浅谈正则表达式在 web 系统中的应用

The application of Regular Expression on Web system

杜冬梅　许彩欣　苏健　（石家庄邮电职业技术学院　河北 石家庄 050021）

摘要：本文阐述了正则表达式的用途和用法，介绍了运用正则表达式在 web 系统中检测及限制数据格式、解析文本内容等方面的应用，并以 java 和 javascript 语言为例，对 web 系统客户端和服务端的常见应用进行了实现。

关键词：正则表达式　元字符

1　引言

随着 internet 的发展，web 应用越来越深入到人们的生活中。通过 internet 应用进行电子商务等重要活动时，一方面，web 应用经常需要把数据呈现给用户，用户操作完再次提交到服务器或者是第三方服务，如果服务方直接使用用户提交的数据，那么网站的安全就要受到威胁，而如果将数据的安全性验证交给服务器完成，必然加重网络传输和服务器的负担。正则表达式可以在 web 应用的客户端检测数据的完整性、安全性，另一方面，web 的服务端应用经常需要进行文本解析工作，需要拆分客户端上传的内容，分析后进行数据的操作处理。

正则表达式在客户端的前台认证和服务器端的数据处理方面能够起到很大的作用。本文将以前台的 javascript 和后台的 java 应用为例，介绍正则表达式的应用。

2　正则表达式

2.1　什么是正则表达式

正则表达式由美国数学家 Stephen Kleene 于 1956 年提出，主要用于描述正则集代数。正则表达式提供了一种从字符集合中搜寻特定字符串的机制[1]，正则表达式可以让用户通过使用一系列的特殊字符构建匹配模式，然后把匹配模式与数据文件、程序输入以及 WEB 页面的表单输入等目标对象进行比较，根据比较对象中是否包含匹配模式，执行相应的程序。正则表达式的使用，可以通过简单的办法来实现强大的功能[2]。

正则表达式可以应用在包括 Linux，HP 等在内的多种操作系统，支持正则表达式的语言也很多，如 PHP，C#，java，JavaScript 等，所以正则表达式已经在很多系统中得到广泛的应用。

2.2　什么时候使用正则表达式

使用正则表达式可以处理很多不同的任务，特别是对于 web 系统中常见的录入信息可以进行分析和检测。比如：对用户输入的数据如信用卡号或订单号进行确认；对用户在 Web 表单中输入的电子邮件地址进行确认；对某个聊天室的输入进行检查，确保其只含有正当的词语；为避免 SQL Injection 攻击而对输入进行检查等等。

总之，正则表达式功能强大，利用它可以加强 web 系统数据的完整性和安全性。

2.3　正则表达式的基本语法

正则表达式的形式为 /匹配模式/：

其中位于“/”定界符之间的部分就是将要在目标对象中进行匹配的模式。用户只要把希望查找匹配对象的模式内容放入“/”定界符之间即可。为了能够使用户更加灵活的定制模式内容，正则表达式提供了专门的“元字符”。

所谓元字符就是指那些在正则表达式中具有特殊意义的专用字符，可以用来规定其前导字符（即位于元字符前面的字符）在目标对象中的出现模式。

较为常用的元字符包括：“+”，“*”，以及“?”。

“+”规定其前导字符必须在目标对象中连续出现一次或多次。

“*”规定其前导字符必须在目标对象中出现零次或连续多次。

“?”规定其前导对象必须在目标对象中连续出现零次或一次。

“^”表示字符串的开始位置或多行匹配模式下每行的开始位置。

“$”表示字符串的结束位置或多行匹配模式下每行的结束位置。

以几个典型的正则表达式为例说明其含义。

(1) /fo+/。表达式中包含“+”元字符,表示可以与目标对象中的“fool”,“fo”,或者“football”等在字母 f 后面连续出现一个或多个字母 o 的字符串相匹配。

(2) /eg*/。表达式中包含“*”元字符,表示可以与目标对象中的“easy”,“ego”,或者“egg”等在字母 e 后面连续出现零个或多个字母 g 的字符串相匹配。

(3) /Wil?/。表达式中包含“?”元字符,表示可以与目标对象中的“Win”,或者“Wilson”,等在字母 i 后面连续出现零个或一个字母 l 的字符串相匹配。

其他主要元字符的使用方式如下。

\s:用于匹配单个空格符,包括 tab 键和换行符;

\S:用于匹配除单个空格符之外的所有字符;

\d:用于匹配从 0 到 9 的数字;

\w:用于匹配字母,数字或下划线字符;

\W:用于匹配所有与 \w 不匹配的字符;

.:用于匹配除换行符之外的所有字符。

在正则表达式中,可以用方括号括起若干个字符来表示一个元字符[3]。除元字符外,正则表达式支持限定符的概念。这些限定符可以指定正则表达式的一个给定组件必须要出现多少次才能满足匹配,因而可以适应不知道要匹配多少字符时的不确定情况。限定符有的使用说明如下。

(1) {n} n 是一个非负整数。匹配确定的 n 次。例如,'o{2}'不能匹配 "Bob" 中的 'o',但是能匹配 "food" 中的两个 o。

(2) {n,} n 是一个非负整数。至少匹配 n 次。例如,'o{2,}'不能匹配 "Bob" 中的 'o',但能匹配 "foooood" 中的所有 o。'o{1,}' 等价于 'o+'。'o{0,}' 则等价于 'o*'。

(3) {n,m} m 和 n 均为非负整数,其中 n <= m。最少匹配 n 次且最多匹配 m 次。例如,"o{1,3}" 将匹配 "fooooood" 中的前三个 o。'o{0,1}' 等价于 'o?'。

2.4 正则表达式的构造方法

(1) 文字量法创建正则表达式

使用文字量法创建正则表达式即将文字量的正则表达式赋值给一个变量。使用文字量法在 javascript 中创建正则表达式的一般格式为:

```
var varname = /pattern/;
```

(2) 构造函数 RegExp() 创建正则表达式

利用正则表达式 RegExp() 可以创建正则表达式。RegExp() 使用一个或两个参数:第一个参数指定正则表达式,第二个参数是可选参数,指定正则表达式的选项,其标记字符与文字量正则表达式使用的标记字符相同。例如:

```
var regex = new RegExp("my country","ig");
```

3 正则表达式在 web 系统中的应用

正则表达式在 web 系统中的应用非常广泛,只要理解了如何编写正则表达式,可以采用不同的途径利用它进行判断,比如:检测数据格式、解析文本数据文件、替换相关文本、提取感兴趣的文本内容等。验证的方法可以采用字符串的 match() 等方法,也可以采用正则表达式对象 RegExp 的 test() 和 exec() 方法。

下面从比较常见的几个应用入手,以 javascript 和 java 语言为例,说明正则表达式在 web 系统客户端和服务端中的应用。

3.1 检测客户端数据格式

检测数据格式的合法性是正则表达式的一项最常用的功能。正则表达式是一种形式化的字符串描述方法,只需很少的代码即可描述出应用遇到的任意字符串模式。如果遇到的字符串与正则表达式不符,说明数据格式不正确。常见的有各种输入内容的合法性检

查，下面以邮件地址，IP 地址，身份证号码的合法性验证举例说明。

（1）使用 RegExp 对象的 test( ) 方法检测邮件地址是否合法

```
function checke_mail(e_mail)
{
  var = new RegExp(" /^[_A-Za-z0-9-]+(。[_A-Za-z0-9-]+)*@[A-Za-z0-9-]+(。[A-Za-z0-9-]+)*$/");
  if(e_mail != "") {
    if(! e_mailstr.test(e_mail)) {
      alert("E-mail 地址有误！");
      return false;
    }
    return true;
  }
  else{
    alert("E-mail 地址不能为空！");
    return false;
  }
  return true;
}
```

上面的关键是分析 email 的正则表达式。上述 email 正则表达式负责验证 email 的合法性，一个合法的 email 由两部分组成，第一部分是用户名，第二部分是 email 服务器所在的域名。用户名可以包含"."，比如"dongmei.xu"，也可以包含数字字符，英语字母，及"_"和"-"，而正则表达式中"@"前面的就表示了用户名的规则。[_A-Za-z0-9-]+表示 1 到多个字母字符，数字字符下划线中划线的串，(。[_A-Za-z0-9-]+)*代表由"."开头的后面跟若干个字母数字及"-"及"_"的串，由于合法的 email 可以出现若干个"."分割的部分，所以后面跟一个*，代表 0 个到多个，后面的域名部分可以照此分析。

（2）采用字符串的 match( ) 方法检测 Ip 地址是否合法

IP 地址的合法性要求需要四个数字域，每个域之间由"."分割，其中每个域小于等于 255。IP 地址的认证原则是，每个域的数字有 5 种可能，1）250－255，?2）20?　－24? 3）1?? 4）?? 5）?

上面每个? 代表任意一个数字。

```
function isIp(ip)
{
  var ip_ip = "(25[0-5]|2[0-4]\d|1\d\d|\d\d|\d)";
  var ip_ipdot = ip_ip + "\.";
  //下面是 ip 地址的正则表达式，将每个域的表达式连接起来，最后一个不加"."。
  var isIPaddress = "/^" + ip_ipdot + ip_ipdot + ip_ipdot + ip_ip + "$";
  var matcharr = new Array();
  matcharr = ip.match(isIPaddress); //匹配结果放入数组
  if(matcharr.length == =0) //数组长度为 0 表示 Ip 不正确
  return false;
  else
  return true;
}
```

（3）采用 test( ) 方法验证身份证号码是否合法

以下代码验证身份证的长度和内容是否正确，为简单起见，其中检查生日和所在地市部分代码省略，相关的错误提示也省略。

```
function checkid_num(id_num) {
  var id_numstr = /^\d{1,18}$/;
  if(id_num != "") {
    if(! id_numstr.test(id_num)) {
      return false;}
    if(len == = 15 || len == = 18) {
      if(len == = 18) {
        var str_a = id_num.substring(6,14);
        re = /^[0-9]{17}[0-9,x,X]{1}$/;
        if (! re.test(id_num)) {
          return false;}
      }
```

```
        else{re = /^[0-9]{15}$/
          if(! re.test(id_num)){return false;}
      }
        }
      else{return false;}
    }
    else{return false;}
    return true;
  }
```

身份证要求合法位数为 15 位或 18 位，如果 18 位，前 17 位为数字，最后一位为数字或 X,x，所以正则表达式为"/^[0-9]{17}[0-9,x,X]{1}$/"，而 15 位的要求全部为数字，所以"/^[0-9]{15}$/"。

3.2 使用 replace()限定网页表单输入框内容

对客户端程序来说，输入项目的种类很多，在 C/S 模式开发下，一般采用以控件的属性限制输入内容的方法，而在 web 系统中，以正则表达式也可以达到同样的效果。

下面的正则表达式表示如果输入的内容不符合正则表达式的格式，要将输入的字符串以空代替，表现的结果即用户不能输入不合法内容。例如：

(1) 限制输入框只能中文输入

```
onkeyup="value=value.replace(/[^\u4E00-\u9FA5]/g,'')" onbeforepaste=
"clipboardData.setData('text',clipboardData.getData('text').replace(/[^\u4E00-\u9FA5]/g,''))"
```

(2) 限制输入框只能输入数字

```
onkeyup="value=value.replace(/[^\d]/g,'')" onbeforepaste=
"clipboardData.setData('text',clipboardData.getData('text').replace(/[^\d]/g,''))"
```

(3) 限制输入框只能输入数字和英文

```
onkeyup="value=value.replace(/[\W]/g,'')" onbeforepaste=
"clipboardData.setData('text',clipboardData.getData('text').replace(/[^\d]/g,''))"
```

3.3 服务端使用 split()按照正则表达式进行文本拆分，并写入数据库

实际应用中，经常遇到将某文本信息导入数据库表的要求，该类操作常用于批量的插入、删除、更新操作。Web 应用中，为了简化传输接口，常由前台采用字符串上传方式将文本文件上传，后台服务端对字符串进行拆分，实现相关数据库操作。

字符串对象的 split()方法支持正则表达式，该方法将字符串分割为几个部分，并保存在数组中。使用 split()拆分文本信息，形成既定格式的数据内容，进行相关数据库操作，是一种简单、高效的方法。

下面的例子将 string 中的内容拆分到数组中，然后将数组中的各个元素作为数据库表的不同字段插入到数据库中。

```
String[] strlst;
String str="李军,女,1987,计算机系"; //字符串信息
String regex=",";s
strlst=str.split(regex);
……
```

拆分之后，可以将 strlst 中的内容作为数据库表中各个字段(姓名，性别，入校时间，系)的内容插入到数据库中。

4 总结

正则表达式在很多语言中都会用到，它用一些特殊的符号灵活的生成各种文本模式，即正则表达式，来匹配符合该模式的所有文本串，达到检测数据有效性、替换文本、解析文本内容、查找文本内容的作用。

本文只是在 web 系统中几个常用的方面论述了正则表达式的应用，正则表达式因其灵活形应用范围非常广，希望本文能为同行起到抛砖引玉的作用。

参考文献

1 张长富、黄中敏，javascript 动态网页编程实例手册[M]，海洋出版社，2005.

2 (美)埃克尔(Eckel,B.) 陈昊鹏 等译，Java 编程思想，第 3 版，机械工业出版社，2005.

3 耿祥义、张跃平，java2 实用教程[M]，清华大学出版社，2006[3].