**DeWeb控件开发指南**

（2020-08-12 v1.0）

1. **什么是DeWeb?**

DeWeb是一个可以直接将Delphi程序快速转换为网页应用的工具！

使用DeWeb, 开发者不需要学习HTML、JavaScript、Java、PHP、ASP、C#等新知识，用Delphi搞定一切。

DeWeb开发的网页支持所有客户端，包括手机、平板等。

网址：<https://gitee.com/fenglinyushu/dewebsdk>

1. **如何自行开发控件以支持DeWeb?**

自行开发控件很简单，只需要开发一个适合需要的DLL即可。

注意几点：

1. uses的第1个单元必须是sharemem;
2. 必须包含5个函数dwGetExtra,dwGetEvent,dwGetHead,dwGetTail,dwGetData，各个函数的参数请和附件DEMO中的保持一致；

每个函数（除dwGetEvent外）都返回一个JSON字符串数组类型字符，如：

[

‘Edit1\_\_lef:10px’,

‘Edit1\_\_top:10px’,

‘Edit1\_\_wid:60px’,

‘Edit1\_\_hei:35px’,

‘Edit1\_\_txt:"Edit1"’

]

1. DLL输出或复制到RunTime\dwVcls\目录中；
2. dwGetExtra用于在html中增加对第三方网页控件的支持。

目前DeWeb中默认采用element-ui进行处理。如果element-ui不存在某控件，比如TChart，就需要引入第三方UI库。

在dwGetExtra中可以增加以下代码，以实现第三方UI库支持。

joRes.Add('<script src="dist/charts/echarts.min.js"></script>');

joRes.Add('<script src="dist/charts/lib/index.min.js"></script>');

joRes.Add('<link rel="stylesheet" href="dist/charts/lib/style.min.css">');

1. dwGetEvent用于处理控件事件

有2个参数

* ACtrl:TComponent，控件自身
* AData:String，控件信息，为JSON字符串，一般应先转换为JSON对象，比如joData。其中joData.event为事件名称，如“onchange”或“onclick”等（大小写敏感）；joData.value为值，如Edit的Value,或CheckBox的Checked等；

1. dwGetHead用于生成当前控件改造时的头部信息，同时包含是否可见，是否可用，Left/Top/Width/Height，Caption等信息。
2. dwGetTail用于生成当前控件改造时的尾部信息，一般与dwGetHead中的HTML要素对应即可。
3. dwGetData用于当前控件的各项关键参数值。
4. **常见问题**
5. 运行时报内存异常

请把uses的第1个单元设置为sharemem;

1. JSON是什么？

[JSON](https://baike.baidu.com/item/JSON)([JavaScript](https://baike.baidu.com/item/JavaScript) Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 [ECMAScript](https://baike.baidu.com/item/ECMAScript) (欧洲计算机协会制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

如果不会的同学，强烈建议学习并熟练掌握！

1. **联系我们**

网址：<https://gitee.com/fenglinyushu/dewebsdk>

QQ群：120283369

1. **修订历史**
2. V1.0 2020－08－12

创建本文档

**附件：**

library dwTEdit;

uses

ShareMem, //必须添加

//

dwCtrlBase, //一些基础函数

//

SynCommons, //mormot用于解析JSON的单元

//

SysUtils,

Classes,

Dialogs,

StdCtrls,

Windows,

Controls,

Forms;

//当前控件需要引入的第三方JS/CSS ,一般为不做改动,目前仅在TChart使用时需要用到

function dwGetExtra(ACtrl:TComponent):string;stdCall;

var

joRes : Variant;

begin

//生成返回值数组

joRes := \_Json('[]');

{

//以下是TChart时的代码,供参考

joRes.Add('<script src="dist/charts/echarts.min.js"></script>');

joRes.Add('<script src="dist/charts/lib/index.min.js"></script>');

joRes.Add('<link rel="stylesheet" href="dist/charts/lib/style.min.css">');

}

//

Result := joRes;

end;

//根据JSON对象AData执行当前控件的事件, 并返回结果字符串

function dwGetEvent(ACtrl:TComponent;AData:String):string;StdCall;

var

joData : Variant;

begin

//

joData := \_Json(AData);

if joData.event = 'onchange' then begin

//保存事件

TEdit(ACtrl).OnExit := TEdit(ACtrl).OnChange;

//清空事件,以防止自动执行

TEdit(ACtrl).OnChange := nil;

//更新值

TEdit(ACtrl).Text := dwUnescape(joData.value);

//恢复事件

TEdit(ACtrl).OnChange := TEdit(ACtrl).OnExit;

//执行事件

if Assigned(TEdit(ACtrl).OnChange) then begin

TEdit(ACtrl).OnChange(TEdit(ACtrl));

end;

end else if joData.event = 'onenter' then begin

//保存事件

TEdit(ACtrl).OnExit := TEdit(ACtrl).OnEnter;

//清空事件,以防止自动执行

TEdit(ACtrl).OnEnter := nil;

//更新值

TEdit(ACtrl).Text := dwUnescape(joData.value);

//恢复事件

TEdit(ACtrl).OnEnter := TEdit(ACtrl).OnExit;

//执行事件

if Assigned(TEdit(ACtrl).OnEnter) then begin

TEdit(ACtrl).OnEnter(TEdit(ACtrl));

end;

end;

//清空OnExit事件

TEdit(ACtrl).OnExit := nil;

end;

//取得HTML头部消息

function dwGetHead(ACtrl:TComponent):string;StdCall;

var

sCode : string;

joHint : Variant;

joRes : Variant;

begin

//生成返回值数组

joRes := \_Json('[]');

//取得HINT对象JSON

joHint := dwGetHintJson(TControl(ACtrl));

with TEdit(ACtrl) do begin

sCode := '<el-input'

+dwVisible(TControl(ACtrl)) //用于控制可见性Visible

+dwDisable(TControl(ACtrl)) //用于控制可用性Enabled(部分控件不支持)

+dwIIF(PasswordChar=#0,'',' show-password') //是否为密码

+' v-model="'+Name+'\_\_txt"' //

+dwGetHintValue(joHint,'placeholder','placeholder','') //placeholder,提示

+dwGetHintValue(joHint,'prefix-icon','prefix-icon','') //前置Icon

+dwGetHintValue(joHint,'suffix-icon','suffix-icon','') //后置Icon

+dwLTWH(TControl(ACtrl)) //Left/Top/Width/Height

+'"' // 封闭style

+Format(\_DWEVENT,['input',Name,'(this.'+Name+'\_\_txt)','onchange','']) //绑定OnChange事件

+Format(\_DWEVENT,['focus',Name,'(this.'+Name+'\_\_txt)','onenter','']) //绑定OnEnter事件

+'>';

//添加到返回值数据

joRes.Add(sCode);

end;

//

Result := (joRes);

end;

//取得HTML尾部消息

function dwGetTail(ACtrl:TComponent):string;StdCall;

var

joRes : Variant;

begin

//生成返回值数组

joRes := \_Json('[]');

//生成返回值数组

joRes.Add('</el-input>'); //此处需要和dwGetHead对应

//

Result := (joRes);

end;

//取得Data消息, ASeparator为分隔符, 一般为:或=

function dwGetData(ACtrl:TComponent;ASeparator:String):string;StdCall;

var

joRes : Variant;

begin

//生成返回值数组

joRes := \_Json('[]');

//

with TEdit(ACtrl) do begin

joRes.Add(Name+'\_\_lef'+ASeparator+'"'+IntToStr(Left)+'px"');

joRes.Add(Name+'\_\_top'+ASeparator+'"'+IntToStr(Top)+'px"');

joRes.Add(Name+'\_\_wid'+ASeparator+'"'+IntToStr(Width)+'px"');

joRes.Add(Name+'\_\_hei'+ASeparator+'"'+IntToStr(Height)+'px"');

//

joRes.Add(Name+'\_\_vis'+ASeparator+''+dwIIF(Visible,'true','false'));

joRes.Add(Name+'\_\_dis'+ASeparator+''+dwIIF(Enabled,'false','true'));

//

joRes.Add(Name+'\_\_txt'+ASeparator+'"'+(Text)+'"');

end;

//

Result := (joRes);

end;

exports

dwGetExtra,

dwGetEvent,

dwGetHead,

dwGetTail,

dwGetData;

begin

end.