触发菜单事件，返回值中type为：ir\_actions\_act\_window会创建新的ViewManagerAction对象，返回值中还有一个字段是“views”，这个字段是一个列表类型，列表中的每一个值也是一个列表，这个子列表中包含两个值，第一个是view的id，第二个是view的类型，在ViewManagerAction初始化完成后会调用其父类ViewManager的init方法，init方法会解析字段，并把view的id和view的类型赋给变量views\_src；在随后的start方法也会用到这个views\_src变量，总之views\_src包含了当前页面需要显示的类型，并且这个类型在do\_create\_view方法中实例化成对象。也就是说假如服务端返回的views中视图类型为form，则在do\_create\_view方法中会实例化FormView。

* Model在data.js中有扩展
* 时区处理在chrome.js WebClient的check\_timezone方法中，用于检测浏览器时区
* ViewManagerAction 构造函数中的第二个参数action就是从服务端返回的json串
* Widget在core.js中有扩展(65行)
* 视图内容初始化在ViewManager的switch\_mode方法中初始化，视图的widget就是ViewManager的controller，controller的初始化具体在do\_create\_view方法中，这些视图都有一个参数options，这个参数是在ViewManager的start方法中初始化(622行)
* openerp.web.Controller在core.js中
* ListView在view\_list.js第9行开始
* webclient初始化时初始化菜单和注册菜单事件在show\_application中，self.menu.on(‘menu\_click’, this, this.on\_menu\_action) (1281行)
* on\_menu\_action方法(1422行)中会进行远程调用**this**.rpc("/web/action/load,…)，在回调函数中会调用actionManager的do\_action方法
* ActionManager在文件view.js中(12行)：
* web.Dialog在chrome.js中定义(68行)
* web.View 在views.js中定义(1359行)
* button点击会自动创建数据的问题：如果button包含special属性(不管属性内容是什么)，就不会强制创建数据
* form视图下的widget组件注册在view\_form.js中(6279行)
* hashchange事件绑定在webclient中(1378行)
* 时区转换在format.js中format\_value方法转换，在各种视图widget中有各自的format方法调用该方法。如view\_list里面的format方法(2224行)
* form视图中弹出窗口的基类定义在5150行，instance.web.form.AbstractFormPopup类
* DataSet的使用方式
* Model创建和初始化是在DataSet中(data.js 788行)，视图会初始化DataSet
* ProxyDataSet继承自DataSetSearch，DataSetSearch继承自DataSet