

E9流程表单前端接口API

1.说明

1.1 简介

所有接口统一封装在全局对象window.WfForm中

部分接口存在使用范围，最低kb版本以及是否移动端/PC端独有。没有特殊注明情况下通用

表单字段相关操作，不推荐使用jQuery，禁止原生JS直接操作DOM结构！

大家开发过程中，推荐都使用API接口操作，由产品统一运维；同时使用API才能完整的兼容移动终端

1.2 移动端兼容

WfForm对象下接口，兼容新版移动端EM7

由于API接口在PC端与移动端已经统一，为减少开发工作量以及后期维护成本；故EM7表单在移动终端不再引入workflow_base表custompage4emobile列作为自定义页面，直接引入custompage列(与PC模板一致)作为自定义页面

前端（JS方法）区分终端：

可通过方法WfForm.isMobile()判断是否移动端

```
var isMobile = WfForm.isMobile(); //true表示是eMobile、微信、钉钉等移动终端，false代表PC端
```

后端请求（自定义页面等）区分终端：

```
boolean isMobile = "true".equals(request.getParameter("_ec_ismobile")); //true表示是eMobile、微信、钉钉等移动终端，false代表PC端
```

1.3 前端代码开发方式

方式1：模板上代码块，针对单个节点，在显示/打印/移动模板单独配置

方式2：workflow_base表custompage，针对当前路径所有节点，显示及打印及移动端 注意流程如果与预算、考勤、车辆等关联此字段可能有值。

方式3：路径管理-应用设置-全局自定义页面，针对系统所有非模板模式的场景(PC及移动)。注意此页面为全局custompage，应避免写ready、checkCustomize等全局函数，只定义些函数体

特别注意：方式二、方式三禁止引入init_wev8.js。

如遇配置不生效，请先将代码块/custompage仅写alert确认是否生效，再逐步排查错误原因。

1.4 PC端打开表单的方式

新建请求：传参路径id，会自动计算活动版本的路径id

```
window.open("/workflow/request/CreateRequestForward.jsp?workflowid=747");
```

查看请求：传参请求id，用户需本身具备此请求查看权限，主次账号需带入账号信息

```
window.open("/workflow/request/ViewRequestForwardSPA.jsp?requestid=5963690");
```

1.5 移动端打开表单的方式

移动端表单链接：

```
//新建链接，传参路径id
var createUrl = "/spa/workflow/forwardMobileForm.html?
iscreate=1&workflowid=747";
//查看链接，传参请求id
var viewUrl = "/spa/workflow/forwardMobileForm.html?requestid=4503066";
```

第一种方式(推荐)：调用封装好的方法 如果是通过移动端脚手架打包的模块，可以直接调用 如果是自行开发的界面，需要引入/spa/coms/openLink.js

最低支持版本：KB900190601

openLink.openWorkflow(url, callbackFun, returnUrl)

参数	参数类型	说明
url	String	打开表单的链接
callbackFun	Function	仅限EM客户端，返回时的回调函数
returnUrl	String	非EM客户端，返回/提交后到指定链接

```
window.openLink.openWorkflow(createUrl, function(){
    alert("E-mobile打开表单链接，返回或提交后触发此回调函数");
});
```

```
//非EM终端打开，返回或提交后返回到流程中心界面
window.openLink.openWorkflow(createUrl, null,
"/spa/workflow/static4mobile/index.html#/center/doing");
```

第二种方式：仅限于EM客户端，打开表单并可控制表单返回/提交后事件回调 使用EM-SDK，弹webview方式实现

```
//分为两步，第一步调用SDK弹webview，第二步调用SDK控制回调刷新
window.em.openLink({
    url: viewUrl,
    openType: 2
});
window.em.ready(function(){
    window.em.registerBroadcast({
        name: "_closeWfFormCallback",
        action: function (argument) {
            alert("E-mobile打开表单链接，返回或提交后触发此回调函数");
        }
    });
});
```

第三种方式：window.open或者window.location.href跳转。此方式url需要传参returnUrl并转码，用于指定返回/提交后跳转地址。建议如果是EM客户端采用方式一或方式二！

```
window.open(viewUrl+"&returnUrl="+window.encodeURIComponent("/test.jsp?param1=test11&param2=test22"));
```

2.注册自定义事件

2.1 注册拦截事件，指定动作执行前触发，并可阻断/放行后续操作

- 支持多次注册，按注册顺序依次执行；支持异步ajax，避免请求卡住
- 场景1：表单提交、保存、退回、转发、强制收回等操作执行前，执行自定义逻辑并阻断/放行后续操作
- 场景2：明细添加行、删除行前，执行自定义逻辑并阻断/允许后续操作

动作类型	说明	最低版本要求
WfForm.OPER_SAVE	保存	
WfForm.OPER_SUBMIT	提交/批准/提交需反馈/不需反馈等	
WfForm.OPER_SUBMITCONFIRM	提交至确认页面，如果是确认界面，点确认触发的是SUBMIT	
WfForm.OPER_REJECT	退回	
WfForm.OPER_REMARK	批注提交	
WfForm.OPER_INTERVENE	干预	
WfForm.OPER_FORWARD	转发	
WfForm.OPER_TAKEBACK	强制收回	
WfForm.OPER_DELETE	删除	
WfForm.OPER_ADDROW	添加明细行，需拼明细表序号	
WfForm.OPER_DELROW	删除明细行，需拼明细表序号	
WfForm.OPER_PRINTPREVIEW	打印预览	KB900190501

WfForm.OPER_EDITDETAILROW	移动端-编辑明细	KB900191101
WfForm.OPER_BEFOREVERIFY	校验必填前触发事件	KB900191201

接口名称及参数说明

registerCheckEvent:function(type,fun)

参数	参数类型	必须	说明
type	String	是	动作类型(详见上表)，多个逗号分隔
fun	Function	是	自定义函数，此函数入参为callback，执行自定义逻辑完成或异步ajax的success函数体内，放行需调用callback，不调用代表阻断后续操作

样例

```
jQuery().ready(function(){
    WfForm.registerCheckEvent(WfForm.OPER_SAVE, function(callback){
        jQuery("#field27495").val("保存自动赋值");
        callback(); //继续提交需调用callback，不调用代表阻断
    });

    WfForm.registerCheckEvent(WfForm.OPER_SAVE+", "+WfForm.OPER_SUBMIT,function(callback){
        //... 执行自定义逻辑
        callback();
    });

    WfForm.registerCheckEvent(WfForm.OPER_ADDROW+"1", function(callback){
        alert("添加明细1前执行逻辑，明细1则是OPER_ADDROW+1，依次类推")
        callback(); //允许继续添加行调用callback，不调用代表阻断添加
    });

    WfForm.registerCheckEvent(WfForm.OPER_DELROW+"2", function(callback){
        alert("删除明细2前执行逻辑")
        callback(); //允许继续删除行调用callback，不调用代表阻断删除
    });

    WfForm.registerCheckEvent(WfForm.OPER_PRINTPREVIEW, function(callback){
        alert("控制默认弹出的打印预览窗口")
        alert("当打印含签字意见列表，此接口需要放到跳转路由前执行，组件库提供此机制");
        window.WfForm.printTimeout = 3000; //产品是默认延时1s自动弹出，可通过此方式控制延时时间
        callback(); //允许继续弹出调用callback，不调用代表不自动弹预览
    });

    WfForm.registerCheckEvent(WfForm.OPER_EDITDETAILROW,
function(callback,params){
    alert(JSON.stringify(params)); //参数含当前点击哪个明细表哪一行
    callback(); //允许跳转明细编辑窗口，不调用阻断跳转
});
});
```

2.2 注册钩子事件，指定动作完成后触发

支持多次调用注册，按注册的先后顺序依次执行

类型	说明	最低版本要求
WfForm.ACTION_ADDROW	添加明细行，需拼明细表序号	KB900190407
WfForm.ACTION_DELROW	删除明细行，需拼明细表序号	KB900190407
WfForm.ACTION_EDITDETAILROW	移动端-编辑明细行，需拼明细表序号	KB900190501
WfForm.ACTION_SWITCHDETAILPAGING	切换明细分页	KB900191201
WfForm.ACTION_SWITCHTABLAYOUT	切换模板布局标签页	KB900191201

接口名称及参数说明

registerAction: function(actionname, fn)

参数	参数类型	必须	说明
actionname	String	是	动作类型，详见上表
fn	Function	是	触发事件

样例

```
WfForm.registerAction(WfForm.ACTION_ADDROW+"1", function(index){
    alert("添加行下标是"+index);
}); //下标从1开始，明细1添加行触发事件，注册函数入参为新添加行下标

WfForm.registerAction(WfForm.ACTION_DELROW+"2", function(arg){
    alert("删除行下标集合是"+arg.join(","));
}); //下标从1开始，明细2删除行触发事件

WfForm.registerAction(WfForm.ACTION_SWITCHDETAILPAGING, function(groupid){
    alert("切换明细表"+(groupid+1)+"的页码触发事件");
});

WfForm.registerAction(WfForm.ACTION_SWITCHTABLAYOUT, function(tabid){
    alert("切换到标签项"+tabid+"触发事件");
});
```

3. 字段基础操作接口(适用表单所有字段类型)

3.1 将字段名称转换成字段id

灵活运用此方法，可实现多表单、多环境，代码块通用；解耦代码块中指定fieldid

convertFieldNameToId: function(fieldname,symbol,prefix)

参数说明

参数	参数类型	必须	说明
fieldname	String	是	字段名称
symbol	String	否	表单标示，主表(main)/具体明细表(detail_1),默认为main
prefix	Boolean	否	返回值是否需要 field 字符串前缀，默认为true

样例

```
var fieldid = wfForm.convertFieldNameToId("zs");
var fieldid = wfForm.convertFieldNameToId("zs_mx", "detail_1");
var fieldid = wfForm.convertFieldNameToId("zs_mx", "detail_1", false);
```

3.2 获取单个字段值

getFieldValue: function(fieldMark)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}

样例

```
var fieldvalue = wfForm.getFieldValue("field110");
```

3.3 修改单个字段值

- 此方法修改的字段如果涉及到触发联动、单元格格式化等，修改完值会自动触发联动/格式化
- 改值的格式在添加明细行初始化、批量修改字段等场景类同
- changeFieldValue: function(fieldMark, valueInfo)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
valueInfo	JSON	是	字段值信息，非浏览按钮字段格式为{value:"修改的值"};specialobj为浏览按钮信息，数组格式;showhtml属性只在单行文本类型且只读情况下生效；

样例

```
//修改文本框、多行文本、选择框等字段类型
wfForm.changeFieldValue("field123", {value:"1.234"});

//修改浏览框字段的值，必须有specialobj数组结构对象
```

```
WfForm.changeFieldValue("field11_2", {
    value: "2,3",
    specialobj:[
        {id:"2",name:"张三"},
        {id:"3",name:"李四"}
    ]
});

//修改check框字段(0不勾选、1勾选)
WfForm.changeFieldValue("field123", {value:"1"});

//针对单行文本框字段类型，只读情况，支持显示值跟入库值不一致
WfForm.changeFieldValue("field123", {
    value: "入库真实值",
    specialobj: {
        showhtml: "界面显示值"
    }
});
```

3.4 改变单个字段显示属性(只读/必填等)

changeFieldAttr: function(fieldMark, viewAttr)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
viewAttr	int	是	改变字段的状态，1：只读，2：可编辑，3：必填，4：隐藏字段标签及内容

样例

```
WfForm.changeFieldAttr("field110", 1); //字段修改为只读

WfForm.changeFieldAttr("field110", 4); //字段标签以及内容都隐藏，效果与显示属性联动隐藏一致，只支持主表字段
```

3.5 同时修改字段的值及显示属性

changeSingleField: function(fieldMark, valueInfo, variableInfo)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
valueInfo	JSON	否	字段值信息，与接口2格式一致，例：{value:"修改的值"}
variableInfo	JSON	否	变更属性，例：{viewAttr:3}

样例

```
wfForm.changeSingleField("field110", {value:"修改的值"}, {viewAttr:"1"}); //修改值同时置为只读
```

3.6 批量修改字段值或显示属性

changeMoreField: function(changeDatas, changeVariable)

参数说明

参数	参数类型	必须	说明
changeMoreField	JSON	是	修改的字段值信息集合
changeVariable	JSON	否	修改的字段显示属性集合

样例

```
wfForm.changeMoreField({
  field110:{value:"修改后的值"},
  field111:{value:"2,3",specialobj:[
    {id:"2",name:"张三"},{id:"3",name:"李四"}
  ]},
  ...
},{
  field110:{viewAttr:2},
  field111:{viewAttr:3},
  ...
});
```

3.7 触发指定字段涉及的所有联动

说明：手动触发一次字段涉及的所有联动，包括字段联动、SQL联动、日期时间计算、字段赋值、公式、行列规则、显示属性联动、选择框联动、bindPropertyChange事件绑定等

场景：触发出的子流程打开默认不执行字段联动、归档节点查看表单不执行联动，可通过此接口实现

triggerFieldAllLinkage:function(fieldMark)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}

样例

```
//表单打开强制执行某字段的联动
jQuery(document).ready(function(){
  wfForm.triggerFieldAllLinkage("field110"); //执行字段涉及的所有联动
});
```

3.8 根据字段ID获取字段信息

说明：根据字段ID获取字段信息，JSON格式，包括名称、类型、只读必填属性等

getFieldInfo:function(fieldid)

参数说明

参数	参数类型	必须	说明
fieldid	String	是	字段ID，不带任何标示

返回值字段信息JSON重要属性说明

参数	说明
htmltype	字段大类型(文本/多行文本...)
detailtype	字段小类型(整数/浮点数...)
fieldname	字段数据库名称
fieldlabel	字段显示名
viewattr	字段属性(1:只读；2：可编辑；3：必填)

```
wfForm.getFieldInfo("111");
```

3.9 获取字段当前的只读/必填属性

此方法为实时获取字段显示属性，包含显示属性联动、代码接口变更、已办、明细已有字段不可修改等可能的变更情况，不是仅仅获取后台配置的字段属性；

如只想获取后台配置的字段属性，调用接口3.8取返回值viewattr属性

getFieldCurViewAttr:function(fieldMark)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}

```
wfForm.getFieldCurViewAttr("field110_2"); //获取明细字段属性，1：只读、2：可编辑、3：必填；已办全部为只读；
```

4.表单字段事件绑定、自定义渲染

4.1 表单字段值变化触发事件

字段值变化即会触发所绑定的函数，可多次绑定

bindFieldChangeEvent: function(fieldMarkStr,funobj)

参数说明

参数	参数类型	必须	说明
fieldMarkStr	String	是	绑定字段标示,可多个拼接逗号隔开,例如:field110(主字段),field111_2(明细字段).....
funobj	Function	是	字段值变化触发的自定义函数,函数默认传递以下三个参数,参数1:触发字段的DOM对象,参数2:触发字段的标示(field27555等),参数3:修改后的值

```
wfForm.bindFieldChangeEvent("field27555,field27556", function(obj,id,value){
    console.log("wfForm.bindFieldChangeEvent--",obj,id,value);
});
```

4.2 明细字段值变化触发事件

绑定后对新添加的明细行字段以及加载的已有行明细字段,值变更触发所绑定的事件

bindDetailFieldChangeEvent: function(fieldMarkStr,funobj)

参数说明

参数	参数类型	必须	说明
fieldMarkStr	String	是	绑定的明细字段标示,不能有下列线标示,可多个拼接逗号隔开,例如:field110,field111
funobj	Function	是	字段值变更触发自定义函数,函数默认传递以下三个参数,参数1:字段标示(field27583),参数2:行标示,参数3:修改后的值

```
jQuery(document).ready(function(){

wfForm.bindDetailFieldChangeEvent("field27583,field27584",function(id,rowIndex,value){
    console.log("wfForm.bindDetailFieldChangeEvent--",id,rowIndex,value);
});
});
```

4.3 字段区域绑定动作事件

推荐使用值变化事件实现开发,因为此接口点击、双击等动作不是绑定到字段元素,是字段所在单元格区域即会触发

此接口所有功能都通可以新版公式实现

类型	说明
onblur	获取焦点事件，仅支持单行文本字段类型
onfocus	失去焦点事件，仅支持单行文本类型
onclick	单击事件，字段所在单元格区域单击触发
ondblclick	双击事件，字段所在单元格区域双击触发
mouseover	鼠标移入事件，鼠标移入字段所在单元格区域触发
mouseout	鼠标移出事件，鼠标移出字段所在单元格区域触发

bindFieldAction: function(type, fieldids, fn)

参数说明

参数	参数类型	必须	说明
type	String	是	动作类型，见上表
fieldids	String	是	字段id集合，多个逗号分隔，明细字段不加下划线对所有行生效
fn	Function	是	触发函数，此函数入参接收两个参数，fieldid以及rowIndex行号

样例

```
WfForm.bindFieldAction("onfocus", "field111,field222",
function(fieldid,rowIndex){
    alert("单行文本字段111获取焦点触发事件");
    alert("明细第"+rowIndex+"行字段222获取焦点触发事件");
});

WfForm.bindFieldAction("onclick", "field333", function(){
    alert("浏览按钮字段单击触发事件，不是指点放大镜选择，是整个字段所在单元格区域单击都会触发");
});
```

4.4 自定义代理渲染单行文本框字段

最低版本要求：KB900190407

此接口仅对单行文本框字段类型生效，即数据库字段类型为varchar

显示效果、事件、字段值交互都可自行控制，通过接口3.3修改的可编辑字段值也会正常入库

此接口传入的组件，产品会传入此字段依赖的相关props，具体可通过React Developer Tools自行抓取，按需调用

proxyFieldComp: function(fieldMark, el, range)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
el	React Comp	是	渲染的组件
range	String	否	作用范围，默认全部，(1:只读、2:可编辑、3:必填),组合后逗号分隔

样例

```
wfForm.proxyFieldComp("field111", React.createElement("div",{
  style:{background:"red"},
  children:"子内容"
}));
//字段111在只读、可编辑、必填情况下自定义渲染

wfForm.proxyFieldComp("field222_1", "<div>自定义渲染字段</div>", "2,3");
//明细某行字段222再可编辑、必填情况下自定义渲染
```

4.5 自定义追加渲染表单字段

- 最低版本要求：KB900190407
- 在标准字段展现内容的基础上，after方式追加渲染自定义组件
- 此接口参数说明与用法，与接口4.4类同
- afterFieldComp: function(fieldMark, el, range)

样例

```
wfForm.afterFieldComp("field111",
React.createElement("a",{
  href:"/test.jsp?userid="+wfForm.getFieldValue("field222"),
  children:"自定义链接"
}));
//字段111在只读、可编辑、必填情况下，追加渲染个自定义链接，链接参数依赖表单其它字段值
```

4.6 函数式自定义渲染表单字段

- 最低版本要求：KB900190701
- 以函数返回值方式自定义渲染表单字段，支持全部的字段类型，可实现基于原组件追加/复写/重新布局等等
- 建议结合ecode工具，放到模块加载前调用，使用JSX，可实现与表单字段渲染有关的二次开发
- 此接口的优先级高于4.4、4.5，即使用此接口代理的字段，如再使用4.4、4.5会直接无效
- proxyFieldContentComp: function(fieldid,fn)

接口参数说明

参数	参数类型	必须	说明
field	String	是	主表/明细表字段ID,格式 <code>\$fieldid\$</code>
fn	Function	是	代理的函数，此函数必须有返回值，返回自定义渲染的组件

代理的函数参数说明

参数	参数类型	必须	说明
info	JSON	是	字段基础信息，包括字段值、字段属性等等
compFn	Function	是	代理前原字段组件函数，可通过此函数获取原组件

样例

```
wfForm.proxyFieldContentComp("111", function(info, compFn){
  console.log("字段id: ",info.fieldid);
  console.log("明细行号: ",info.rowIndex);
  console.log("字段只读必填属性: ",info.viewAttr);
  console.log("字段值: ",info.fieldValue);
  //返回自定义渲染的组件
  return React.createElement("div", {}, ["想怎么玩就怎么玩"]);
  //返回原组件
  return compFn();
  //返回基于原组件的复写组件
  return React.createElement("div", {}, ["前置组件",compFn(),"后置组件"]);
});
//如果此接口调用在代码块、custompage等(非模块加载前调用)，需强制渲染字段一次
wfForm.forceRenderField("field111");
```

4.7 根据字段标识获取字段组件

最低版本要求：KB900190701

根据字段标识，获取字段组件，即字段组件可单独提取出来放在任意地方渲染。注意字段的只读可编辑属性仍需后台事先设置好

建议结合ecode工具，使用JSX，再结合设计器自定义属性或接口4.6，可实现某一区域自定义排版布局渲染多个表单字段

`generateFieldContentComp:function(fieldMark)`

接口参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 <code>field\${字段ID}_\${明细行号}</code>

样例

```
//以明细多字段、子明细的需求为例子
步骤1：模板单元格给自定义class: area_1
步骤2：自定义排版渲染area_1区域
```

```
ReactDOM.render(<div>
  <table>
    <tr>
      <td>{wfForm.generateFieldContentComp("field111_1")}</td>
      <td>{wfForm.generateFieldContentComp("field112_1")}</td>
    </tr>
    <tr>
      <td>{wfForm.generateFieldContentComp("field113_1")}</td>
      <td>{wfForm.generateFieldContentComp("field114_1")}</td>
    </tr>
  </table>
</div>, document.getElementByIdByClassName("area_1")[0]);    //仅供参考，参数二要区分行号
                        定位到具体单元格
```

5.明细表操作相关接口

5.1 添加明细行并设置初始值

```
addDetailRow: function(detailMark, initAddRowData={})
```

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推
initAddRowData	JSON	否	给新增后设置初始值，格式为{field110: {value:"11"},field112:{value:"22"},...},注意key不带下划线标示

样例

```
//明细2添加一行并给新添加的行字段field111赋值
wfForm.addDetailRow("detail_2",{field111:{value:"初始值"}});

//动态字段赋值，明细1添加一行并给字段名称为begindate的字段赋值
var begindatefield = wfForm.convertFieldNameToId("begindate", "detail_1");
var addObj = {};
addObj[begindatefield] = {value:"2019-03-01"};
wfForm.addDetailRow("detail_1", addObj);

//不推荐这种动态键值写法，IE不支持，避免掉
wfForm.addDetailRow("detail_1", {[begindatefield]:{value:"2019-03-01"}})
```

5.2 删除明细表指定行/全部行

```
delDetailRow: function(detailMark, rowIndexMark)
```

说明：此方法会清空明细已选情况，删除时没有提示"是否删除"的确认框

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推
rowIndexMark	String	是	需要删除的行标示，删除全部行:all,删除部分行："1，2，3"

样例

```
WfForm.delDetailRow("detail_1", "all"); //删除明细1所有行
WfForm.delDetailRow("detail_1", "3,6"); //删除明细1行标为3,6的行
```

5.3 选中明细指定行/全部行

checkDetailRow: function(detailMark, rowIndexMark,needClearBeforeChecked)

说明：此方法可灵活使用，依靠参数needClearBeforeChecked可实现清除选中的逻辑

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推
rowIndexMark	String	否	需要选中的行标示，选中全部行:all,选中部分行："1，2，3"
needClearBeforeChecked	Boolean	否	是否需要清除已选

样例

```
WfForm.checkDetailRow("detail_2", "all"); //勾选明细2所有行
WfForm.checkDetailRow("detail_2", "", true); //清除明细2所有已选
WfForm.checkDetailRow("detail_2", "3,6", true); //清除明细2全部已选，再勾选行标为3,6的行
WfForm.checkDetailRow("detail_2", "7", false);
//保持已选记录，追加选中行标为7的行
```

5.4 获取明细行所有行标示

getDetailAllRowIndexStr : function(detailMark)

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推

样例

```
console.log(WfForm.getDetailAllRowIndexStr("detail_2")); //输出1，3...等等
```

5.5 获取明细选中行下标

- 最低版本要求：KB900190501
- getDetailCheckedRowIndexStr：function(detailMark)

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推

样例

```
console.log(WfForm.getDetailCheckedRowIndexStr("detail_2"));    //输出选中行1, 3...等等
```

5.6 控制明细行check框是否禁用勾选

- 注：后台配置的置灰行(不允许删除情况)，不支持通过此API控制
- controlDetailRowDisableCheck: function(detailMark, rowIndexMark, disableCheck)

参数说明

参数	参数类型	必须	说明
detailMark	String	true	明细表标示，明细1就是detail_1，以此递增类推
rowIndexMark	String	是	需要选中的行标示，选中全部行:all,选中部分行："1，2，3"
disableCheck	boolean	true	是否禁用勾选，true:置灰禁止勾选，false:允许勾选

```
WfForm.controlDetailRowDisableCheck("detail_1", "all", true);    //明细1所有行check框置灰禁止选中

WfForm.controlDetailRowDisableCheck("detail_1", "1,2", false);
//明细1行标为1,2的行释放置灰，允许勾选
```

5.7 控制明细数据行的显示及隐藏

- 注：只是界面效果隐藏，序号不会变化，即被隐藏行的前后行序号会断层不连续
- controlDetailRowDisplay: function(detailMark, rowIndexMark, needHide)

参数说明

参数	参数类型	必须	说明
detailMark	String	true	明细表标示，明细1就是detail_1，以此递增类推
rowIndexMark	String	是	需要选中的行标示，选中全部行:all,选中部分行："1，2，3"
needHide	boolean	true	是否隐藏行，true:隐藏，false:显示

```
WfForm.controlDetailRowDisplay("detail_1", "3,5", true); //明细1行标为3,5的隐藏不显示
```

```
WfForm.controlDetailRowDisplay("detail_1", "all", false);  
//明细1所有行不隐藏都显示
```

5.8 获取明细已有行的数据库主键

```
getDetailRowKey: function(fieldMark)
```

此方法只对明细已有行生效，新增加的行/不存在的行返回-1

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 <code>field\${字段ID}_\${明细行号}</code> ，用于定位属于哪个明细表

```
WfForm.getDetailRowKey("field112_3"); //获取明细第四行主键
```

5.9 获取明细总行数

```
getDetailRowCount : function(detailMark)
```

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推

样例

```
console.log(WfForm.getDetailRowCount("detail_2")); //输出明细总行数，注意此结果只代表  
明细总行数，不能用作循环行
```

5.10 添加行、删除行前执行逻辑或阻断事件

最低版本要求：KB900190501

场景：可实现添加行前执行自定义逻辑、限制超过多少行添加无效、不允许删除等

采用注册函数机制，详见接口2.1

5.11 添加行、删除行后触发事件

采用钩子机制，详见接口2.2

5.12 移动端跳转至明细编辑行页面执行事件

最低版本要求：KB900190501

仅应用于移动端编辑明细行，采用钩子机制，详见接口2.2

5.13 添加明细时默认复制最后一行记录

setDetailAddUseCopy: function(detailMark, needCopy)

说明：此方法在ready时调用，手动点添加时自动赋值最后行字段内容，覆盖节点前设置及默认值等，附件上传字段不予复制

注：E9是异步ready执行后点添加明细才生效，例如默认新增空明细无效;

参数说明

参数	参数类型	必须	说明
detailMark	String	是	明细表标示，明细1就是detail_1，以此递增类推
needCopy	Boolean	是	是否需要启用复制，true：启用，false：不启用

```
jQuery(document).ready(function(){
    wfForm.setDetailAddUseCopy("detail_1", true);
});
```

5.14 根据明细行标识获取序号(第几行)

最低版本要求：KB900190601

场景：根据明细行的下标，获取当前是第几行明细，可用于提示某某行异常等

getDetailRowSerailNum: function(mark, rowIndex)

参数说明

参数	参数类型	必须	说明
mark	String	是	明细表标示，支持两种格式 detail_\${dindex} 或者 field\${fieldid}_\${rowIndex}
rowIndex	Int	是	行标识，第一种格式 detail_\${dindex} 才需要传此参数

样例

```
wfForm.getDetailRowSerailNum("detail_1", 3); //获取明细1下标为3的行序号
wfForm.getDetailRowSerailNum("field222_3"); //获取字段222对应明细表下标为3的行序号
```

6.常用全局接口(与字段无关)

6.1 获取当前打开请求的基础信息

说明：包括路径id、节点id、表单id、主次账号信息

getBaseInfo: function()

```
console.log(wfForm.getBaseInfo());    //返回当前请求基础信息

//输出对象说明:
{
  f_weaver_belongto_userid: "5240"    //用户信息
  f_weaver_belongto_usertype: "0"
  formid: -2010                        //表单id
  isbill: "1"                          //新表单/老表单
  nodeid: 19275                        //节点id
  requestid: 4487931                  //请求id
  workflowid: 16084                   //路径id
}
```

6.2 可控制显示时间的message信息

showMessage: function(msg, type, duration)

参数说明

参数	参数类型	必须	说明
msg	String	true	提示信息内容
type	int	false	提示类型，1(警告)、2(错误)、3(成功)、4(一般)，默认为1，不同类型提示信息效果不同
duration	Float	false	多长时间自动消失，单位秒，默认为1.5秒

```
wfForm.showMessage("结束时间需大于开始时间");    //警告信息，1.5s后自动消失

wfForm.showMessage("运算错误", 2, 10);    //错误信息，10s后消失
```

6.3 系统样式的Confirm确认框

说明：兼容移动端，可自定义确认内容及按钮名称

showConfirm: function(content, okEvent, cancelEvent, otherInfo={})

参数说明

参数	参数类型	必须	说明
content	String	是	确认信息
okEvent	Function	是	点击确认事件
cancelEvent	Function	否	点击取消事件
otherInfo	Object	否	自定义信息(按钮名称)

```

WfForm.showConfirm("确认删除吗？", function(){
    alert("删除成功");
});

WfForm.showConfirm("请问你是否需要技术协助？",function(){
    alert("点击确认调用的事件");
},function(){
    alert("点击取消调用的事件");
},{
    title:"信息确认",          //弹确认框的title, 仅PC端有效
    okText:"需要",             //自定义确认按钮名称
    cancelText:"不需要"        //自定义取消按钮名称
});

```

6.4 表单顶部按钮、右键菜单置灰

说明：设置表单顶部按钮、右键菜单置灰不可操作和恢复操作功能参数isDisabled

controlBtnDisabled: function(isDisabled)

参数说明

参数	参数类型	必须	说明
isDisabled	boolean	是	true：按钮全部置灰不可操作,false：恢复按钮可操作状态

```

function subimtForm(params){
    wfForm.controlBtnDisabled(true);    //操作按钮置灰
    ...
    wfForm.controlBtnDisabled(false);
}

```

6.5 调用右键按钮事件

说明：调用表单右键事件逻辑，只可调用，不允许复写

doRightBtnEvent: function(type)

参数说明

参数	参数类型	必须	说明
type	String	是	按钮类型(控制台可通过 mobx.toJS(WfForm.getGlobalStore().rightMenu.rightMenus)方式定位具体按钮type)

```
WfForm.doRightBtnEvent("BTN_SUBBACKNAME"); //触发提交需反馈
WfForm.doRightBtnEvent("BTN_SUBMIT"); //触发提交不需反馈
WfForm.doRightBtnEvent("BTN_WFSAVE"); //触发保存
WfForm.doRightBtnEvent("BTN_REJECTNAME"); //触发退回
WfForm.doRightBtnEvent("BTN_FORWARD"); //触发转发
WfForm.doRightBtnEvent("BTN_REMARKADVICE"); //触发意见征询
WfForm.doRightBtnEvent("BTN_TURNTODO"); //触发转办
WfForm.doRightBtnEvent("BTN_DORETRACT"); //触发强制收回
WfForm.doRightBtnEvent("BTN_PRINT"); //触发打印
```

6.6 刷新表单页面

强制刷新表单页面，默认为当前requestid

```
reloadPage: function(params={})
```

参数说明

参数	参数类型	必须	说明
params	Object	否	自定义参数，覆盖默认参数

```
WfForm.reloadPage();

WfForm.reloadPage({requestid:"11"}); //覆盖参数
```

6.7 移动端打开链接方式

仅支持移动端，特别是非表单主界面(例如：明细编辑)需要用此方式打开链接。此方式打开链接返回不会刷新表单

```
window.showHoverWindow : function(url,baseRoute)
```

参数说明

参数	参数类型	必须	说明
url	String	是	打开的链接地址
baseRoute	String	是	当前路由地址，具体见url地址，明细编辑打开传'/req/editDetailRow'

样例

```
window.showHoverWindow('/workflow/test.jsp', '/req'); //主界面打开链接
window.showHoverWindow('https://www.baidu.com', '/req/editDetailRow'); //明细行
编辑界面打开链接
```

6.8 扩展提交操作发送给服务端的参数

最低版本要求：KB900190801

自定义扩展提交/保存动作发送给服务端的参数，服务端可通过request.getParameter方式取到对应参数值

推荐：扩展的自定义参数都以cus_开头，避免影响/覆盖标准产品所必需的参数，导致功能异常

```
appendSubmitParam: function(obj={})
```

参数说明

参数	参数类型	必须	说明
obj	Object	否	JSON格式自定义参数

```
WfForm.appendSubmitParam({cus_param1:"11", cus_param2:"测试"}); //服务端可通过  
request对象取到参数值request.getParameter("cus_param1");
```

6.9 获取校验必填逻辑第一个未必填的字段

最低版本要求：KB900191201

场景：调用产品的必填校验逻辑，获取第一个未必填字段；例如结合接口2.1中
WfForm.OPER_BEOPER_BEFOREVERIFY可实现自定义控制必填提示的效果

```
getFirstRequiredEmptyField: function()
```

```
var emptyField = WfForm.getFirstRequiredEmptyField(); //获取调用时刻的第一个未必填字  
段，返回值格式是`field${fieldid}_${rowIndex}`
```

6.10 触发一次必填验证

最低版本要求：KB900191201

手动触发一次必填验证并提示，可选控制校验必须新增空明细/校验字段必填

```
verifyFormRequired: function(mustAddDetail=true, fieldRequired=true)
```

参数说明

参数	参数类型	必须	说明
mustAddDetail	Boolean	否	是否校验必须新增空明细，默认是
fieldRequired	Boolean	否	是否校验字段必填，默认是

```
WfForm.verifyFormRequired(); //触发必填验证并提示，先校验必须新增空明细，后校验字段必填  
  
var result = WfForm.verifyFormRequired(false, true); //仅校验字段必填并提示  
if(!result)  
    alert('存在未必填情况');
```

7.不同字段类型特定接口(限定指定字段类型可用)

7.1 扩展浏览按钮取数接口参数值

限定条件：仅适用非日期时间的浏览按钮类型

场景：控制浏览按钮可选数据范围，限定范围、依赖表单字段过滤数据范围等；对联想输入范围
及弹窗选择范围都生效；

实现方式：接口扩充的参数会通过url参数提交到服务端接口，需结合修改浏览按钮接口类方可生效

appendBrowserDataUrlParam: function(fieldMark, jsonParam)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
jsonParam	JSON	是	扩展的url参数，JSON中key-value格式

```
wfForm.appendBrowserDataUrlParam("field395",{"customerid":"2"}); //给浏览按钮395请求后台数据时增加url参数customerid
```

7.2 获取浏览按钮的显示值

限定条件：仅适用非日期时间的浏览按钮类型

获取浏览按钮的显示名称，多个则以splitChar字符分隔拼接成串

getBrowserShowName: function(fieldMark, splitChar)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
splitChar	String	否	分隔符，默认以逗号分隔

```
wfForm.getBrowserShowName("field110"); //以逗号分隔获取浏览按钮字段显示值
```

7.3 移除选择框字段选项

限定条件：仅适用选择框类型字段

removeSelectOption: function(fieldMark, optionKeys)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
optionKeys	String	是	需要移除的Option选项key值，多个以逗号分隔

```
wfForm.removeSelectOption("field112", "3,4"); //移除选择框中id值为3/4的选项
```

7.4 控制选择框字段选项

限定条件：仅适用选择框类型字段

controlSelectOption: function(fieldMark, optionKeys)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
optionKeys	String	是	完全控制选择框的选项范围

```
wfForm.controlSelectOption("field112", "1,2,4"); //控制选择框只显示1/2/4的选项
wfForm.controlSelectOption("field112", ""); //清除选择框所有选项
```

7.5 获取选择框字段的显示值

- 限定条件：仅适用选择框类型字段
- 获取选择框类型的显示名称，多个则以splitChar字符分隔拼接成串
- getSelectShowName:function(fieldMark,splitChar)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
splitChar	String	否	分隔符，默认以逗号分隔(只有复选框多选才会用到)

```
wfForm.getSelectShowName("field110_3"); //获取选择框字段显示值
```

7.6 文本字段可编辑状态，当值为空显示默认灰色提示信息，鼠标点击输入时提示消失

- 限定条件：仅支持单行文本、整数、浮点数、千分位、多行文本字段(非html)字段类型；支持主字段及明细字段
- setTextFieldEmptyShowContent:function(fieldMark,showContent)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
showContent	String	是	空值时显示的提示信息，灰色

```
jQuery(document).ready(function(){
    wfForm.setTextFieldEmptyShowContent("field27555", "单文本默认提示信息1");
    wfForm.setTextFieldEmptyShowContent("field27566", "多文本默认提示2");
    wfForm.setTextFieldEmptyShowContent("field222_0", "明细字段提示信息"); //需要结合接口5.9添加行事件一并使用
});
```

7.7 复写浏览按钮组件的props

- 仅支持浏览按钮类型，谨慎使用，完全重写覆盖浏览按钮的props
- 只是传递props，具体传递的属性实现何种需求由组件内部控制
- overrideBrowserProp: function(fieldMark,jsonParam)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
jsonParam	JSON	是	扩展或复写的props参数

样例

```
WfForm.overrideBrowserProp("field111",{
  onBeforeFocusCheck: function(success, fail){/***/}
}); //复写浏览按钮字段111的props
```

7.8 控制日期浏览按钮的可选日期范围

- 最低版本要求：KB900190501
- 仅支持日期类型，控制手动选择时的可选日期范围
- controlDateRange: function(fieldMark,start,end)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
start	String	是	支持两种格式，第一种标准的日期格式，比如2019-05-28，第二种整数，相比于当前日期往前/后多少天
end	String	否	格式与start参数一致

样例

```
WfForm.controlDateRange("field111", -5, 10); //限定日期可选范围，往前5天，往后10天

WfForm.controlDateRange("field111", 0, '2019-12-31'); //限定今天至本年

WfForm.controlDateRange("field222_0", '2019-05-01', '2019-05-31'); //明细字段，限定当月
```

7.9 控制Radio框字段打印是否仅显示选中项文字

- 最低版本要求：KB900190501
- 仅支持选择框中单选框类型，打印场景，是否仅显示选中项文字，都未选中显示空
- controlRadioPrintText: function(fieldid)

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段id，格式 <code>\${字段ID}</code> ，支持明细字段

样例

```
WfForm.controlRadioPrintText("12580"); //单选框字段12580打印只显示文字
```

8. 签字意见接口

8.1 获取签字意见内容

最低版本要求：KB900190501

getSignRemark: function()

样例

```
WfForm.getSignRemark(); //获取签字意见内容
```

8.2 设置签字意见内容

最低版本要求：KB900190501

setSignRemark: function(text,isClear=true,isAfter=true,callback)

参数说明

参数	参数类型	必须	说明
text	String	是	需设置的内容
isClear	Boolean	否	是否先清除意见，默认为true，即覆盖意见内容，false为追加意见内容
isAfter	Boolean	否	追加的位置，默认为true，原意见内容尾部追加，false再头部追加
callback	Function	否	意见设置成功后的回调函数

样例

```
WfForm.setSignRemark("覆盖设置签字意见内容");  
  
WfForm.setSignRemark("原有意见内容前追加请审批", false, false);
```

8.3 扩展签字意见输入框底部按钮

签字意见输入框底部按钮，在产品内置的附件、文档、流程基础上，支持自定义扩充

appendSignEditorBottomBar: function(comps=[])

参数说明

参数	参数类型	必须	说明
comps	React Comp Array	是	需要扩充的React组件数组

```
WfForm.appendSignEditorBottomBar([
  React.createElement("div",{className:"wf-req-signbtn",children:"自定义按钮1"}),
  <div>自定义按钮2</div>
]);
```

9.历史E8代码块的相关兼容

以下方法不推荐使用，只为兼容历史代码块，上述WfForm中API都涵盖包括

建议全部改造通过WfForm接口实现，使用老版本接口、操作Dom结构等，很容易引发问题及后续版本升级不兼容

9.1 提交事件执行自定义函数

比如checkCustomize、checkCarSubmit会继续执行,根据函数返回值判断是否阻塞提交，返回值true：继续流转，false:阻断提交

开发示例

```
window.checkCustomize = ()=>{
  var flag = true;
  //...
  return flag;
}
```

9.2 字段值变化触发事件bindPropertyChange

说明：与E8一致，依赖DOM，字段值变化时触发此函数; 自定义函数默认传递以下三个参数，参数1：触发字段的DOM对象，参数2：触发字段的标示(field27563等)，参数3：修改后的值

样例

```
jQuery(document).ready(function(){
  jQuery("#field27563").bindPropertyChange(function(obj,id,value){
    console.log("tri...",obj,id,value);
  });
});
```

9.3 明细新增行渲染后触发事件

说明：重载_customAddFun"+groupid+"函数，groupid从0开始递增，0代表明细1；不论手动添加、联动添加、接口添加都会触发

注意：此方法内是可以获取到新增行的DOM结构，自定义函数默认传参addIndexStr为新增行的标示串；如联动一次性增加多行，此方法只触发一次，参数addIndexStr值为多行的行标示，逗号分隔；

样例

```
function _customAddFun1(addIndexStr){           //明细2新增成功后触发事件，addIndexStr即刚新增的行标示，添加多行为(1,2,3)
    console.log("新增的行标示: "+addIndexStr);
}
```

9.4 明细删除行渲染后触发事件

说明：重载_customDelFun"+groupid+"函数，groupid从0开始递增，0代表明细1；不论手动删除、接口删除都会触发

样例

```
function _customDelFun1(){ //明细2删除成功后触发事件
    console.log("删除明细");
}
```

9.5 修改浏览按钮字段值window._writeBackData

此方法只为兼容E8已有代码，新开发的请使用WfForm.changeFieldValue方式赋值

参数说明

参数	参数类型	必须	说明
fieldMark	String	是	字段标示，格式 field\${字段ID}_\${明细行号}
isMustInput	String	是	E8参数，E9暂时不用
data	JSON	是	修改的浏览按钮数据，格式{id:"",name:""}
_options	JSON	否	修改方式，格式{replace:false,isSingle:true}，replace或isSingle任意一个为true即为覆盖修改，否则为在原有浏览按钮数据基础上追加修改，默认为覆盖修改

```
_writeBackData("field110", 1, {id:"22",name:"lzy"},
{replace:false,issingle:false}); //多人力浏览按钮追加个值lzy
```

9.6 少用jQuery操作

不推荐使用,历史jQuery("#field111").val()取值、赋值等操作已兼容

9.7 禁止JS原生操作

不允许，会引发各种不可控异常

类似document.write("");document.getElementById("field111").value操作需要调整为WfForm接口操作

10 常用配置文件修改方式

以下配置修改都是针对当前ecology系统全部流程生效

10.1 修改意见默认字体

sysadmin账号登录，在浏览器调用接口的方式修改配置

/api/workflow/index/updateWfConfig?name=signinput_default_fontfamily&value=仿宋_GB2312/FangSong_GB2312

value参数(字体)范围

字体值	--
仿宋_GB2312/FangSong_GB2312	
宋体/SimSun	
微软雅黑/Microsoft YaHei	
楷体/KaiTi	
黑体/SimHei	
Arial/Arial, Helvetica, sans-serif	
新宋体/NSimSun	
楷体_GB2312/KaiTi_GB2312	

10.2 修改意见默认字体大小

sysadmin账号登录，在浏览器调用接口的方式修改配置

/api/workflow/index/updateWfConfig?name=signinput_default_fontsize&value=36/36px

value参数(字体大小)范围

字体大小	--	--	--
8/8px	9/9px	10/10px	11/11px
2/12px	14/14px	16/16px	18/18px
20/20px	22/22px	24/24px	26/26px
28/28px	36/36px		

10.3 流程自定义浏览框缓存功能开关

sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=un_use_customize_browser_cache&value=0

value参数1关闭，0开启

缓存任务参数及清理数据缓存页面（任务功能：保存表单、打开表单进行自定义浏览框数据缓存及刷新、以及清理自定义浏览框缓存数据批量清理以及单个流程清理）

/workflow/request/CustomizeBrowserCacheUtil.jsp

10.4 非多行文本html字段类型支持html格式

场景：E9限制只有当字段类型为多行文本且勾选html的字段，内容才支持html格式 单行文本字段、多行文本字段是不支持，但是部分场景此类字段值是通过外面接口等情况赋值html串 此情况可通过改配置实现

第一步：找到需要支持html格式的字段id(可通过表单设计器模板单元格选中看右下角) 假设字段id为12345，则格式为：field12345_1 (如是老表单，则格式为 field12345_0，老表单指字段从字段库选择生成的表单，一般是E8系统前) 第二步：执行SQL(不同数据库请转换拼接符)

```
update workflow_config set value=value||',field12345_1' where  
name='support_html_textarea_field'
```

第三步：重启resin生效

10.5 PC端-流程表单显示底部耗时信息开个(调试分析用)

最低版本要求：KB900190801 sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=show_duration_log&value=1

value参数1开启，0关闭，默认为关闭

10.6 明细开启横向滚动条情况下，首行(按钮所在行)固定不跟随滚动条滚动

最低版本要求：KB900191101 场景: 明细开启横向滚动条、首行不包含除按钮/文字外类型、首行含添加删除按钮且处于非列锁定区域 满足上述三条件，首行会固定，不跟随横向滚动条滚动，此时根据模板不同可能会存在首行与下一行存在不对齐问题 sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=detail_locked_button_row&value=1

value参数1锁定，0取消锁定，默认为锁定

10.7 明细字段合计给主字段，当明细未添加行，赋零值或空值

场景: 配置列合计给主字段，当明细没有添加过行，主表合计字段，有些场景需要赋值为空，用于校验必填 有些场景需要赋值为零，用于出口判断

sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=colRule_noRow_empty&value=1

value参数1赋零值，0赋空值，默认为赋空值

10.8 pc端-手写签批按钮开关

最低版本要求：KB900191001 sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=handwrittensign_switch&value=1

value参数1开启，0关闭

10.9 移动端-选择框单选框类型显示成radio效果开关

最低版本要求：KB900191101 sysadmin访问链接修改配置，实时生效

/api/workflow/index/updateWfConfig?name=mobile_show_radio&value=1

value参数1开启，0关闭，默认为关闭

10.10 移动端-表单正文、附件签批功能开关

最低版本要求：KB900190308 修改配置文件：MobileWFOfficeSign.properties

mobileWFOffice=1

mobileWFOffice参数1开启，0关闭，开启后显示签批按钮，否则不显示签批按钮

mobilePDFSign=1

mobilePDFSign 最低版本要求：KB900190901 mobilePDFSign 配置项只对pdf文件生效

mobilePDFSign 参数1 pdf文件签批后放在签字意见附件中 参数2 pdf文件签批后直接替换原文件

参数3 由用户选择签批后放在签字意见附件中还是替换原文件

11 常用CSS样式案例分享

根据客户需求范围，灵活变通 仅对当前模板生效--->写在代码块里面 仅对当前路径所有节点生效--->写在路径基础设置-自定义页面(CSS文件) 对系统所有路径生效--->写在应用设置-全局自定义页面(CSS文件)

注：如写在代码块中需包一层style标签，写在CSS文件中不需要

11.1 实现明细添加删除按钮靠左显示

场景：明细添加删除按钮，始终靠右显示，如何实现靠左显示

插入样式到代码块：

```
<style>
.detailButtonDiv{float:left}
</style>
```

如写到CSS文件则不需要style标签

```
.detailButtonDiv{float:left}
```

11.2 实现单元格图片居中、自适应缩放

场景：单元格插入图片，始终按原始尺寸从左上角平铺显示，如何居中，如何自适应缩放

最低版本要求：KB900190901

图片所在单元格增加自定义属性class标识：

```
imageCell
```

单元格宽度大于图片尺寸，实现图片居中，增加样式：

```
.imageCell_swap{background-position:center}
```

单元格宽度小于图片尺寸，实现图片自适应缩放显示完整，增加样式：

```
.imageCell_swap{background-size:100% 100%}
```

11.3 控制浏览按钮链接颜色

场景：浏览按钮链接的颜色不受单元格颜色控制，如何修改浏览按钮链接颜色

方案一：要求版本达到KB900190901以上

可通过登录sysadmin，访问以下链接开启配置，开启后浏览按钮链接颜色完全取单元格设置的颜色
`/api/workflow/index/updateWfConfig?name=browser_color_controlByCell&value=1` (value参数0关闭，1开启，默认为关闭)

方案二：不限制版本

浏览按钮所在单元格增加自定义属性class标识：

```
browserColorCell
```

增加样式

```
.browserColorCell a{color:red !important} //强制将浏览按钮链接显示为红色
```

11.4 控制主表选择框字段最小宽度

场景：主表选择框字段，最小宽度要求100px，当选项仅为"是/否"时，如何再减小宽度

选择框所在单元格增加自定义属性class标识：

```
selectCell
```

增加样式

```
.selectCell .wea-select{min-width:50px !important}
```

12 其它场景分享

12.1 移动端异构系统提交表单后，如何刷新流程列表

最低支持版本：KB900191101

```
//提交后跳转至此页面  
window.location.href = 当前ecology服务器地址  
+ "/workflow/workflow/WfRefreshList.jsp"
```

注意：移动端可能存在反向代理，请将当前ecology服务器地址配置成反向代理对应的地址