

OpalBinding - 控件

- 控件输入事件触发 vars更新

```
<%= input binding: :text, value:0 %>
```

JS

生成的前端HTML代码，每个widget生成唯一的id

HTML

```
<input type='text' value='0' id='text_ABCDEF'>
```

OpalBinding中保存的对应关系

```
binding = [ [:text, :text_ABCDEF, ...] ]  
vars = [:text => 0]
```

JS

```
$vars["text"] = $document["text_ABCDEF"].value
```

```
## closure proc object  
def on_change_event_proc(css_id,name)  
  proc {  
    fetch_change(css_id)  
    $dirty_vars.push(name)  
    trigger_on_change(name)  
  }  
end
```

```
$document["text_ABCDEF"].on("input",  
&on_change_proc("text_ABCDEF","text"))
```

OpalBinding - 控件

- 控件输入事件触发 vars更新

```
<%= input binding: :text, value:0 %>
```

JS

生成的前端HTML代码，每个widget生成唯一的id

HTML

```
<input type='text' value='0' id='text_ABCDEF'>
```

OpalBinding中保存的对应关系

```
binding = [ [:text, :text_ABCDEF, ...] ]  
vars = [:text => 0]
```

JS

```
$vars["text"] = $document["text_ABCDEF"].value
```

dirty_vars 是队列，有脏的变量（代表值发生变化需要处理），就放入队列

后续循环来处理脏的变量

```
## closure proc object  
def on_change_event_proc(css_id,name)  
  proc {  
    fetch_change(css_id)  
    $dirty_vars.push(name)  
    trigger_on_change(name)  
  }  
end
```

```
$document["text_ABCDEF"].on("input",  
&on_change_proc("text_ABCDEF","text"))
```