

模块开发准则

模块

模块开发接口

设计模板接口

- **Designer.getPages()**

获取当前系统所有页面信息

- **Designer.selectPage(callback)**

弹出页面列表给客户选择，返回选择的页面信息给客户

示例

```
Designer.selectPage(function(pageinfo) {  
    console.log(pageinfo.name);  
});
```

- **Designer.getRelatePage(object_name)**

获取跟当前对象关联的对象列表

TODO:

- 封装获取 controller 的方式
- 封装获取 option 的方式

数据模板接口

- **getContext(ele, tag)**

用来获取当前上下文

ele 是当前激活的元素，比如点击了一个按钮，通过 ele 就可以找到 ele 所在的模块，并得到对应模块的上下文，tag 可以用来区分同个模块的子上下文

上下文的操作

```
var ctx = getContext(this);
```

```
ctx.get("name", defaultValue) //获取保存在上下文中的 name 参数
```

```
ctx.set("name", value) // 设置对应的变量到当前上下文
```

```
ctx.erase(key) // 删除 key 对应的上下文变量
```

```
ctx.reset() // 清空从当前控件以及相关子控件的上下文
```

```
ctx.getContainer() // 获取当前控件所在的容器，也就是说，可以用来替换自己的 html
```

`ctx.responseType("json" + "html")` // 可以用来设置控件接收的是绑定的相关数据，或者是渲染好的 html，一般在初始化的时候设置 (此接口已作废)

此外，还有相关的全局定时器，不过暂时没有开放，可以把控件放进定时器，以达到自动更新的效果，比如自动更新页面上的统计数据。

具体的实现可以查看 findsth.js 里面的 STimer

`ctx.getOptions()` // 获取在设计界面保存的信息

```
ctx.configure(config): // 设置当前模块的一些配置信息"
```

示例

```
// 设置新窗口使用 modal 框弹出
ctx.configure({ "EIDT_WINDOW_TYPE": "modal" })
// 设置新窗口使用 slice 界面，从右边滑出
ctx.configure({ "EIDT_WINDOW_TYPE": "slice" })
// 默认情况下，模块在更新时会自动增加 loading 元素，通过以下接口的设计
// 可以取消默认的 loading，自己来控制 loading
ctx.configure({ "NO_LOADING": true })
```

当前对象以及关联对象 相关操作

```
ctx.operateObj(type, obj, callback)
```

type 是指要对当前对象执行的操作，有 add、edit、remove

obj 是指要操作的对象数据，如果是 add 则将 obj 加入数据库；

如果是 edit 则更新 obj 对应的数据；

如果是 remove 则是删除 obj 对应的数据

callback 则是完成操作后执行的回调

```
ctx.objectRel(objname, field, obj)
```

objname 是关联对象的名称，field 是关联的字段，obj 是当前对象的数据

- **update(ctx, tag)**

用来触发关系链上的交互行为

每个模块都定义了统一的行为，通过 update 可以自动的获取绑定在行为上的数据，并调用绑定的处理函数

- **toSubPage(pageid, url, args, target)**

让 target 来承载 pageid 跟 url 对应的页面，并传递相关的参数来初始化目标页面

- **activeTo(pageid)**

在 toSubPage 调用的函数，当页面上存在多个子页时，可以把当前页面的上下文激活到 pageid 对应的子页

- **Navigate(urlOrPid, neww, params)**

跳转到 urlOrPid 对应的页面去，而 neww 则决定了是打开新的窗口，还是直接从当前页导向过去。最后的 params 则是你从这个页面过去的时候，想要附加的参数。

- **Destroy(pid)**

当 Pid 为空时，摧毁当前页面

摧毁当前页面，默认会注册为：当浏览器关闭时，自动摧毁当前页面

这里的摧毁包括删除本地的缓存，上下文信息，然后发送信息告诉服务器，当前页面缓存在服务器上的信息也可以销毁（如关联的模块跟页面信息）

模板定义说明

在预览模板中，获取当前模块的控件信息

```
var control = window.get_control("{ context['control_identity'] }");
```