TUSCANY

REST 风格的数据查询服务

陈栋

2013-5-28

NJU

1.引言

本文介绍了如何在 tuscany2.0 版本中实现对 mysql 数据库的访问，并将对数据库的操作暴露成 Web service。

tuscany 是目前开源界最成熟的 SCA 框架之一，它能够非常方便地将 SCA 和 WEB2.0 结合，这一点官网也提供了一些例子，例如购物系统等。官网上并没有 Tuscany2.0 的例子，但是源码的 samples 目录下有很多示例可以使用。

Tuscany 是基于组件(Component)的实现方式，一个 SCA 组件可以由多种语言或者技术平台实现，如：POJO，EJB，Spring Bean，bpel 流程，各种脚本语言等等。此外，SCA 服务（Service）和引用（Reference）的绑定方式很多样化，即一个 SCA 服务可以暴露为 Web Service，Java RMI 服务，http 资源，jms 消息等等，一个 SCA 引用也可以通过 Web Service，RMI 调用，http 调用，jms 调用等方式调用远端服务。

例如在$(tuscany-dir)\samples\getting-started 中提供了许多种绑定方式，如 jaxrs[1]，jsonp，jsonrpc，soap webservice 等。

考虑到对数据库访问的服务要提供给另一个 web 页面使用，为了简单起见，采用 JAX-RS，并通过标注允许外部通过 get 请求访问 web 服务。

1.1. REST

表征状态转移（英文：Representational State Transfer，简称 REST）是 Roy Fielding 博士在 2000 年他的博士论文中提出来的一种软件架构风格。

目前在三种主流的 Web 服务实现方案中，因为 REST 模式的 Web 服务与复杂的 SOAP 和 XML-RPC 对比来讲明显的更加简洁，越来越多的 web 服务开始采用 REST 风格设计和实现。例如，Amazon.com 提供接近 REST 风格的 Web 服务进行图书查找；雅虎提供的 Web 服务也是 REST 风格的。

1.2. JAX-RS

JAX-RS 提供了一些标注将一个资源类，一个 POJOJava 类，封装为 Web 资源。标注包括：

@Path，标注资源类或方法的相对路径

@GET，@PUT，@POST，@DELETE，标注方法是用的 HTTP 请求的类型

@Produces，标注返回的 MIME 媒体类型

@Consumes，标注可接受请求的 MIME 媒体类型

@PathParam，@QueryParam，@HeaderParam，@CookieParam，@MatrixParam，@FormParam,分别标注方法的参数来自于 HTTP 请求的不同位置，例如@PathParam 来自于 URL 的路径，@QueryParam 来自于 URL 的查询参数，@HeaderParam 来自于 HTTP 请求的头信息，@CookieParam 来自于 HTTP 请求的 Cookie。

[1] Jaxrs: Java API for RESTful Web Services 是一个 Java 编程语言的应用程序接口,支持按照 表象化状态转变(REST)架构风格创建 Web 服务 Web 服务. JAX-RS 使用了 Java SE 5 引入的 Java 标注来简化 Web 服务客户端和服务端的开发和部署。

2.数据库设计

我们需要将水利数据存储到数据库中，这里采用 mysql 数据库。前端要求提供几个大城市的降水量数据和南京市几个重要水文站的降水量和水位数据，所以数据库表按照如下方式设计：

```
DROP TABLE IF EXISTS `CITY`;
CREATE TABLE `CITY`(
	`ID_` int(10) unsigned NOT NULL auto_increment,
	`NAME_` varchar(80) character set utf8 NOT NULL default '',
	`ALIAS_` varchar(80) NOT NULL default '',
	PRIMARY KEY	(`ID_`)
);

DROP TABLE IF EXISTS `RAINFALL`;
CREATE TABLE `RAINFALL` (
	`id` int(10) unsigned NOT NULL,
	`TIME_` datetime default '0000-00-00 00:00:00',
	`AMOUNT_` float,
	FOREIGN KEY (`id`) REFERENCES `CITY` (`ID_`)
);

DROP TABLE IF EXISTS `STATIONINFO`;
CREATE TABLE `STATIONINFO`(
	`ID_` int(10) unsigned NOT NULL auto_increment,
	`NAME_` varchar(80) character set utf8 NOT NULL default '',
	`ALIAS_` varchar(80) NOT NULL default '',
	`TYPE_` int unsigned,
	PRIMARY KEY	(`ID_`)
);

DROP TABLE IF EXISTS `PONDAGE`;
CREATE TABLE `PONDAGE` (
	`id` int(10) unsigned NOT NULL,
	`TIME_` datetime default '0000-00-00 00:00:00',
	`WATERLEVEL_` float,
	FOREIGN KEY (`id`) REFERENCES `STATIONINFO` (`ID_`)
);

DROP TABLE IF EXISTS `STATIONRAINFALL`;
CREATE TABLE `STATIONRAINFALL` (
	`id` int(10) unsigned NOT NULL,
	`TIME_` datetime default '0000-00-00 00:00:00',
	`AMOUNT_` float,
```

```
    FOREIGN KEY (`id`) REFERENCES `STATIONINFO` (`ID_`)
);
```

2.1. City 表

几个重要城市的信息，ALIAS_字段是城市的拼音。前端通过 get 请求查询时会发送城市的拼音。

ID_	NAME_	ALIAS_
1	台州	taizhou
2	舟山	zhoushan
3	宁波	ningbo
4	温州	wenzhou
5	绍兴	shaoxing
6	杭州	hangzhou
7	湖州	huzhou
8	无锡	wuxi
9	南京	nanjing

2.2. Rainfall 表

数据如下，其中 AMOUNT_字段是当天某个时刻的从 0 时的累积降雨量，id 是外键，来源于 city 表。

id	TIME_	AMOUNT_
1	2012-08-07 00:00:00	0
1	2012-08-07 01:00:00	0
1	2012-08-07 02:00:00	0
1	2012-08-07 03:00:00	0
1	2012-08-07 04:00:00	0
1	2012-08-07 05:00:00	0
1	2012-08-07 06:00:00	0
1	2012-08-07 07:00:00	0
1	2012-08-07 08:00:00	0
1	2012-08-07 09:00:00	0
1	2012-08-07 10:00:00	0
1	2012-08-07 11:00:00	3
1	2012-08-07 12:00:00	6
1	2012-08-07 13:00:00	9
1	2012-08-07 14:00:00	14
1	2012-08-07 15:00:00	20
1	2012-08-07 16:00:00	28
1	2012-08-07 17:00:00	36
1	2012-08-07 18:00:00	45
1	2012-08-07 19:00:00	54
1	2012-08-07 20:00:00	63
1	2012-08-07 21:00:00	73
1	2012-08-07 22:00:00	83
1	2012-08-07 23:00:00	94
1	2012-08-08 00:00:00	10
1	2012-08-08 01:00:00	19
1	2012-08-08 02:00:00	30
1	2012-08-08 03:00:00	40
1	2012-08-08 04:00:00	51
1	2012-08-08 05:00:00	61

2.3. Stationinfo 表

存储了南京市几个重要水文站的信息，TYPE_字段为 0 表示是河道站，为 1 表示是水库站。

ID_	NAME_	ALIAS_	TYPE_
1	晓桥	xiaoqiao	0
2	六合	liuhe	0
3	东山	dongshan	0
4	陈家桥	chenjiaqiao	0
5	高淳	gaochun	0
6	葛塘	getang	0
7	天生桥	tianshengqiao	0
8	茅东闸	maodongzha	0
9	方便	fangbian	1
10	大河桥	daheqiao	1
11	河王坝	hewangba	1
12	老鸦坝	laoyaba	1
13	姚家	yaojia	1
14	中山	zhongshan	1

2.4. pondage 表

Pondage 表存储了南京市水文站的水位数据，格式和 rainfall 表类似，只是这里的 id 作为外键，引用的是 stationinfo 表中的 ID_。

id	TIME_	WATERLEVEL_
1	2012-08-08 00:00:00	7.72
1	2012-08-08 01:00:00	7.72
1	2012-08-08 02:00:00	7.75
1	2012-08-08 03:00:00	7.78
1	2012-08-08 04:00:00	7.8
1	2012-08-08 05:00:00	7.81
1	2012-08-08 06:00:00	7.83

2.5. stationrainfall 表

存储了南京市水文站的降水数据，格式和 rainfall 一样，不赘述。

3. tuscany 项目

3.1. 新建工程

如果是测试 tuscany 自带的例子的话，直接导入即可。需要注意的是 tuscany 的例子都是 maven 工程。

如果新建工程的话，最好也是 maven 工程，这样后期我们加入对其它包的依赖的话都很方便。

3.2. 数据库连接池

可能存在很多个浏览器访问我们的服务，而且可能比较频繁，所以采用一个数据库连接池来帮助提高数据库访问的性能。

这里我们采用 proxool[2]，并将其封装在一个单例模式中，设置初始连接为 10。以后每次访问数据库获取连接时调用这个单例的 getConnection 方法获取一个空闲连接。

```
public enum DatabaseConnector {
    INSTANCES;
    DatabaseConnector() {
        // TODO Auto-generated constructor stub
        …Proxool连接池的初始化
    }
    public Connection getConnection(){
        Connection connection = null;
        try {
            connection = DriverManager.getConnection("proxool.test");
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return connection;
    }
}
```

3.3. pom 的依赖管理

编写 pom.xml 时可以参考例子，将例子中的依赖全部拷贝进来，同时我们需要加入对 mysql 驱动的依赖(mysql-connector-java)和对 proxool 的依赖，maven 库中有 mysql-connector-javajar 包，但是没有 proxool 的 jar 包，所以设置从本地路径中查找的方式。

```
    <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.0.3</version>
     </dependency>
    <dependency>
        <groupId>proxool</groupId>
        <artifactId>proxool-core</artifactId>
```

[2] Proxool 是一种 Java 数据库连接池技术。是 sourceforge 下的一个开源项目,这个项目提供一个健壮、易用的连接池，最为关键的是这个连接池提供监控的功能，方便易用，便于发现连接泄漏的情况。目前是和 DBCP 以及 C3P0 一起，最为常见的三种 JDBC 连接池技术。

```
        <version>1.1</version>
        <scope>system</scope>
        <systemPath>${basedir}/lib/proxool-core-1.1.jar</systemPath>
      </dependency>
      <dependency>
        <groupId>proxool</groupId>
        <artifactId>proxool-cglib</artifactId>
        <version>1.1</version>
        <scope>system</scope>
        <systemPath>${basedir}/lib/proxool-cglib-1.1.jar</systemPath>
      </dependency>
```

3.4. 编写服务和组件

3.4.1. 服务

我们需要 3 个服务：查询城市的降水量、查询南京市几个水文站的降水量、查询南京市几个水文站的水位。在 tuscany 中，定义一个服务要先定义一个接口，例如

```
@Remotable
public interface Rainfall {

    String getRainfall(String city, String time);
}
```

千万不能漏了@Remotable notion。

另外再定义一个 RainfallImpl 类实现这个接口，具体的 sql 语句也在这个类中编写。如查询某个城市某一时刻的从 0 点的累积降水量 sql 语句：

```
String sql = "select AMOUNT_ from rainfall, city   where TIME_ = ? and city.ALIAS_ = ? and city.ID_ = rainfall.id";
```

3.4.2. jax-rs 接口定义

我们将上面的查询城市降水量的服务暴露成 REST 风格的 web 服务，采用的是 jax-rs 接口。JaxrsHydrology 接口中标注了请求类型和参数。

```
@Remotable
public interface JaxrsHydrology {
    @GET
    @Path("getRainfall")
```

```
    @Produces(MediaType.TEXT_PLAIN)
    String getRainfall(@QueryParam("name") String name,
@QueryParam("time") String time);
}
```

3.4.3. composite 文件

Composite 文件用来定义服务和组件之间的关系，刚刚我们的查询城市降水量的服务可以按照如下定义

```
<component name="HydrologyComponent">
        <implementation.java class="rain.RainfallImpl"/>
        <service name="Rainfall">
            <interface.java interface="rain.JaxrsHydrology"/>
            <tuscany:binding.rest/>
        </service>
    </component>
```

其中，tuscany:binding.rest 指明了绑定的方式是 rest 风格的 web 服务，这里也体现了 tuscany 的强大之处，它能够提供很多方式的绑定，例如如果用 jsonp 的绑定方式，只需要将其替换成<tuscany:binding.jsonp/>。

我们这里一共有 3 个服务，且服务之间相对独立，简单起见也将其定义成 3 个组件，方式都一样，就不赘述其余两个服务以及它们是如何编写及绑定的。

3.5. 运行

万事俱备只欠东风，我们还需要一步就可以在 eclipse 中运行 tuscany 工程了，新建一个类，实现对上面 composite 文件的解析和服务及组件的装载即可。

```
Node node = TuscanyRuntime.runComposite("hydrology.composite", "target/classes");
```

在 eclipse 中选择运行方式为 maven build，即可在输出中找到对应服务的访问地址。当然也可以通过命令行中执行 tuscany run xxx 的方式。

4.总结

Tuscany 的功能还是十分强大的，它的功能不仅仅是提供多种形式的服务，正如引言所说，它可以将 WEB2.0 和 SOA 结合起来，前端也可在运行在 tuscany 中，作为 tuscany 组件的一部分，并通过引用更多的服务来提供服务的组合。