**Unity3D中任意类型数据转换成json数据的方法**

Posted on 2013年03月20日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 275 次

[**Unity3D**](http://www.unitymanual.com)中任意类型数据转换成json数据的方法：

/// <summary>  
/// List转成json  
/// </summary>  
/// <typeparam name="T"></typeparam>  
/// <param name="jsonName"></param>  
/// <param name="list"></param>  
/// <returns></returns>  
public static string ListToJson<T>(IList<T> list, string jsonName)  
{  
StringBuilder Json = new StringBuilder();  
if (string.IsNullOrEmpty(jsonName))  
jsonName = list[0].GetType().Name;  
Json.Append("{\"" + jsonName + "\":[");  
if (list.Count > 0)  
{  
for (int i = 0; i < list.Count; i++)  
{  
T obj = Activator.CreateInstance<T>();  
PropertyInfo[] pi = obj.GetType().GetProperties();  
Json.Append("{");  
for (int j = 0; j < pi.Length; j++)  
{  
Type type = pi[j].GetValue(list[i], null).GetType();  
Json.Append("\"" + pi[j].Name.ToString() + "\":" + StringFormat(pi[j].GetValue(list[i], null).ToString(), type));

if (j < pi.Length - 1)  
{  
Json.Append(",");  
}  
}  
Json.Append("}");  
if (i < list.Count - 1)  
{  
Json.Append(",");  
}  
}  
}  
Json.Append("]}");  
return Json.ToString();  
}

/// <summary>  
/// List转成json  
/// </summary>  
/// <typeparam name="T"></typeparam>  
/// <param name="list"></param>  
/// <returns></returns>  
public static string ListToJson<T>(IList<T> list)  
{  
object obj = list[0];  
return ListToJson<T>(list, obj.GetType().Name);  
}

/// <summary>  
/// 对象转换为Json字符串  
/// </summary>  
/// <param name="jsonObject">对象</param>  
/// <returns>Json字符串</returns>  
public static string ToJson(object jsonObject)  
{  
string jsonString = "{";  
PropertyInfo[] propertyInfo = jsonObject.GetType().GetProperties();  
for (int i = 0; i < propertyInfo.Length; i++)  
{  
object objectValue = propertyInfo[i].GetGetMethod().Invoke(jsonObject, null);  
string value = string.Empty;  
if (objectValue is DateTime || objectValue is Guid || objectValue is TimeSpan)  
{  
value = "'" + objectValue.ToString() + "'";  
}  
else if (objectValue is string)  
{  
value = "'" + ToJson(objectValue.ToString()) + "'";  
}  
else if (objectValue is IEnumerable)  
{  
value = ToJson((IEnumerable)objectValue);  
}  
else  
{  
value = ToJson(objectValue.ToString());  
}  
jsonString += "\"" + ToJson(propertyInfo[i].Name) + "\":" + value + ",";  
}  
jsonString.Remove(jsonString.Length - 1, jsonString.Length);  
return jsonString + "}";  
}

/// <summary>  
/// 对象集合转换Json  
/// </summary>  
/// <param name="array">集合对象</param>  
/// <returns>Json字符串</returns>  
public static string ToJson(IEnumerable array)  
{  
string jsonString = "[";  
foreach (object item in array)  
{  
jsonString += ToJson(item) + ",";  
}  
jsonString.Remove(jsonString.Length - 1, jsonString.Length);  
return jsonString + "]";  
}

/// <summary>  
/// 普通集合转换Json  
/// </summary>  
/// <param name="array">集合对象</param>  
/// <returns>Json字符串</returns>  
public static string ToArrayString(IEnumerable array)  
{  
string jsonString = "[";  
foreach (object item in array)  
{  
jsonString = ToJson(item.ToString()) + ",";  
}  
jsonString.Remove(jsonString.Length - 1, jsonString.Length);  
return jsonString + "]";  
}

/// <summary>  
/// Datatable转换为Json  
/// </summary>  
/// <param name="table">Datatable对象</param>  
/// <returns>Json字符串</returns>  
public static string ToJson(DataTable dt)  
{  
StringBuilder jsonString = new StringBuilder();  
jsonString.Append("[");  
DataRowCollection drc = dt.Rows;  
for (int i = 0; i < drc.Count; i++)  
{  
jsonString.Append("{");  
for (int j = 0; j < dt.Columns.Count; j++)  
{  
string strKey = dt.Columns[j].ColumnName;  
string strValue = drc[i][j].ToString();  
Type type = dt.Columns[j].DataType;  
jsonString.Append("\"" + strKey + "\":");  
strValue = StringFormat(strValue, type);  
if (j < dt.Columns.Count - 1)  
{  
jsonString.Append(strValue + ",");  
}  
else  
{  
jsonString.Append(strValue);  
}  
}  
jsonString.Append("},");  
}  
jsonString.Remove(jsonString.Length - 1, 1);  
jsonString.Append("]");  
return jsonString.ToString();  
}

/// <summary>  
/// DataTable转成Json  
/// </summary>  
/// <param name="jsonName"></param>  
/// <param name="dt"></param>  
/// <returns></returns>  
public static string ToJson(DataTable dt, string jsonName)  
{  
StringBuilder Json = new StringBuilder();  
if (string.IsNullOrEmpty(jsonName))  
jsonName = dt.TableName;  
Json.Append("{\"" + jsonName + "\":[");  
if (dt.Rows.Count > 0)  
{  
for (int i = 0; i < dt.Rows.Count; i++)  
{  
Json.Append("{");  
for (int j = 0; j < dt.Columns.Count; j++)  
{  
Type type = dt.Rows[i][j].GetType();  
Json.Append("\"" + dt.Columns[j].ColumnName.ToString() + "\":" + StringFormat(dt.Rows[i][j].ToString(), type));  
if (j < dt.Columns.Count - 1)  
{  
Json.Append(",");  
}  
}  
Json.Append("}");  
if (i < dt.Rows.Count - 1)  
{  
Json.Append(",");  
}  
}  
}  
Json.Append("]}");  
return Json.ToString();  
}

/// <summary>  
/// DataReader转换为Json  
/// </summary>  
/// <param name="dataReader">DataReader对象</param>  
/// <returns>Json字符串</returns>  
public static string ToJson(DbDataReader dataReader)  
{  
StringBuilder jsonString = new StringBuilder();  
jsonString.Append("[");  
while (dataReader.Read())  
{  
jsonString.Append("{");  
for (int i = 0; i < dataReader.FieldCount; i++)  
{  
Type type = dataReader.GetFieldType(i);  
string strKey = dataReader.GetName(i);  
string strValue = dataReader[i].ToString();  
jsonString.Append("\"" + strKey + "\":");  
strValue = StringFormat(strValue, type);  
if (i < dataReader.FieldCount - 1)  
{  
jsonString.Append(strValue + ",");  
}  
else  
{  
jsonString.Append(strValue);  
}  
}  
jsonString.Append("},");  
}  
dataReader.Close();  
jsonString.Remove(jsonString.Length - 1, 1);  
jsonString.Append("]");  
return jsonString.ToString();  
}

/// <summary>  
/// DataSet转换为Json  
/// </summary>  
/// <param name="dataSet">DataSet对象</param>  
/// <returns>Json字符串</returns>  
public static string ToJson(DataSet dataSet)  
{  
string jsonString = "{";  
foreach (DataTable table in dataSet.Tables)  
{  
jsonString += "\"" + table.TableName + "\":" + ToJson(table) + ",";  
}  
jsonString = jsonString.TrimEnd(',');  
return jsonString + "}";  
}

/// <summary>  
/// 过滤特殊字符  
/// </summary>  
/// <param name="s"></param>  
/// <returns></returns>  
private static string String2Json(String s)  
{  
StringBuilder sb = new StringBuilder();  
for (int i = 0; i < s.Length; i++)  
{  
char c = s.ToCharArray()[i];  
switch (c)  
{  
case '\"':  
sb.Append("\\\""); break;  
case '\\':  
sb.Append("\\\\"); break;  
case '/':  
sb.Append("\\/"); break;  
case '\b':  
sb.Append("\\b"); break;  
case '\f':  
sb.Append("\\f"); break;  
case '\n':  
sb.Append("\\n"); break;  
case '\r':  
sb.Append("\\r"); break;  
case '\t':  
sb.Append("\\t"); break;  
default:  
sb.Append(c); break;  
}  
}  
return sb.ToString();  
}

/// <summary>  
/// 格式化字符型、日期型、布尔型  
/// </summary>  
/// <param name="str"></param>  
/// <param name="type"></param>  
/// <returns></returns>  
private static string StringFormat(string str, Type type)  
{  
if (type == typeof(string))  
{  
str = String2Json(str);  
str = "\"" + str + "\"";  
}  
else if (type == typeof(DateTime))  
{  
str = "\"" + str + "\"";  
}  
else if (type == typeof(bool))  
{  
str = str.ToLower();  
}  
return str;  
}