

Research of Conversion Method of Entity Object and JSON Data

Zhang Yu

Guizhou university for nationalities
Guiyang 550025, China

Abstract—As the constant improvement of requirements of software projects and the rapid development of the internet, object-oriented web projects[1] have become the first choice of software solution. In a web object, the entity object is the intermediary of data transmission, and the data need be transmitted to different clients through the internet at last. At the same time, because of the corporations' constant renewal of software and continuously increasing of the data, multi-project cooperative work has been the mainstream. It is the key object of web program study how to solve the problems of data compatibility and data conversion. To realize the convenient and fast data compatibility and data conversion in web objects, the paper presents an analysis and a summary of the conversion method of the entity object and the popular and light-weight JSON data.

Keywords—transmission, object, data, JSON

I. INTRODUCTION

With the rapid development of application of computer technique in every field, sophisticated technologies in most regions are controlled by computers which are the indispensable parts. The core of the control part is the designed program and the data. The program controls the operation mode of a computer, and the data is the note of the results of various activities which is the most important element of scientific researches and enterprise decisions.

For the constant improvement of corporations' requirements and the rapid development of the network technologies, programming language converts from the former process-oriented programming to the current mainstream of object-oriented one, and develops from single machine programs to web ones. The data develop from taking Kb as a unit to Tb as a unit. The system also expands from the original single project to cooperation of multi- projects. Data security and data transmission become the chief problems in programming at present. There are two research emphasises in web system. One is how to guarantee the non-blocking of the internet in the transmission, the other is the data compatibility in various programmes.

At present, owing to the open style of XML[2] data and JSON data, they are compatible with most programs and used by many web projects. Based on XML (eXtensibleMarkupLanguage) spatial data interoperability standard way is to achieve interoperability of spatial information on the Internet sharing and effective way of interoperability.[3] However, XML is a structured document

which needs to be analyzed manually by both the server and the client. It will occupy more system resources. So, using XML to switch the data will lead to problems of poor performance, inadequate compatibility, and low sensitivity. While the data of JSON form are designed for the browser language of Javascript. The generated JSON object can be analyzed directly and it is popular for its simplicity, being read easily and small data. In object-oriented web project, the data are transmitted through entity objects, but lightweighted JSON applies text format storage totally independent of language, which causes no small obstacles.

Through the former description, to make the data transmit more smoothly in web projects, JSON has become a popular data format. It is a lightweighted kind of data, compatible with most programming languages. While, for object-oriented language can not identify the data automatically now, the programmer has to spend much time in data analysis to realize the data conversion. To solve the data conversion of the entity object and JSON, the paper designs an automatic conversion method of the object based on C# language and JSON. Using this method, programmers can achieve the conversion of the entity object and the data easily, and put more time and energy to operation functions of a project.

II. JSON DATA

JSON(JavaScript Object Notation) is a lightweighted data switching format. It is easy to read, analyzed and created. It is a subset based on JavaScript (Standard ECMA-262 3rd Edition - December 1999) . JSON applies text format totally independent of language, but also uses customs similar to c language family including C, C++, C#, Java, JavaScript, Perl, Python and so on, which makes JSON an ideal data switching language.

JSON is constructed in two structure:

1. A collection of name/value pairs. In different languages, it is comprehended as an object, a record, a struct, a dictionary, a hash table, a keyed list, or an associative array.
2. An ordered list of values. In most languages, it is an array.

JSON is expressed as follows:

```
{ province:[
  {
    name:"Hei long jiang",
    citys:{
      city:["Ha'er bin","Da qin"]
    }
  },
  }
```

```

{
    name:"Guang dong",
    citys:{
    city:["Guang zhou","Shen zhen","Zhu hai"]
    }
}
] }

```

III. THEORY OF CONVERSION

Considering the universal property of different entity objects and Generic collection, the paper applied the reflection mechanism[4] of C# language in switching of entity objects and JSON.

As, in the process of switching, we can not ensure that the one needed to be converted is an object or a set, and the type of the object. While in program collection of C#, class object provides access classifier data. Access classifier data include detailed information like the construction method, the field, the attribute and the method. As the following figure shows:

Through inducting System.Reflection in C#, on the basis of the name transmitted by the user, object can be loaded dynamically by calling Assembly.Load.

After getting objects, PropertyInfo can be achieved by GetProperties. Through Name and GetValue in PropertyInfo, we can get the name of the attribute and the value, then make up the character string of JSON format.

As converting from JSON to objects, the name and the value of object will be obtained through regular expression class[5] of Regex in C#. Because the minimum unit of JSON is made up of the content in "[]", we can get MatchCollection set through the character string of "(\\\"(?<key>[^\"]+\")\\\":\\\"(?<value>[^\"]+\")\\\"|\\\"(?<key>[^\"]+\")\\\":(?<value>[^\"]+\")\\\"|(?<value>[^\"]+\")\" which is matching to Matches of Regex. After searching the set, we can get the name of attribute and the value, and at last evaluate to certain attribute of the entity object.

IV. DESIGN AND REALIZATION

The paper applied Visual Studio 2008 C# of Microsoft, and founded two classes, ConvertToJSON and ConvertToObject.

A. conversion from objects to JSON

The class of ConvertToJSON is used to realize the conversion from objects to JSON. This category needs to quote System.Reflection to name space and achieve analysis by reflection. The class of ConvertToJSON includes two functions: the conversion from set to JSON and the conversion from objects to JSON.

1) conversion from set to JSON

For all list sets integrate IList, when transferring set parameter, IList can be used as parameter.

The detailed realization code is as follows:

```

public static string ListToJson<T>(IList<T> list, string
Name)
{
    StringBuilder jsonStr = new StringBuilder();
    if (string.IsNullOrEmpty(jsonStr))
        Name = list[0].GetType().Name;
    jsonStr.Append("\"\" + Name + \"\":[\"");
    if (list.Count > 0)
    {
        for (int i = 0; i < list.Count; i++)
        {
            T obj =
Activator.CreateInstance<T>();
            PropertyInfo[] propertyInfos =
obj.GetType().GetProperties();
            jsonStr.Append("{");
            for (int j = 0; j <
propertyInfos.Length; j++)
            {
                Type type = propertyInfos
[j].GetValue(list[i], null).GetType();
                jsonStr.Append("\"\" + propertyInfos[j].Name.ToString()
+ \":\" + String.Format(propertyInfos [j].GetValue(list[i],
null).ToString(), type));
                if (j <
propertyInfos.Length - 1)
                {
                    jsonStr.Append(",");
                }
            }
            jsonStr.Append("}");
            if (i < list.Count - 1)
            {
                jsonStr.Append(",");
            }
        }
    }
    jsonStr.Append("]");
    return jsonStr.ToString();
}

```

2) conversion from object to JSON

The realization pattern of the conversion from objects to JSON is approximately similar to set conversion, while it need not conduct searching loop to the list.

The detailed realization code is as follows:

```

public static string ObjectToJson<T>(T obj, string
Name)
{
    StringBuilder jsonStr = new StringBuilder();
    T obj = Activator.CreateInstance<T>();
    PropertyInfo[] propertyInfos =
obj.GetType().GetProperties();
    jsonStr.Append("{");
    for (int j = 0; j < propertyInfos.Length; j++)

```

```

    {
        Type type = propertyInfos
[j].GetValue(list[i], null).GetType();
        jsonStr.Append("\""
+ propertyInfos[j].Name.ToString() + "\":\"
+ StringFormat(propertyInfos
[j].GetValue(list[i], null).ToString(),
type));
        if (j < propertyInfos.Length - 1)
        {
            jsonStr.Append(",");
        }
        jsonStr.Append("}");
        return jsonStr.ToString();
    }
}

```

B. conversion from JSON data to object

The class of ConvertToObject is used to realize the conversion from JSON data to the object, and it need quote System.Text.RegularExpressions to name a space whose regular expression class of Regex is demanded. Through the character string of "(\\\"(?<key>[^\"]+\\\")\\\":\\\"(?<value>[^\"]+\\\")\\\")|(\\\"(?<key>[^\"]+\\\")\\\":(?<value>[^\"]+\\\")\\\")" which is matching to the regular expression, the value of JSON data will be achieved, then is filled in object. Among them, the key and the value are the key values obtained through the regular expression.

The detailed realization code is as follows:

```

MatchCollection
matches=Regex.Matches(text,"(\\\"(?<value>[^\"]+\\\")\\\")(?<
value>[^\"]+\\\")");

```

```

foreach (Match match in matches)
{
    string value = match.Groups["value"].Value;
    jsonArray.Add(_json.ContainsKey(value) ?_json[value] :Str
Decode(value));
    base.Add(key, value);
}

```

V. CONCLUSION

The conversion method of the paper makes use of the reflection mechanism of C# and regular expressions, to realize a simple conversion function of entity object and JSON.

The paper also provides an automatic conversion thought of object and text data which is fit for the conversion of data formats like the XML and the object without complex analyzing process. But, for applying the reflection mechanism of C# in the process of analysis, great data will affect the performance efficiency of programs to a certain extent.

REFERENCES

- [1] Hao-qiang Tan. Object-oriented Design (second edition) [M]. Beijing: Tsinghua University Press, 2003.9.
- [2] xiao-fei Sun. The Foundation course of XML and Experiment Instruction [M]. Beijing: Tsinghua University Press, 2008.11.
- [3] Min Han, Feng hao. Research of GIS data exchange method based on JSON [J]. science of surveying and mapping, 2010. 1.
- [4] Xue-feng Liao. An introduction to JSON <http://www.ibm.com/developerworks/cn/web/wa-lo-json/?ca=drs-tp3308>. 2008.8.
- [5] Andrew Troelsen. Programming of C#与.NET 3.5(fourth edition). Beijing: Posts &Telecom Press, 2009.2.
- [6] AndrewWatt. Beginning Regular Expressions[M]. Beijing: Tsinghua University Press, 2008.10.

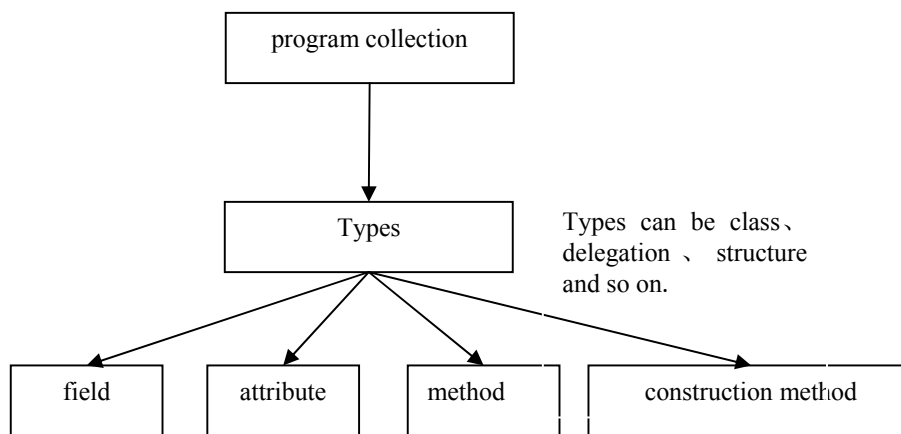


Figure 1. The structure diagram of program collection