-11 所示。

【例 3-2】 抽取 XML 数据并显示。

(1) 启动 Kettle 后,新建转换,在“输入”列表中选择“生成记录”步骤,在“查询”列表



视频讲解

#	string	result
1	http://services.odata.org/V3/Northwind/Northwind.svc/Pro...	<?xml version="1.0" encoding="utf-8"?><feed xmlns="https://services.odata.org/V3/Northwind/Northwind.svc/"

图 3-11 数据抽取结果

中选择 HTTP client 步骤,在“输入”列表中选择 Get data from XML 步骤,在“转换”列表中选择“字段选择”步骤,将其一一拖动到右侧工作区中,并建立彼此之间的节点连接关系,如图 3-12 所示。



图 3-12 Kettle 抽取 XML 数据工作流程

(2) 双击“生成记录”图标,设置字段名称为 url,类型为 String,值为 http://services.odata.org/V3/Northwind/Northwind.svc/Products/,如图 3-13 所示。

步骤名称: 生成记录

限制: 10

Never stop generating rows: ☐

Interval in ms (delay): 5000

Current row time field name: now

Previous row time field name: FiveSecondsAgo

字段:

#	名称	类型	格式	长度	精度	货币类型	小数	分组	值
1	url	String							https://services.odata.org/V3/Northwind/Northwind.svc/Products/

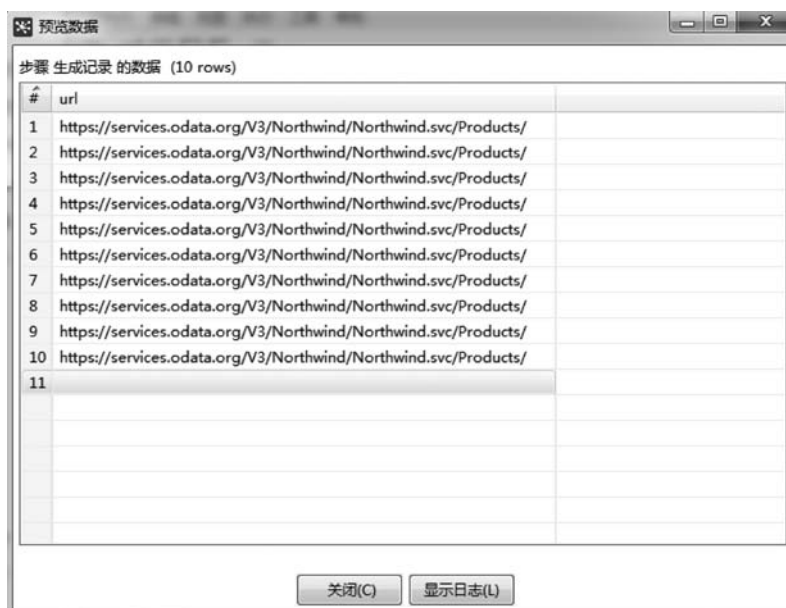
Buttons: ? Help, 确定(O), 预览(P), 取消(C)

图 3-13 设置生成记录(2)

(3) 单击“预览”按钮,可查看生成记录的数据,如图 3-14 所示。

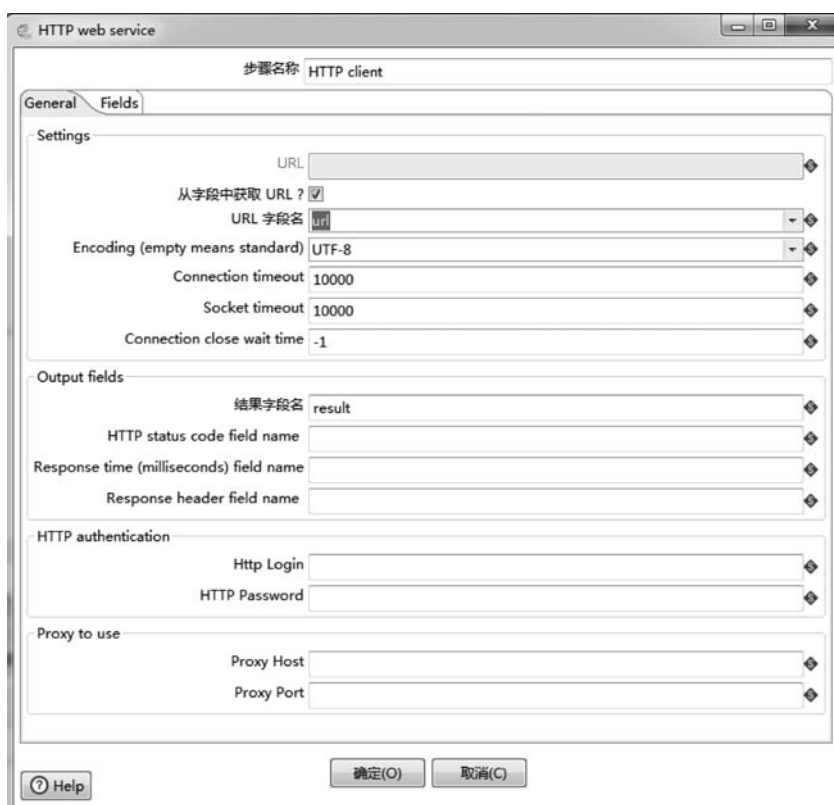
(4) 双击 HTTP client 图标,勾选“从字段中获取 URL?”,在“URL 字段名”下拉列表中选择 url,在“结果字段名”输入框中输入 result,如图 3-15 所示。

(5) 双击 Get data from XML 图标,在“文件”选项卡中勾选“XML 源定义在一个字段里?”,在“XML 源字段名”下拉列表中选择 result,如图 3-16 所示。



#	url
1	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
2	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
3	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
4	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
5	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
6	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
7	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
8	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
9	https://services.odata.org/V3/Northwind/Northwind.svc/Products/
10	https://services.odata.org/V3/Northwind/Northwind.svc/Products/

图 3-14 预览生成记录数据



HTTP web service

步骤名称 HTTP client

General Fields

Settings

URL

从字段中获取 URL ? ☒

URL 字段名

Encoding (empty means standard) UTF-8

Connection timeout 10000

Socket timeout 10000

Connection close wait time -1

Output fields

结果字段名 result

HTTP status code field name

Response time (milliseconds) field name

Response header field name

HTTP authentication

Http Login

HTTP Password

Proxy to use

Proxy Host

Proxy Port

Help 确定(O) 取消(C)

图 3-15 设置 HTTP client



图 3-16 设置 XML 文件输入

(6) 在“内容”选项卡中的“循环读取路径”中输入 `/feed/entry/content/m:properties`。该路径是 XML 语法中的 Xpath 查询,用于读取网页数据中的节点内容,如图 3-17 所示。



图 3-17 设置循环读取路径

(7) 在“字段”选项卡中输入以下字段内容,如图 3-18 所示。

(8) 双击“字段选择”图标,在“选择和修改”选项卡中输入字段内容,如图 3-19 所示。



图 3-18 输入字段内容

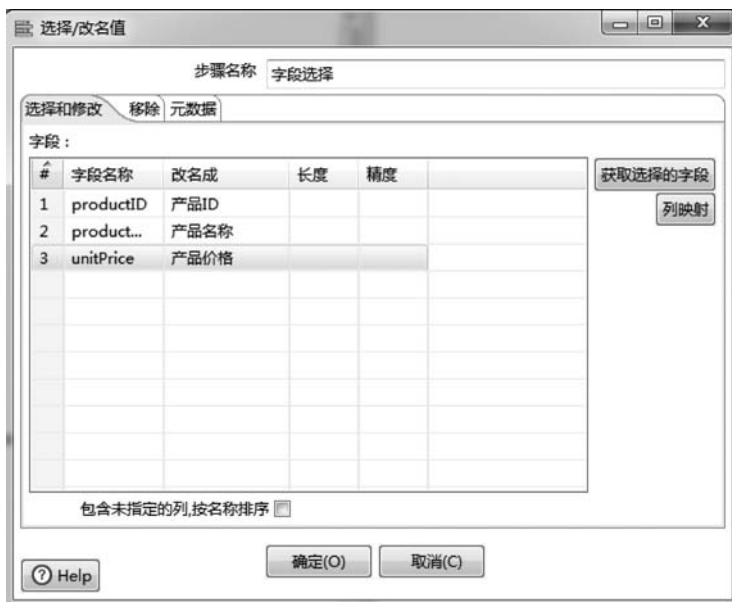


图 3-19 设置“选择和修改”选项卡

(9) 保存该文件,运行转换,可以在执行结果区域的“步骤度量”选项卡中查看该程序的执行状况,如图 3-20 所示。

(10) 在执行结果区域的 Preview data 选项卡中查看该程序抽取网页的数据内容,这里选择前 20 条数据显示。在结果中显示了产品 ID、产品名称以及产品价格,如图 3-21 所示。

【课堂练习】 从生成记录中抽取数据并用文本文件输出显示。

制作如图 3-22 所示的流程,在 HTTP client 后用文本文件输出结果。

其中 URL 为 <http://sports.sina.com.cn/global/>。

生成的文本文件如图 3-23 所示。

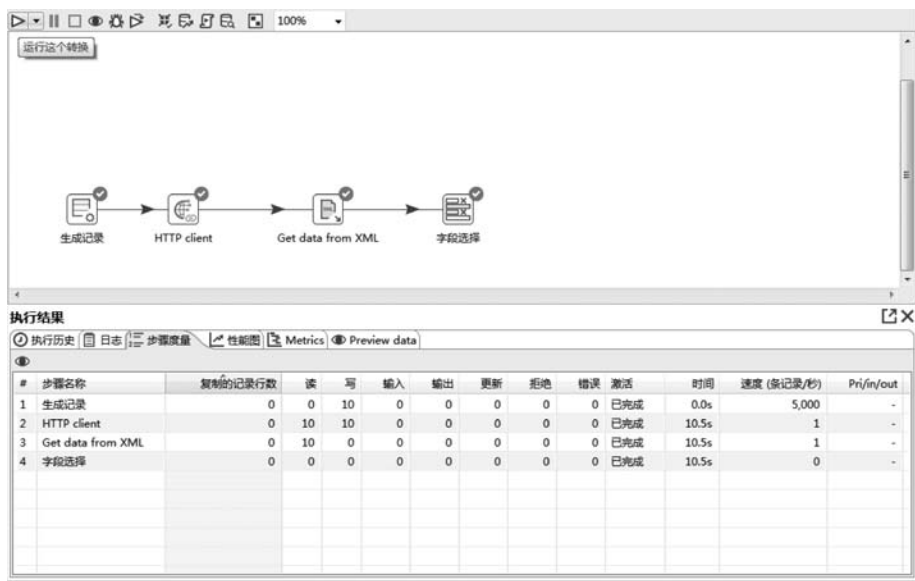


图 3-20 查看程序的执行状况

执行结果

日志 执行历史 步骤度量 性能图 Metrics Preview data

\$(TransPreview.FirstRows.Label) \$(TransPreview.LastRows.Label) \$(TransPreview.Off.Label)

#	产品编号	产品名称	产品单价
1	1	Chai	18.0
2	2	Chang	19.0
3	3	Aniseed Syrup	10.0
4	4	Chef Anton's Cajun Seasoning	22.0
5	5	Chef Anton's Gumbo Mix	21.35
6	6	Grandma's Boysenberry Spread	25.0
7	7	Uncle Bob's Organic Dried Pears	30.0
8	8	Northwoods Cranberry Sauce	40.0
9	9	Mishi Kobe Niku	97.0
10	10	Ikura	31.0
11	11	Queso Cabrales	21.0
12	12	Queso Manchego La Pastora	38.0
13	13	Konbu	6.0
14	14	Tofu	23.25
15	15	Genen Shouyu	15.5
16	16	Pavlova	17.45
17	17	Alice Mutton	39.0
18	18	Carnarvon Tigers	62.5
19	19	Teatime Chocolate Biscuits	9.2
20	20	Sir Rodney's Marmalade	81.0

图 3-21 显示抽取的网页数据

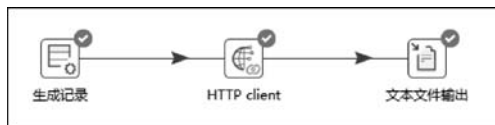


图 3-22 Kettle 流程

[illegible]

图 3-23 生成的文本文件内容

2. Web Service 数据抽取

Kettle 可通过两种方式获取 Web Service 结果,一种是 Web 服务查询,另一种是利用 HTTP POST。下面介绍使用 Web 服务查询的方式获取 Web Service 结果。

【例 3-3】 使用 Web Service 抽取天气数据并显示。

(1) 启动 Kettle 后,新建转换,在“输入”列表中选择“生成记录”步骤,在“查询”列表中选择“Web 服务查询”步骤,在“输出”列表中选择“文本文件输出”步骤,将其一一拖动到右侧工作区中,并建立彼此之间的节点连接关系,如图 3-24 所示。

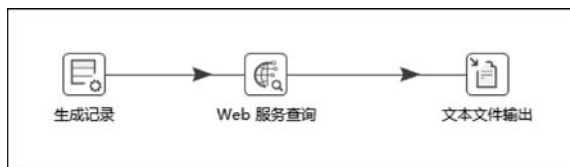


图 3-24 Web Service 数据抽取工作流程

(2) 双击“生成记录”图标,在“限制”输入框中输入 1,设置字段名称为 localcity,类型为 String,值为重庆,如图 3-25 所示。

生成记录

步骤名称

限制

Never stop generating rows ☐

Interval in ms (delay)

Current row time field name

Previous row time field name

字段:

#	名称	类型	模式	长度	精度	货币类型	小数	分组	值	设为空?	
1	localcity	String							重庆	否	

图 3-25 生成记录设置

(3) 双击“Web 服务查询”图标,在 Web Services 选项卡中的 URL 框中输入 <http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL>,在“操作”中选

择 `getWeatherbyCityName`, 注意“v2. x/3. 0 兼容模式”要取消勾选, 最后单击“加载”按钮, 如图 3-26 所示。

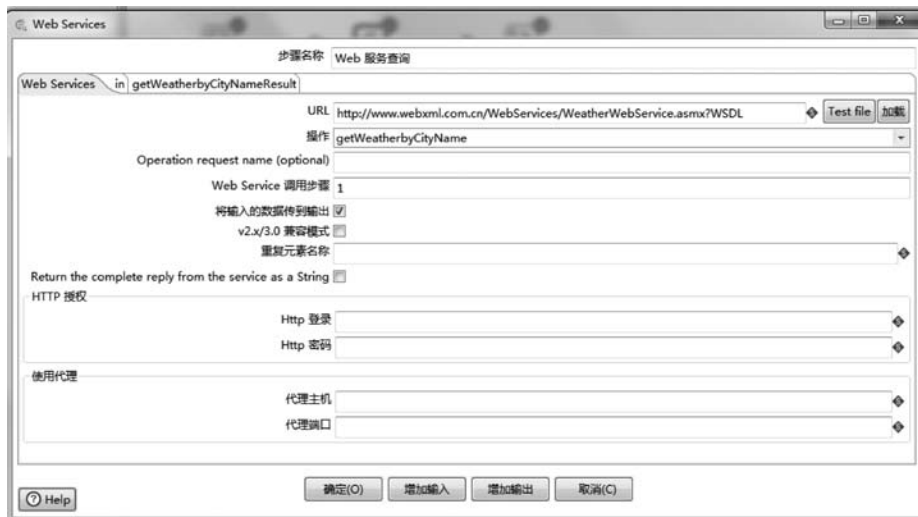


图 3-26 Web Services 设置

该网址为 `http://www.webxml.com.cn/WebServices/WeatherWebService.asmx`, 其页面内容如图 3-27 所示。

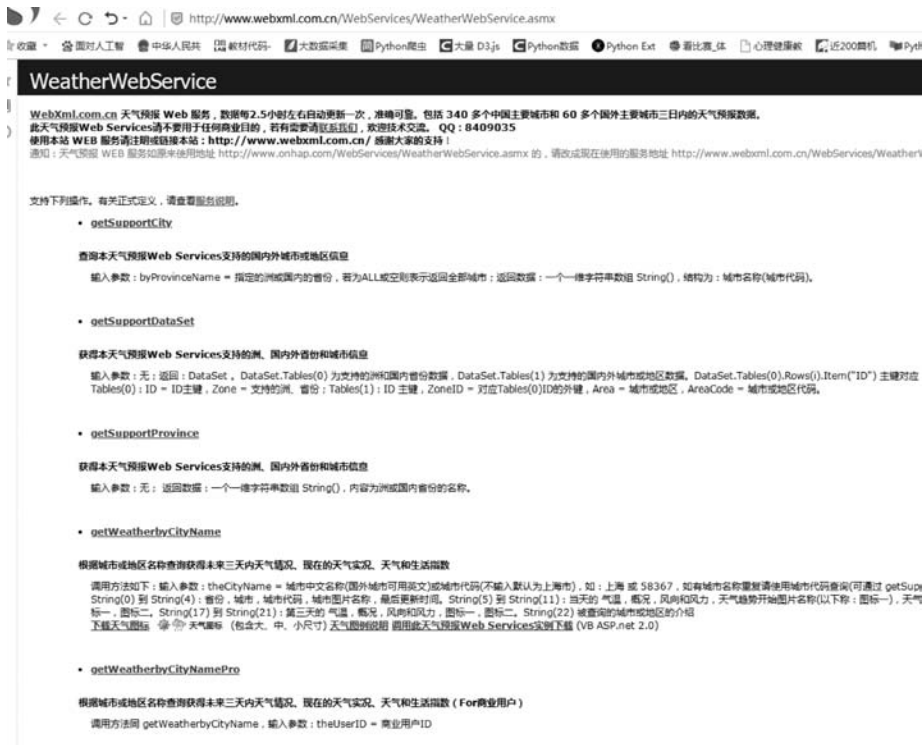


图 3-27 网站内容

在该网站中,我们调用了接口——getWeatherbyCityName,得到城市的天气情况,如图 3-28 所示。

• getWeatherbyCityName

根据城市或地区名称查询获得未来三天内天气情况。现在的天气实况、天气和生活指数

调用方法如下:输入参数:theCityName = 城市中文名称(国外城市可用英文)或城市代码(不输入默认为上海市),如:上海 或 58367,如有城市 String(0) 到 String(4): 省份,城市,城市代码,城市图片名称,最后更新时间。String(5) 到 String(11): 当天的 气温,概况,风向和风力, : 标一,图标二。String(17) 到 String(21): 第三天的 气温,概况,风向和风力,图标一,图标二。String(22) 被查询的城市或地区的介绍
下载天气图标 天气图标 (包含大、中、小尺寸) 天气图标说明 调用此天气预报Web Services实例下载 (VB ASP.net 2.0)

图 3-28 网站接口

由于调用了 Web 服务,因此在前面的 URL 框中网址最后要加上“? WSDL”。

(4) 在 in 选项卡中输入名称、WS 名称和 WS 类型分别为 localcity、theCityName 和 string,如图 3-29 所示。



图 3-29 in 选项卡设置

(5) 在 getWeatherbyCityNameResult 选项卡中输入名称、WS 名称和 WS 类型分别为 WeatherbyCityNameResult、getWeatherbyCityNameResult 和 ArrayOfstring,如图 3-30 所示。



图 3-30 getWeatherbyCityNameResult 选项卡设置

3.4 实训

1. 实训目的

通过本章实训了解 Web 抽取数据的特点,能进行简单的与 Web 抽取有关的操作。

2. 实训内容

1) 使用 Web Service 抽取汇率数据并显示

(1) 页面网址为 <http://webservices.gama-system.com/exchangerates.asmx>,内容如图 3-33 所示。该网站通过 Web Service 提供汇率和货币换算信息。

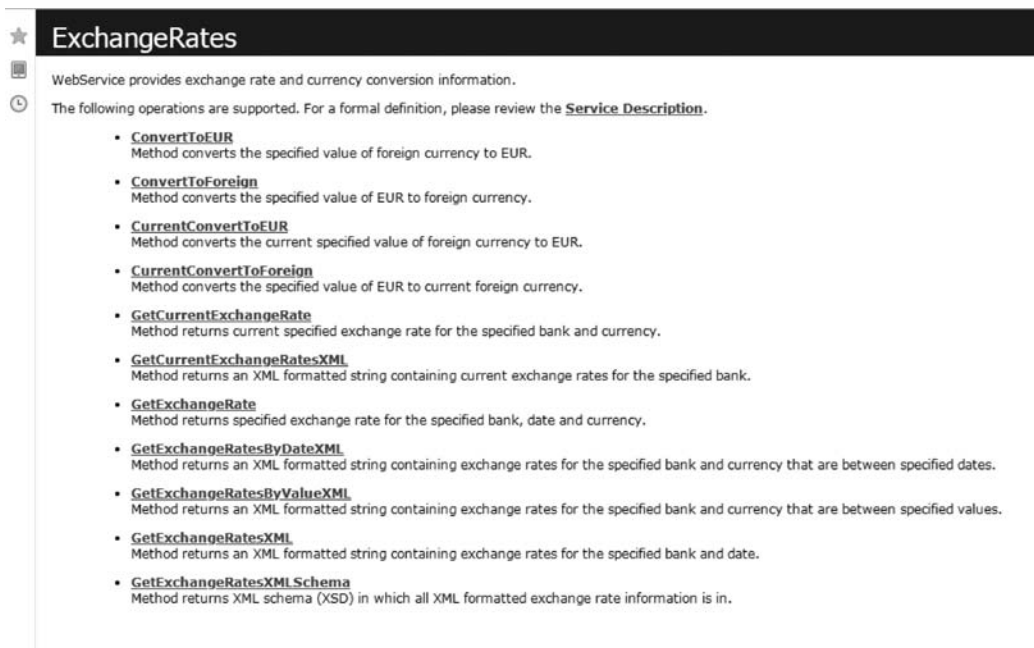


图 3-33 访问的网页页面

在该网站中我们调用了—个接口——GetCurrentExchangeRate,得到货币的汇率情况,如图 3-34 所示。

(2) 启动 Kettle 后,新建转换,在“输入”列表中选择“生成记录”步骤,在“查询”列表中选择“Web 服务查询”步骤,在“输出”列表中选择 XML output 步骤,并将其分别拖动到右侧工作区中建立节点连接,如图 3-35 所示。

(3) 双击“生成记录”图标,在“限制”输入框中输入 1,字段内容设置如图 3-36 所示。其中,GBP 表示英镑;USD 表示美元。

(4) 双击“Web 服务查询”图标,在 Web Services 选项卡中的 URL 框中输入 <http://webservices.gama-system.com/exchangerates.asmx?WSDL>,在“操作”下拉列表中选

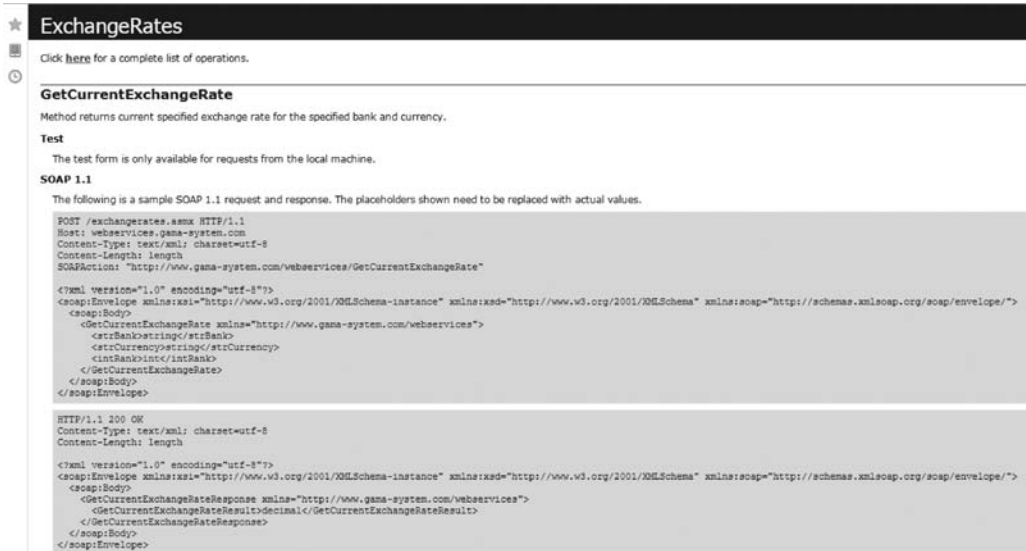


图 3-34 抽取的页面



图 3-35 抽取汇率数据工作流程

生成记录

步骤名称: 生成记录

限制: 1

Never stop generating rows: ☐

Interval in ms (delay):

Current row time field name:

Previous row time field name:

字段:

#	名称	类型	格式	长度	精度	货币类型	小数	分组	值	设为空串?
1	FromCurrency	String		8					GBP	否
2	ToCurrency	String		8					USD	否
3	Bank	String		8					SKB	否
4	Rank	Integer							1	否

Help 确定(O) 预览(P) 取消(C)

图 3-36 设置生成记录

择 GetCurrentExchangeRate, 注意在本例中要将“v2. x/3.0 兼容模式”勾选上, 最后单击“加载”按钮, 如图 3-37 所示。



图 3-37 Web Services 设置

(5) 在 in 选项卡中输入名称、WS 名称和 WS 类型, 内容如图 3-38 所示。



图 3-38 in 选项卡设置

(6) 在 GetCurrentExchangeRateResult 选项卡中输入名称、WS 名称和 WS 类型分别为 GetCurrentExchangeRateResult、GetCurrentExchangeRateResult 和 decimal, 如图 3-39 所示。

(7) 双击 XML output 图标, 在“文件”选项卡中设置要保存的文件名称为 2020-2. xml,

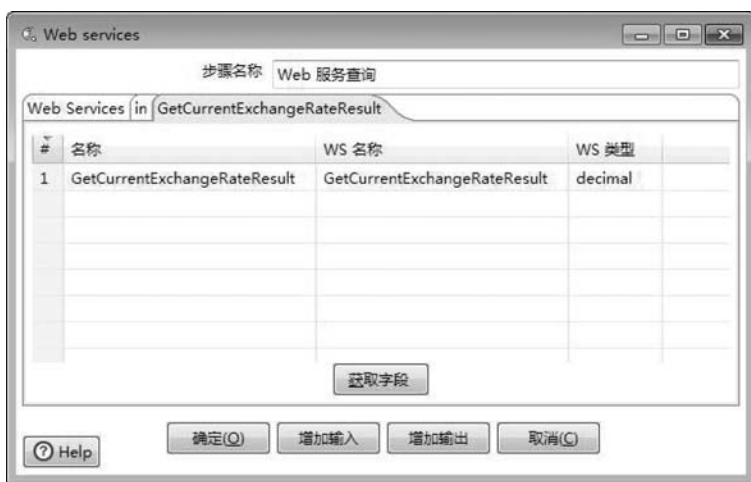


图 3-39 GetCurrentExchangeRateResult 选项卡设置

在“字段”选项卡中单击“获取字段”按钮,如图 3-40 和图 3-41 所示。



图 3-40 保存输出文件

(8) 保存该文件,运行转换,执行该程序,如图 3-42 所示。

(9) 查看执行结果,如图 3-43 所示。

(10) 在生成的 XML 文件中查看结果,如图 3-44 所示。

2) 抽取 Web 中的图片

本例使用 Kettle 抽取 Web 中的图片,URL 地址为 <https://pics3.baidu.com/feed/alcec08fa513d26977ec9e1f4201608f34216d873.jpeg?token=ccfab3dle2305a91be3a1100b46cbfcl>。图片效果如图 3-45 所示。

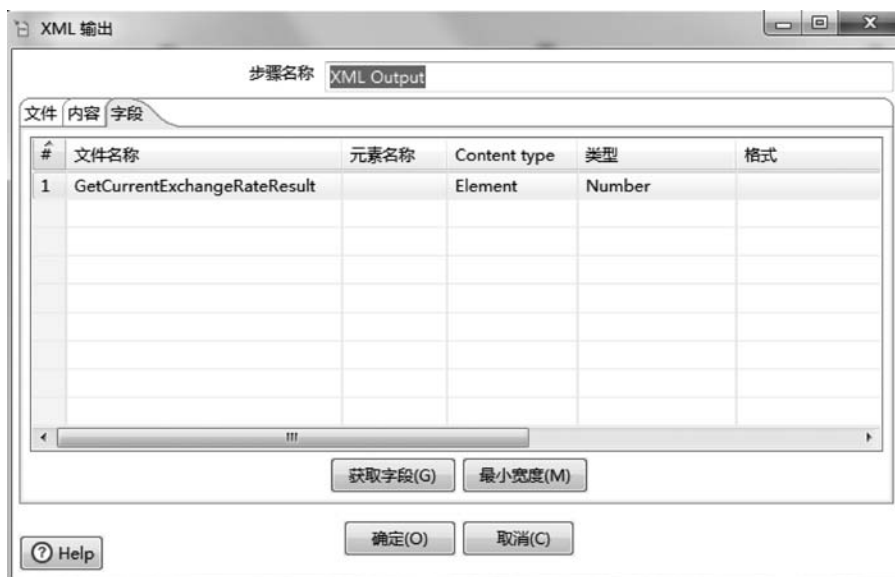


图 3-41 获取字段

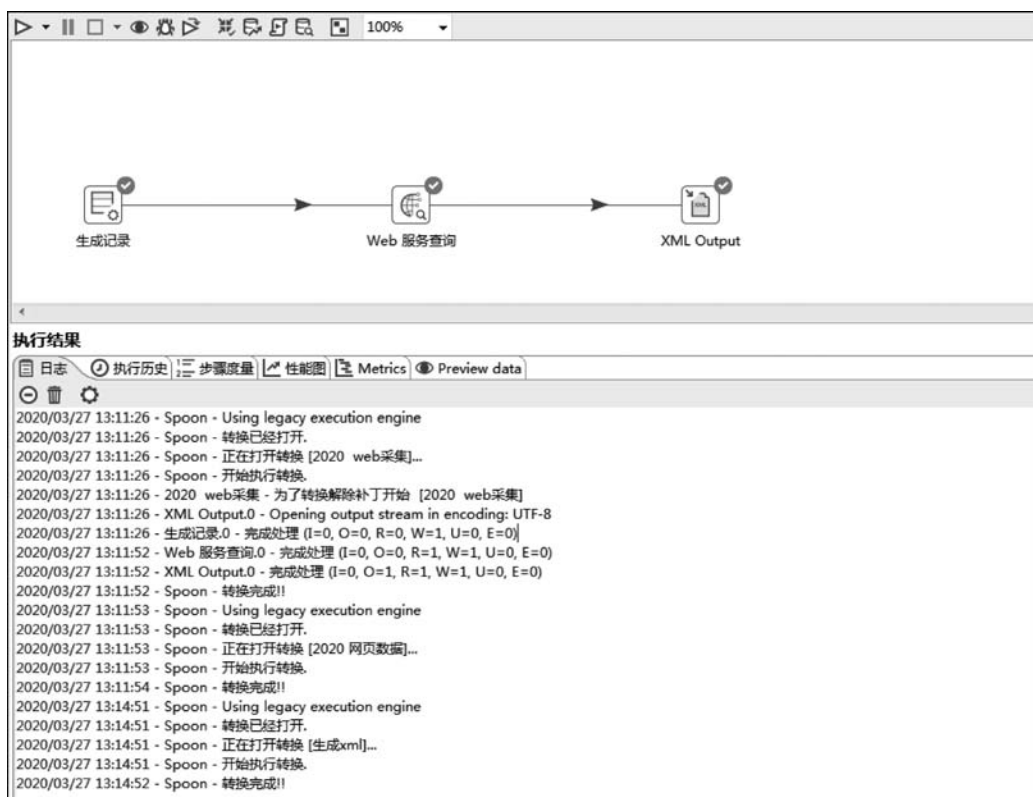


图 3-42 运行转换

执行结果

日志 执行历史 步骤度量 性能图 Metrics Preview data

\$(TransPreview.FirstRows.Label) \$(TransPreview.LastRows.Label) \$(TransPreview.Off.Label)

#	GetCurrentExchangeRateResult
1	-3.0

图 3-43 查看执行结果

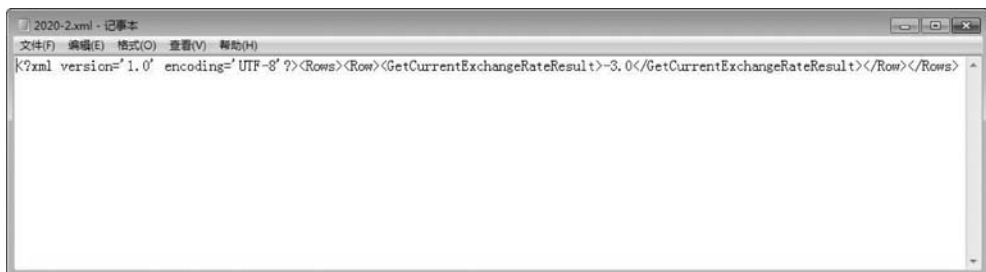


图 3-44 查看生成文件



图 3-45 Web 中的图片

(1) 成功运行 Kettle 后,在菜单栏选择“文件”|“转换”命令,在“输入”选项中选择“自定义常量数据”,在“查询”选项中选择 HTTP client,并将其拖动到右侧工作区中建立节点连接,如图 3-46 所示。

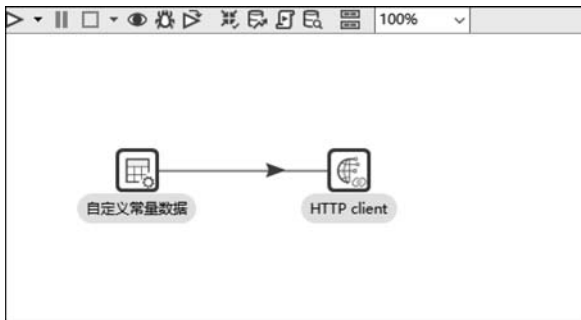


图 3-46 建立连接

(2) 双击“自定义常量数据”图标,在“元数据”选项中的“名称”中输入“url”,在“数据”选项中的 url 中输入图片的网址,如图 3-47、图 3-48 所示。



图 3-47 设置元数据



图 3-48 设置数据

从图 3-47、图 3-48 可以看出,自定义常量数据与生成记录不同,生成记录只有一个选项,字段的定义和值在同一个页签。而自定义常量有两个选项,分别是元数据和数据,其中,元数据用于设置字段的信息,若字段定义为 date 类型,则格式必须选择;而数据用于为定义的字段填充数据,用法类似于数据库中的建表并插入数据,自定义常量数据用法相比于生成记录要灵活很多。值得注意的是:在元数据中定义的字段,会在数据中自动生成供数据填充。

(3) 双击 HTTP client 图标,选中“从字段中获取 URL”复选框,如图 3-49 所示。

(4) 保存该文件,选择“运行这个转换”选项,执行该程序,并查看运行结果,如图 3-50 所示。

从图 3-50 可以看出,该抽取过程已经顺利完成,只是因为 Kettle 中不能直接显示抽取的 Web 图片,因此在执行结果的 result 中显示的是乱码。一般在企业的实际操作中



图 3-49 设置 HTTP client

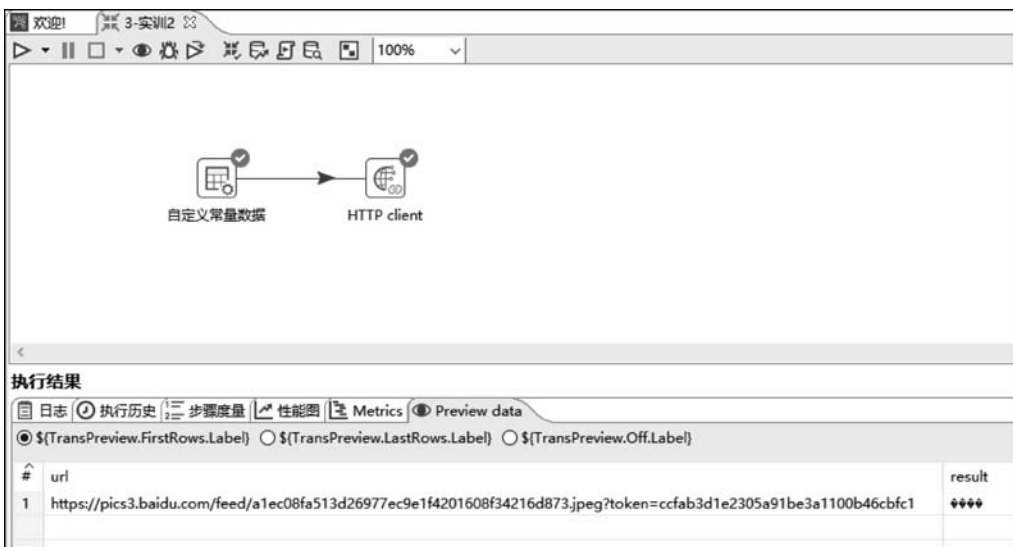


图 3-50 运行程序

往往是将要抽取的图片 URL 地址保存在图片服务器中,然后在图片服务器中打开查看。

习题 3

- (1) 什么是数据抽取?
- (2) 什么是 Web 数据抽取?
- (3) 如何使用 Kettle 进行 Web 数据抽取?