目录：

[0、开发前的几个必备](#准备)

[1、打开Eclipse，新建一个UI5 project，并且关联到FCD系统](#步骤1)

[2、创建component.js文件、i18n文件，view和controller等文件夹](#步骤2)

[3、创建一个主页面的view和controller](#步骤3)

[4、创建master view，实现选择订单，可以跳转它的Detail界面](#步骤4)

[5、创建detail view](#步骤5)

[6、创建createEntity view页面](#步骤6)

[7、在master和detail页面跳转到创建页面](#步骤7)

[8、在createEntity view页面，实现sold-to和product的检查。Item的添加和删除](#步骤8)

[9、在createEntity view页面，实现Header和Item的修改可以定位到具体修改的字段](#步骤9)

[10、在createEntity view页面，实现cancel，或者是跳转到detail页面后，清除变量和跳转](#步骤10)

[11、在createEntity view页面，实现创建和修改订单的传输，以及返回的message处理](#步骤11)

[12、创建FilterDialog.fragment.xml文件](#步骤12)

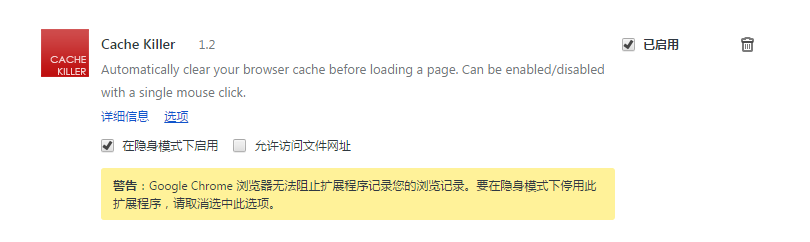
[13、在master view界面，添加一个搜索按钮，实现搜索](#步骤13)

[14、在每个页面实现连接后端的时候，出现busy的情况](#步骤14)

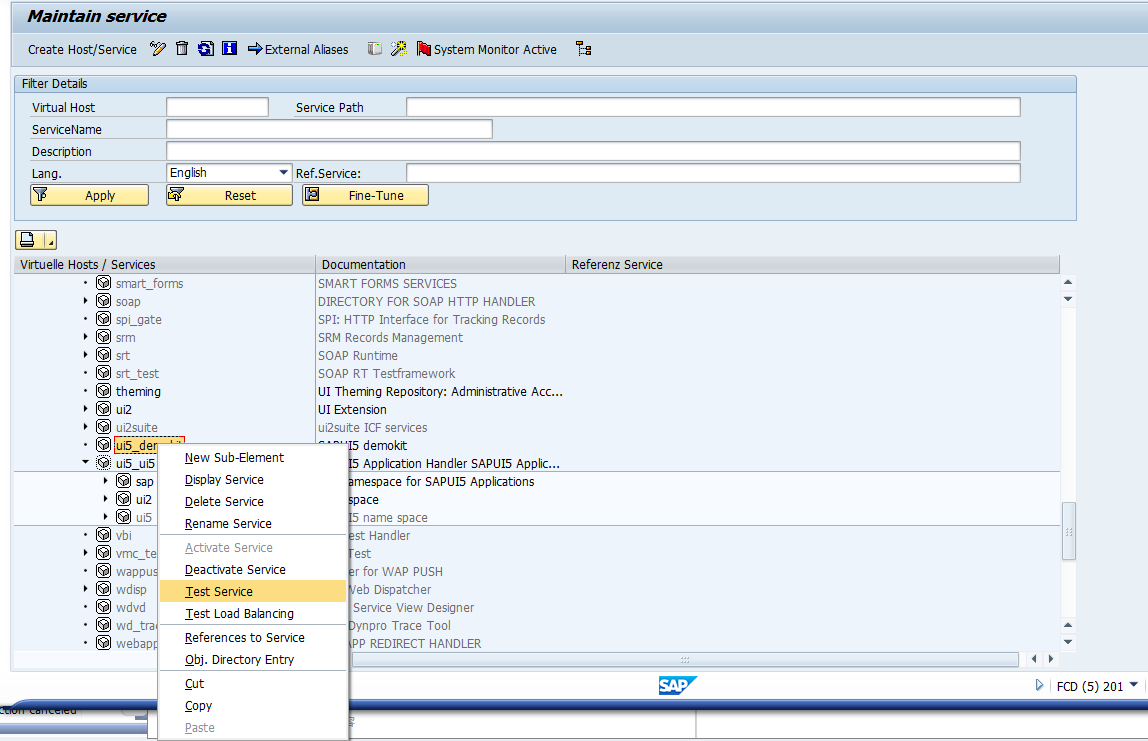
[15、错误](#错误)

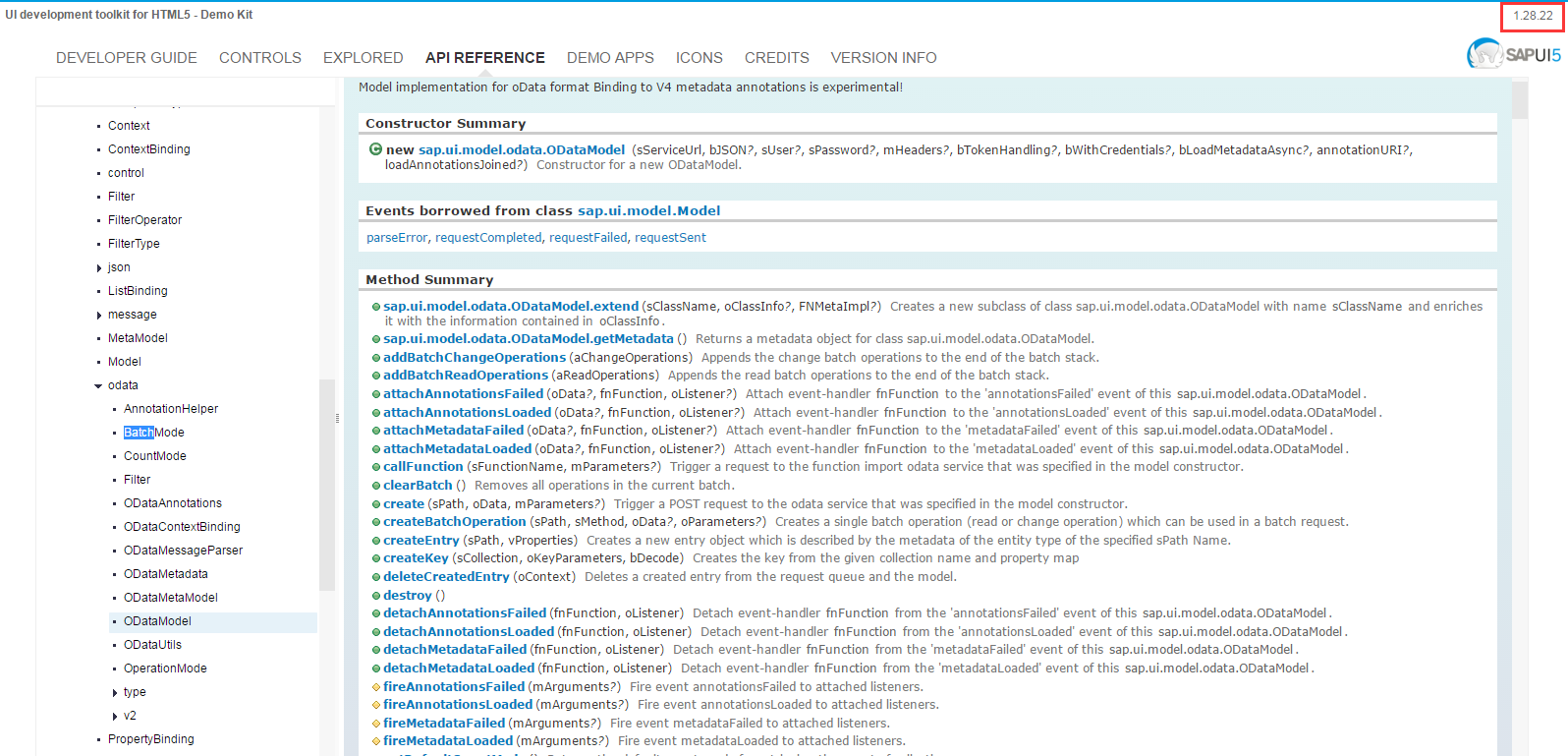
开发之前几个必备：

准备1、安装浏览器的catch killer（谷歌浏览器翻墙后，直接搜索安装）



准备2、通过gateway输入t-code：SICF，查看服务器的UI版本，并且通过查看服务器的SDK，基于此SDK进行开发（目前服务器的版本是1.28，而SAP官方的版本是1.36）。



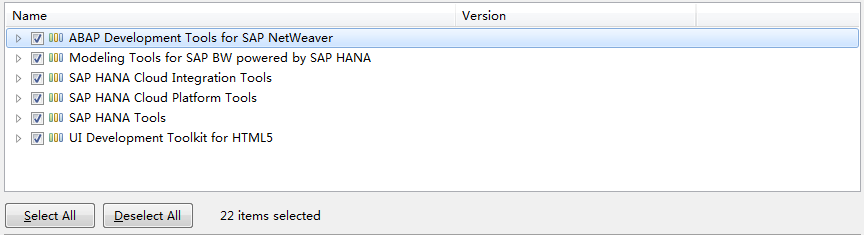


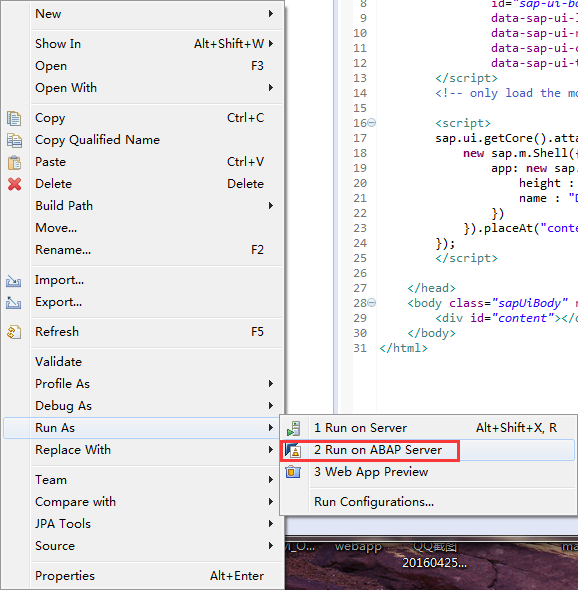
准备3、如果用到WEB IDE建template项目的，可以参考<http://scn.sap.com/docs/DOC-55465> 下载和使用WEB IDE

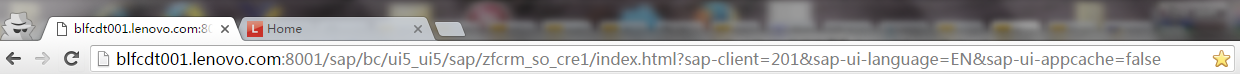
准备4、测试UI的时候，打开catch killer同时进入隐身模式（谷歌浏览器：CTRL+SHIFT+N），这样可以清除缓存和cookie。

准备5、添加gateway前端登陆的权限，没有权限的通过间接登陆的方式去打开（在eclipse里面右键项目，选择run on ABAP server，登陆后，再打开FLP）：

准备6、eclipse里面装好了下面这些包，SAP GUI添加好FCD。

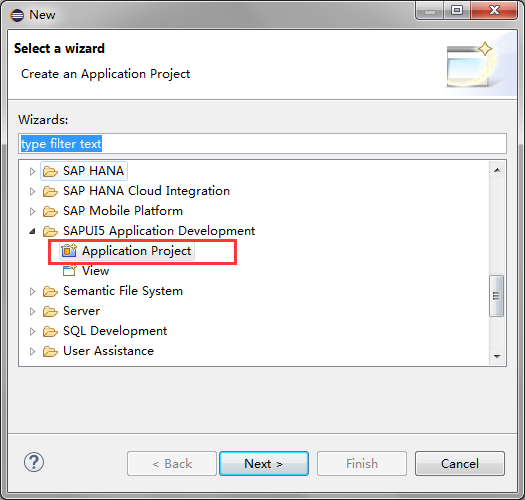


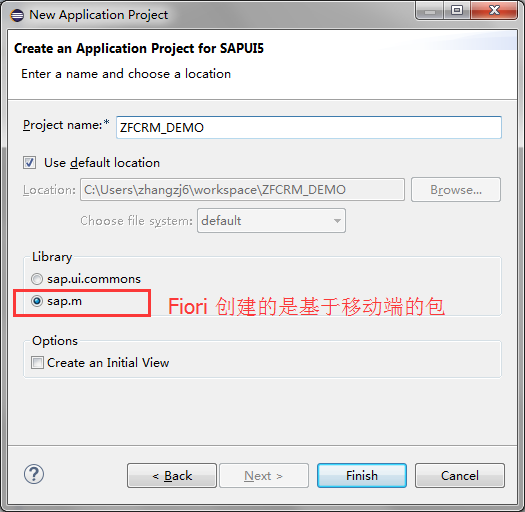


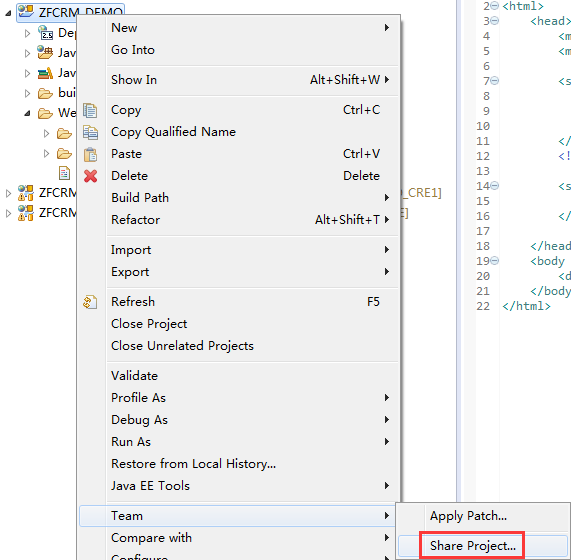


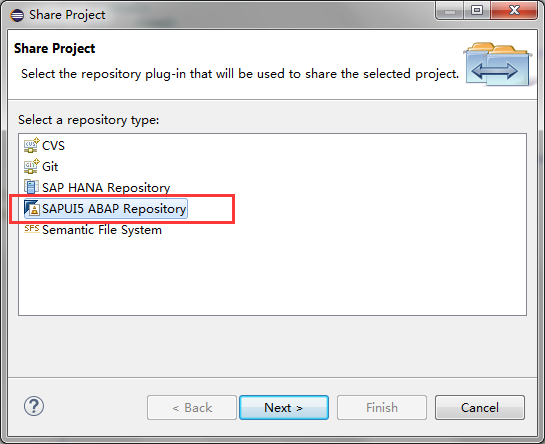
步骤：（步骤2开始你也可以通过WEB IDE去创建template或者是扩展SAP目前已有的项目，然后再去修改和添加你的逻辑。除非是已有的template和项目非常符合你的需求，不然不建议走这步。）

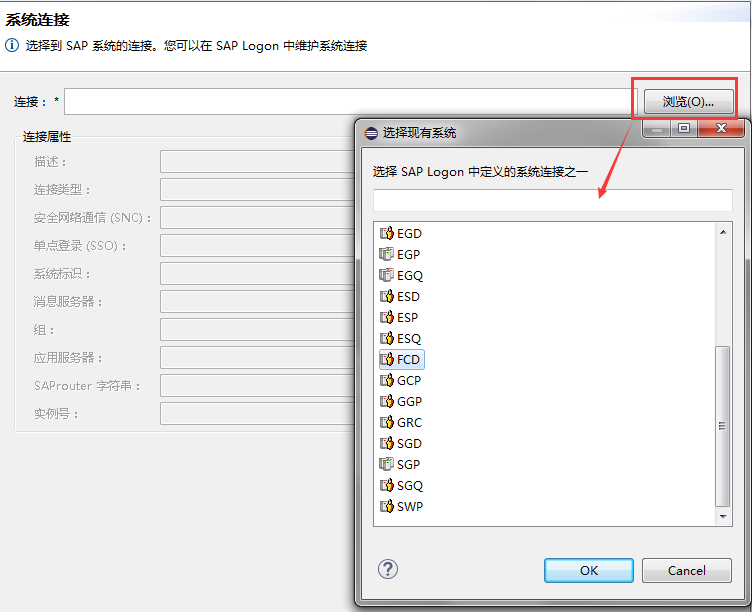
1、打开Eclipse，新建一个UI5 project，并且关联到FCD系统。

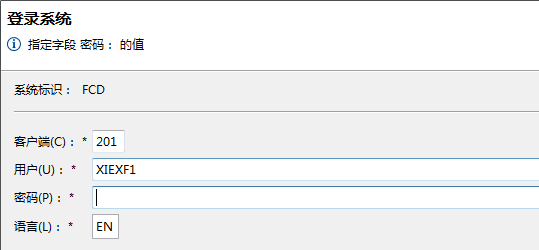


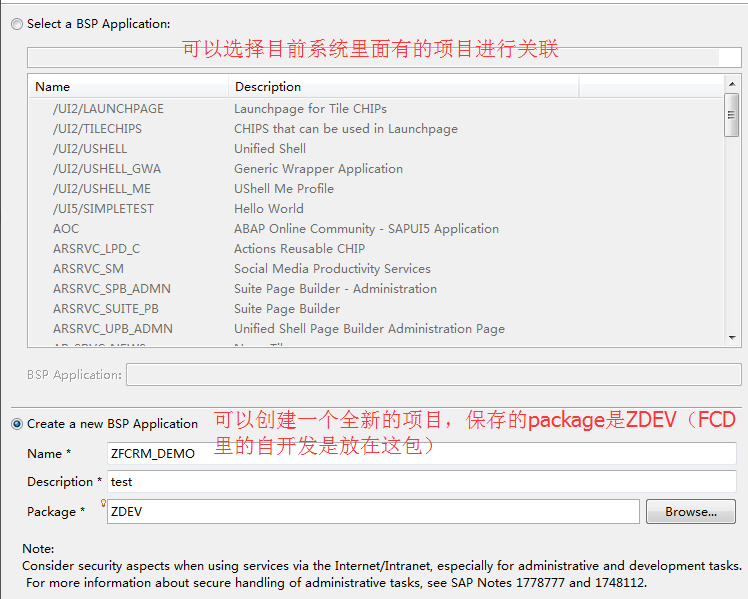






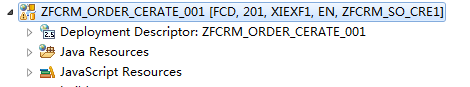




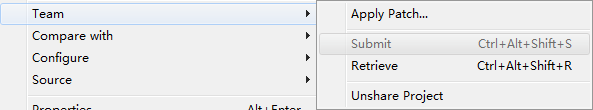




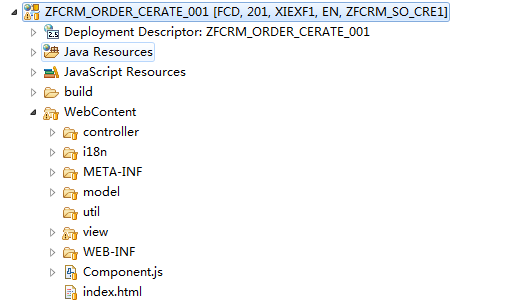
关联好之后：

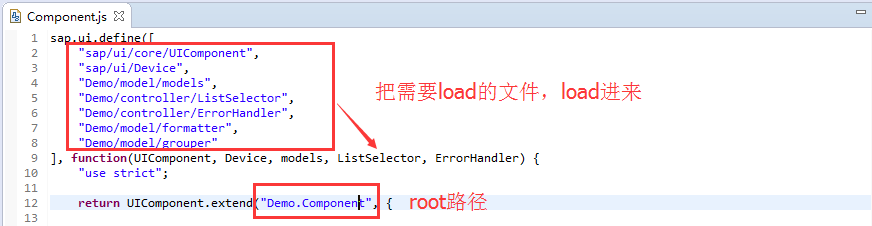


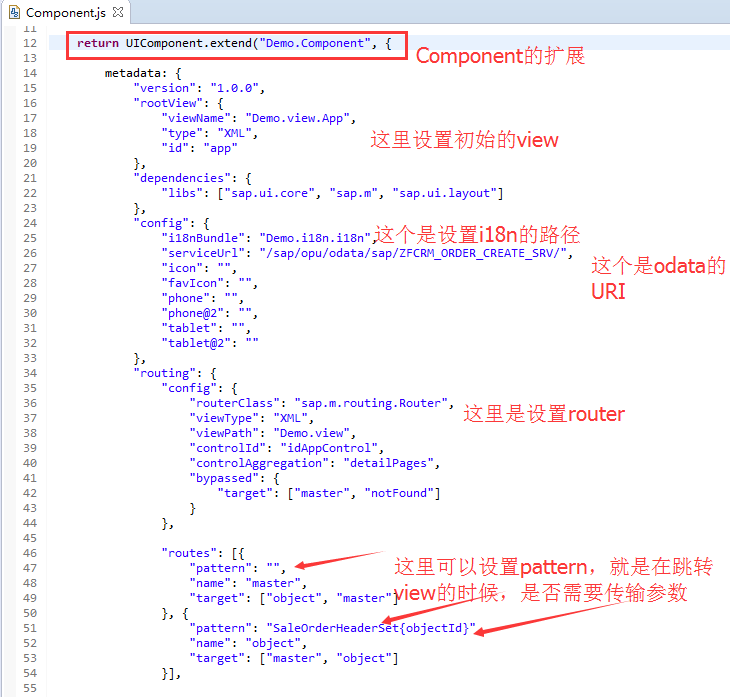
以后每一次修改完毕都要commit才会生效：



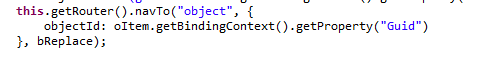
2、创建component.js文件、i18n文件，view和controller等文件夹。















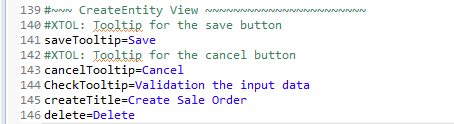
获取component里面的odatamodel：



也可以自己定义：



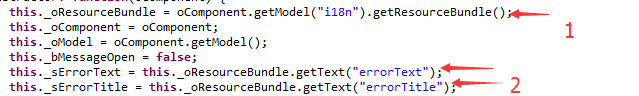
I18n的使用例子：



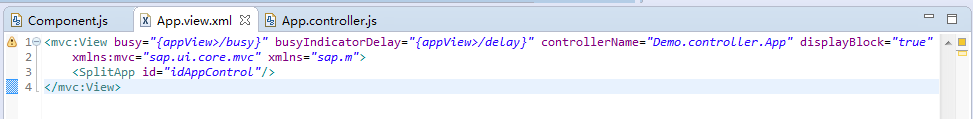
View里面：



Controller里面：



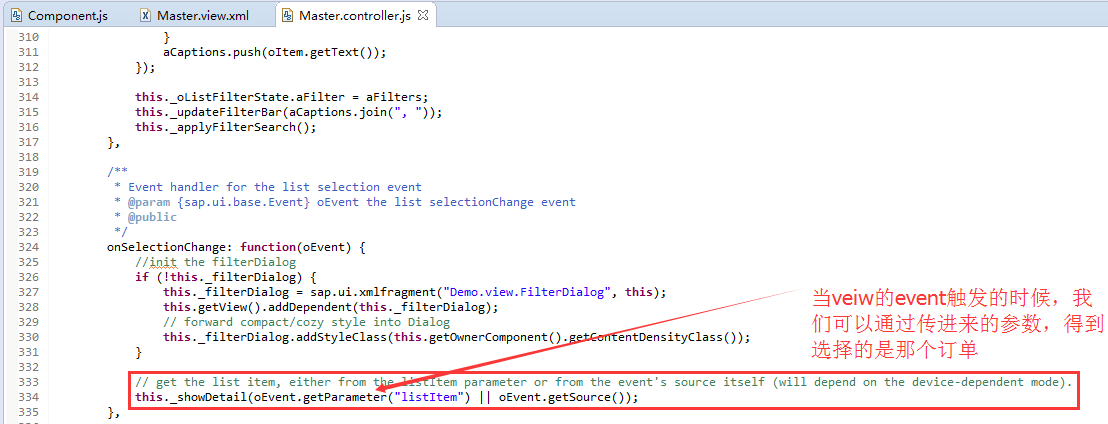
3、创建一个主页面的view和controller：





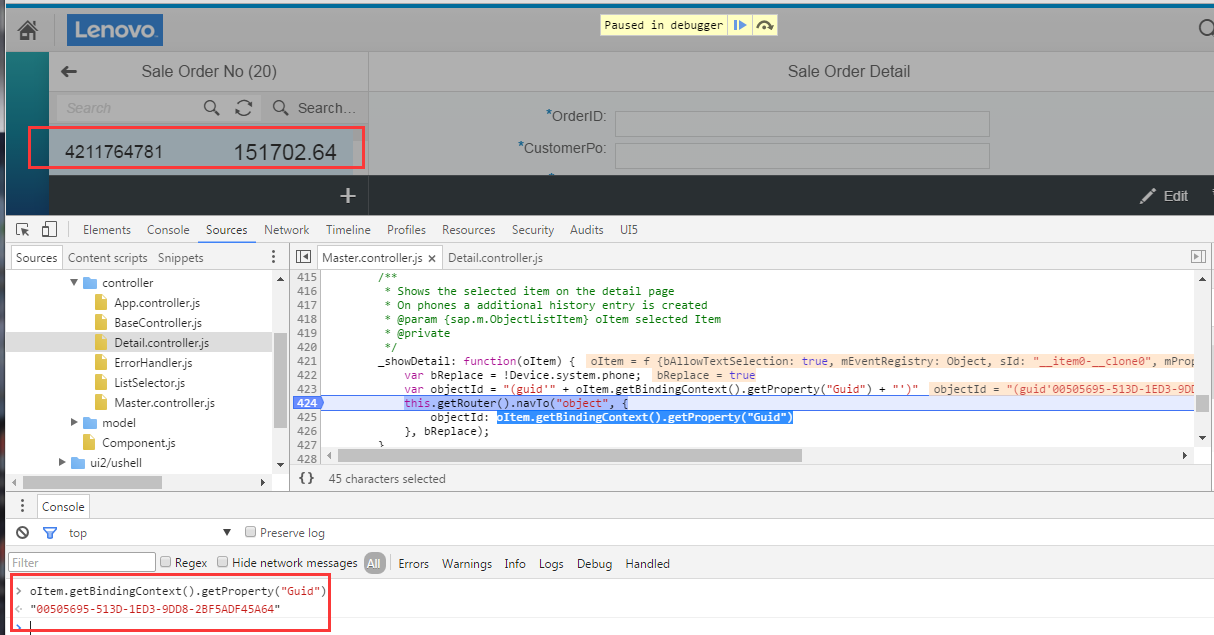
4、创建master view，实现选择订单，可以跳转它的Detail界面：





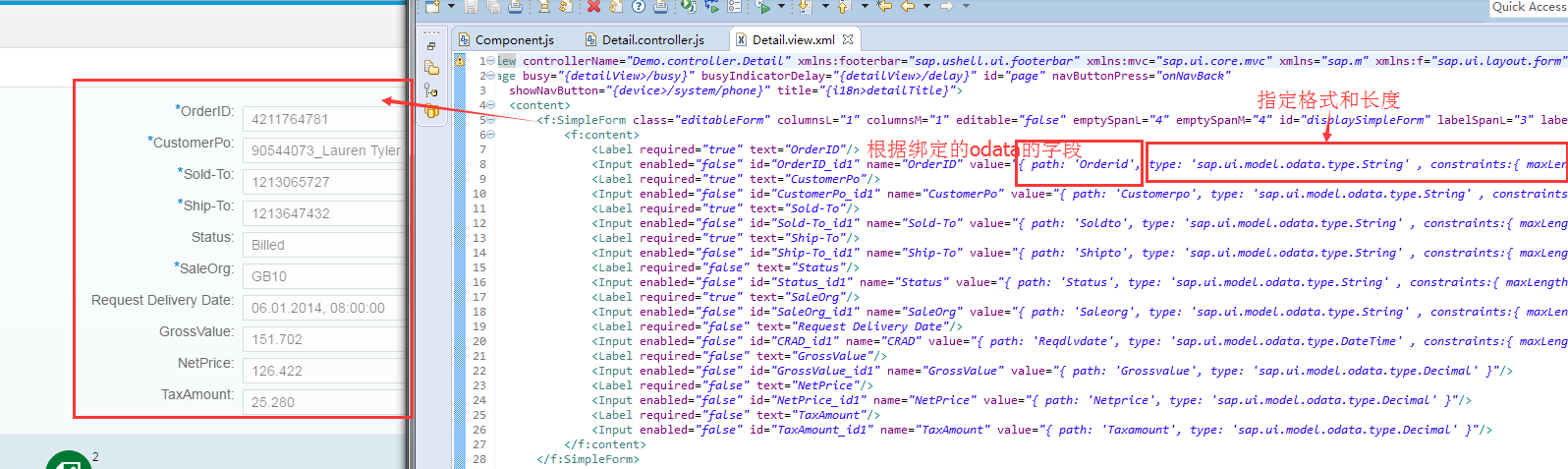


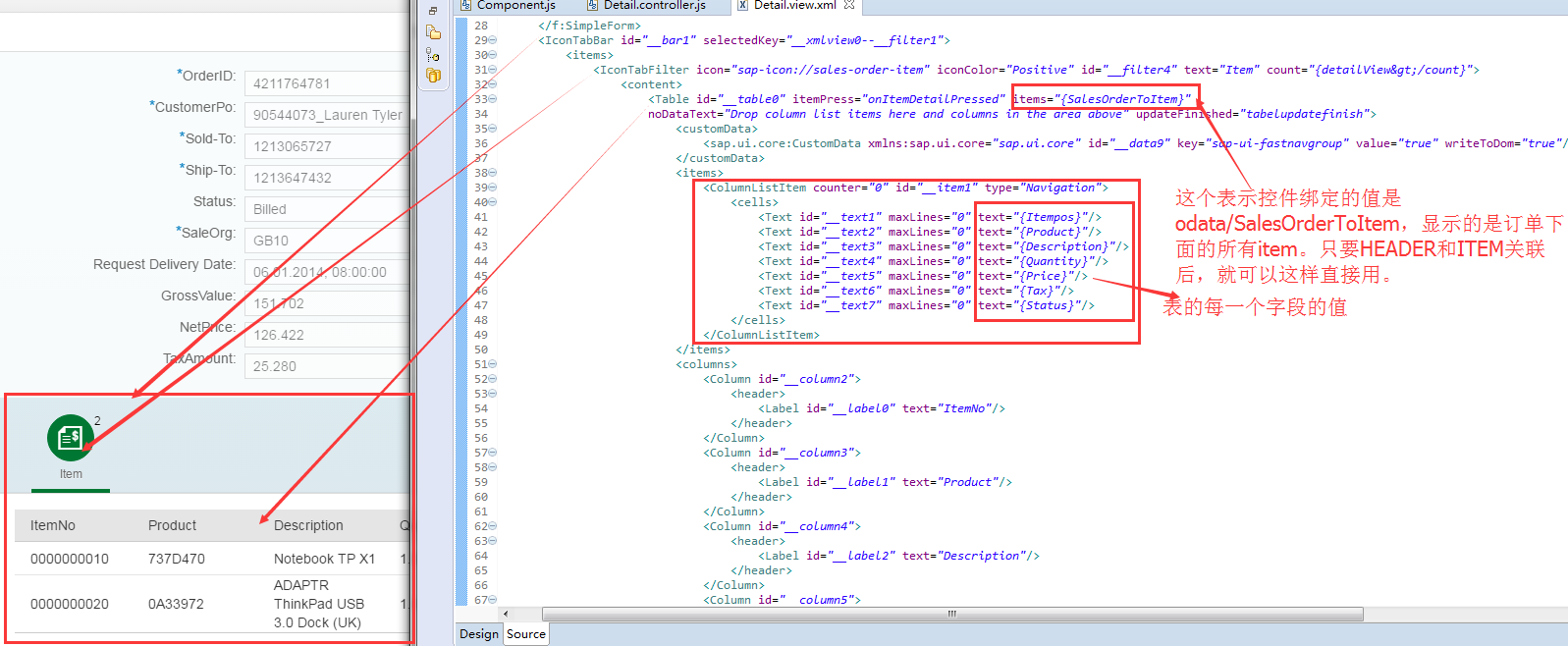
完成跳转：



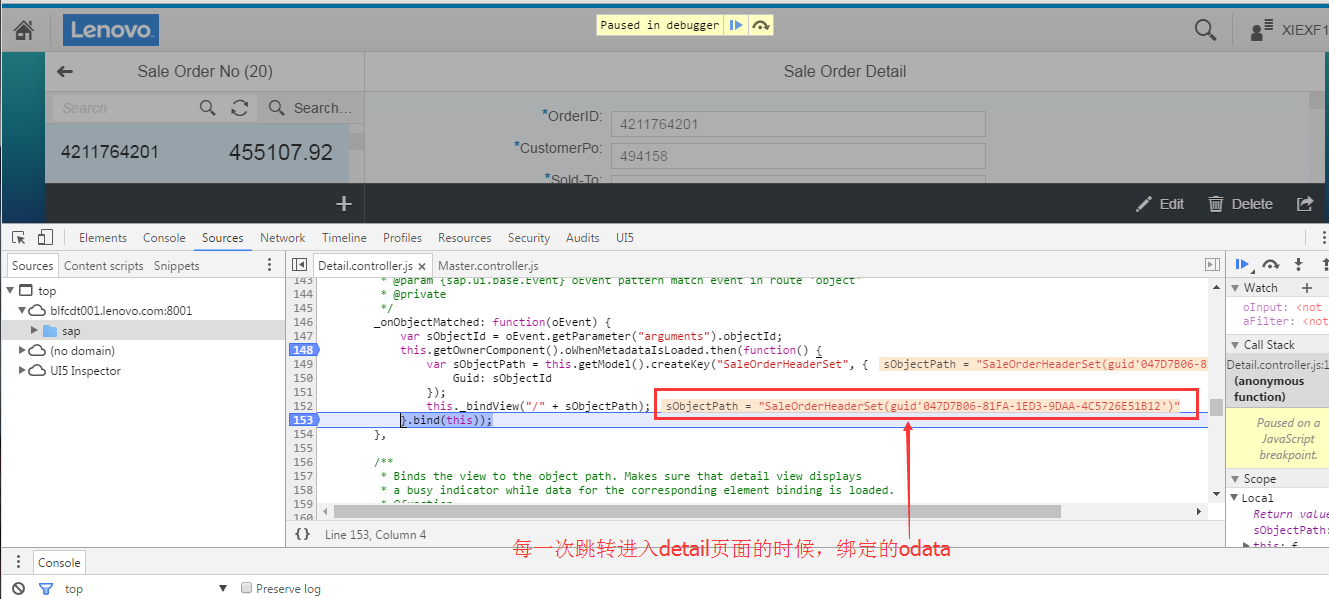
5、创建detail view，显示订单的Header和Item的数据界面

界面：





跳转的时候，当从master controller传入GUID过来的时候，需要去

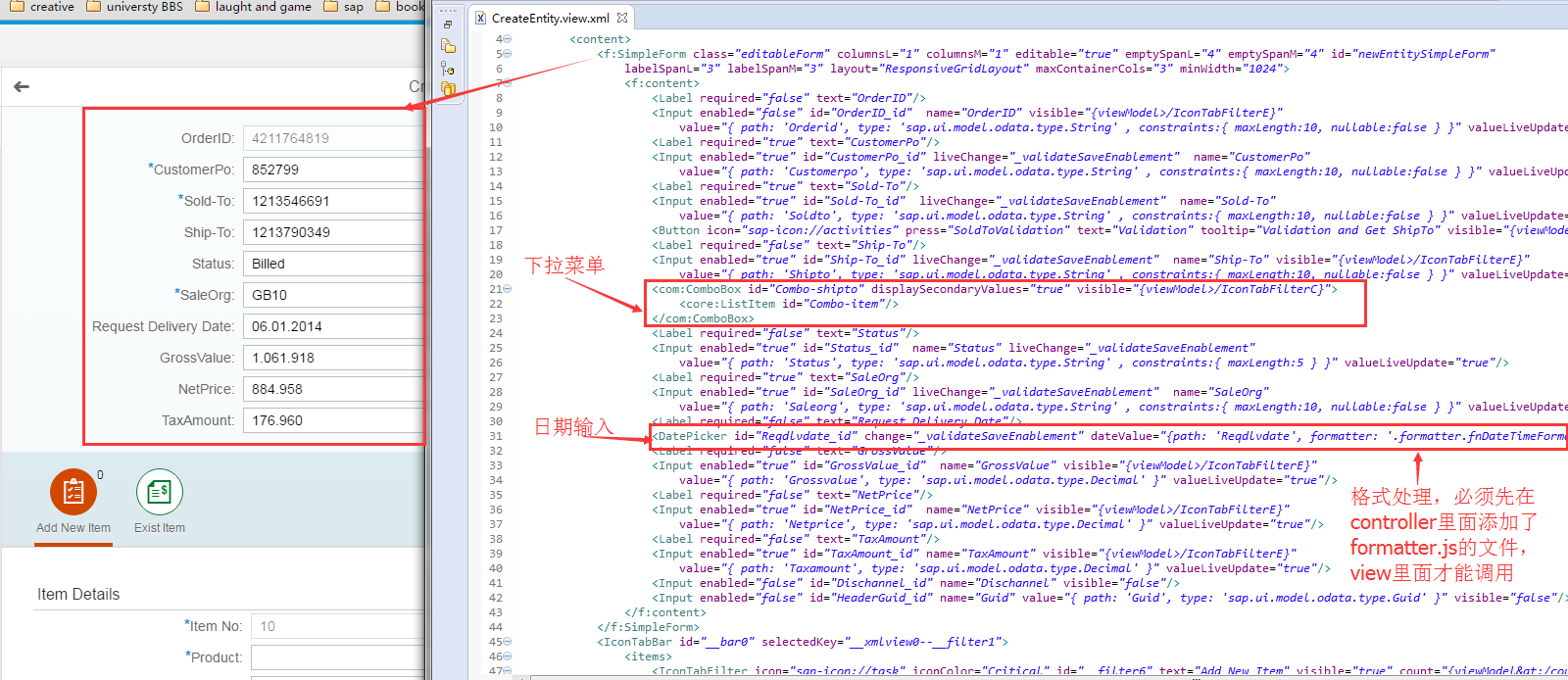


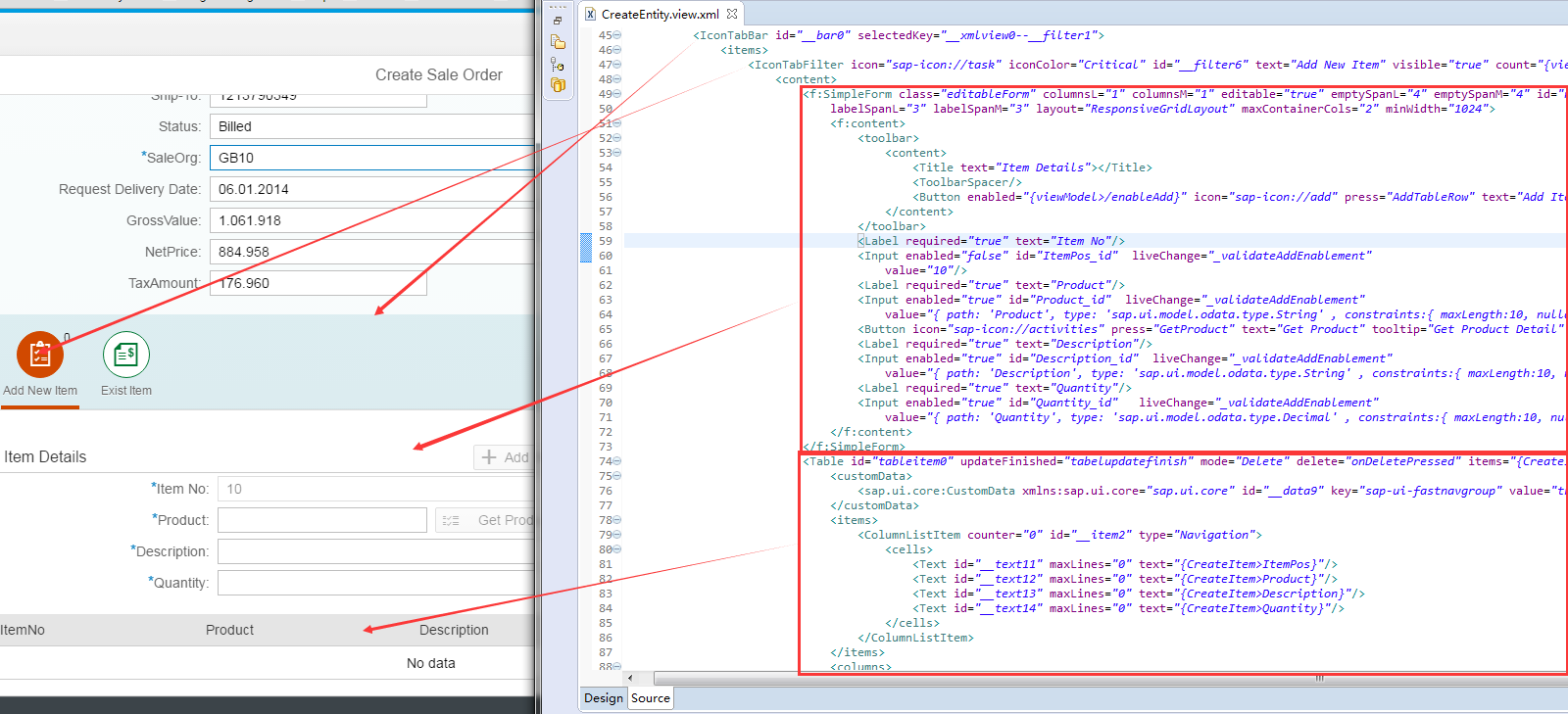
6、创建createEntity view页面，显示需要输入和修改

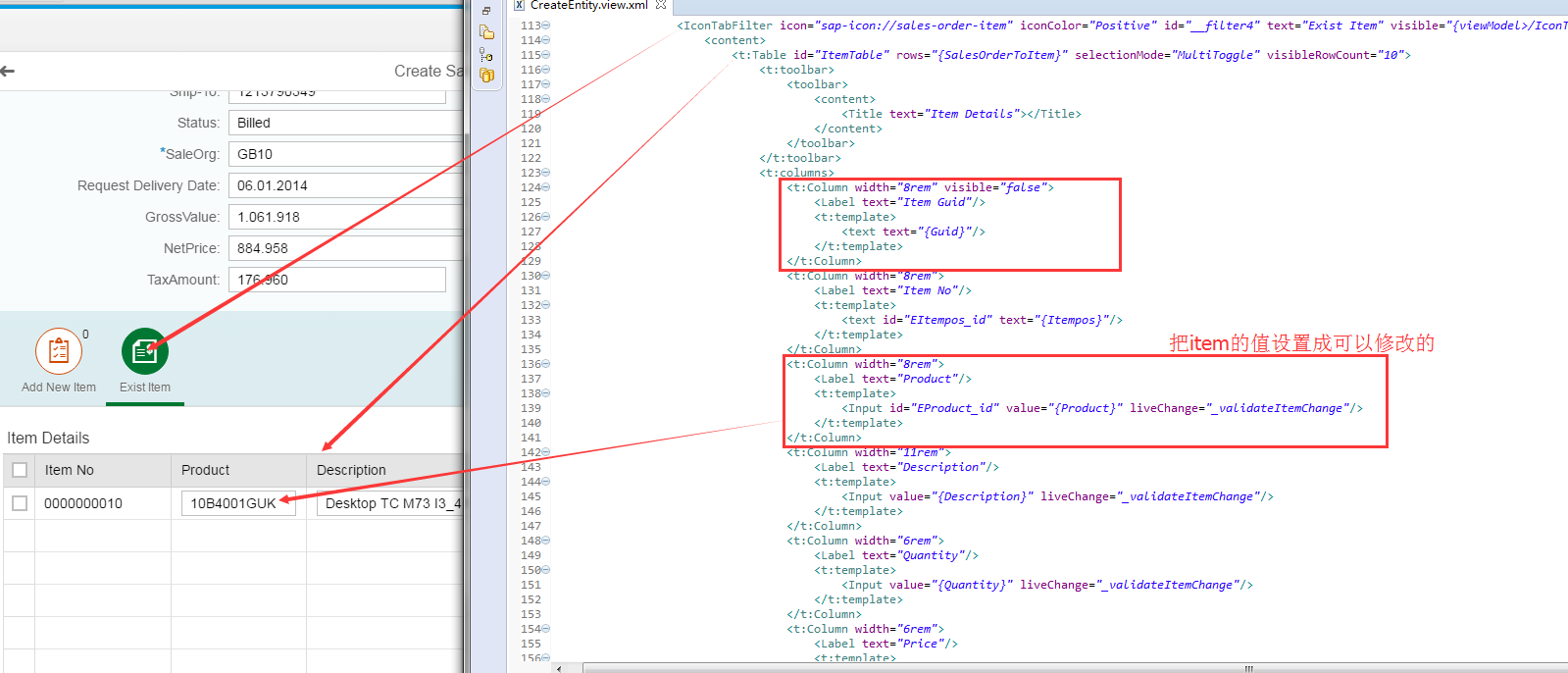
因为这个页面是编辑和创建公用的，所以有些字段需要通过一个JSON model的不同字段去控制在那种情况下才显示：







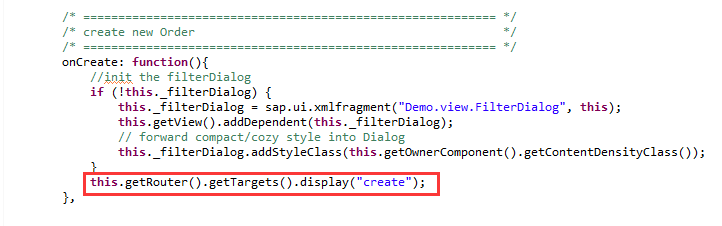


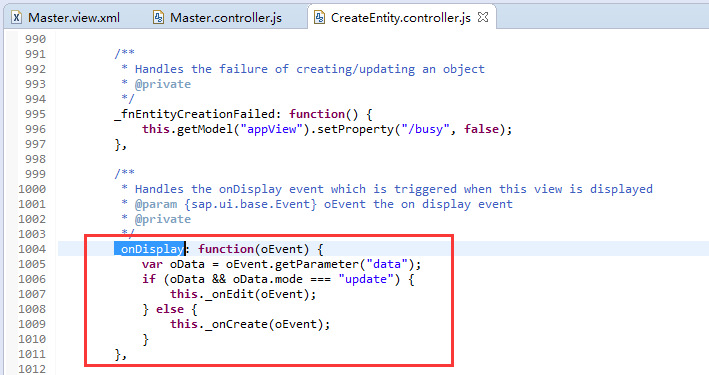


7、在master view里面新建一个创建按钮，点击，跳转到创建的页面，在detail页面点击edit跳转到创建页面

创建：



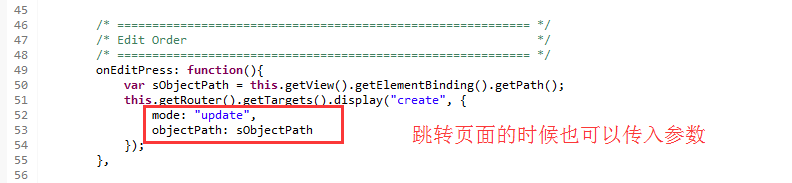






编辑：







8、在createEntity view页面，实现sold-to和product的检查。Item的添加和删除：

Sold-to检查：

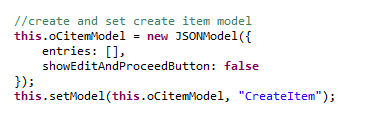


Product检查：

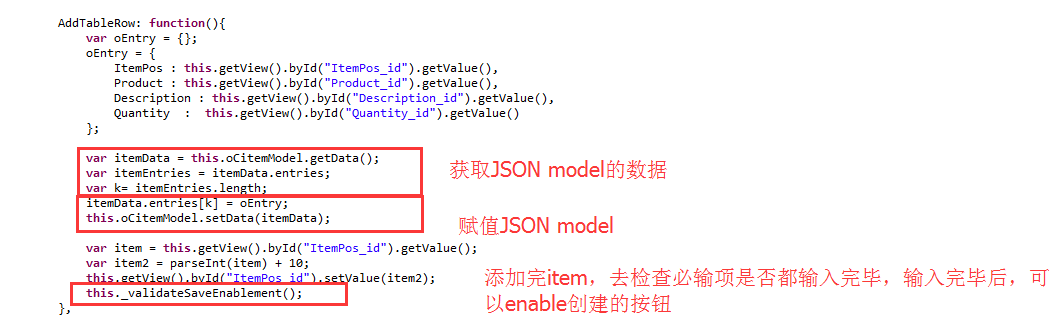


Item的添加和删除：

需要输入的item的表是绑定的JSON model，那么添加item的时候就是把输入添加到对应的JSON model里面，同理删除的处理。



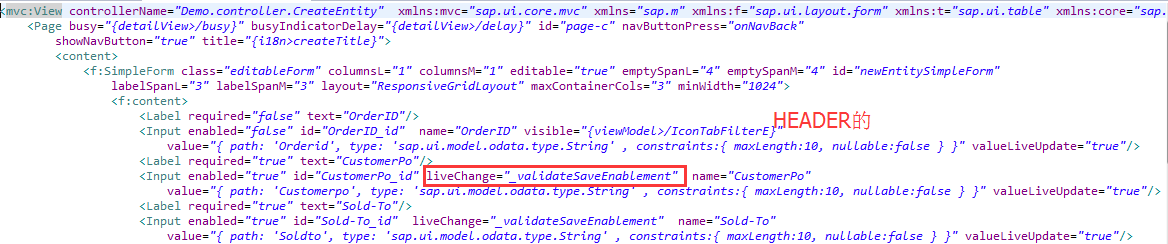


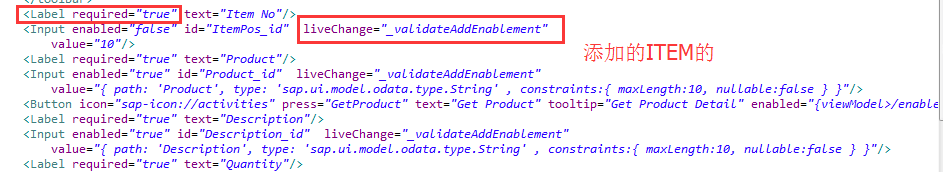


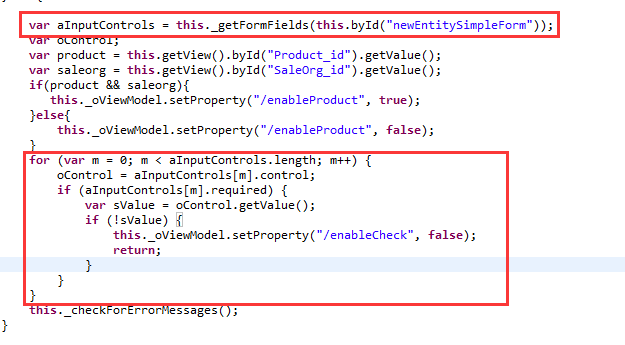
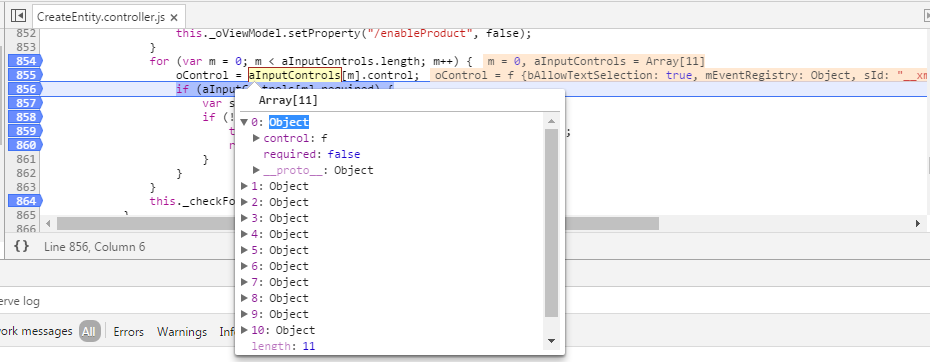


必输项的检查（同时还可以判断saleorg和product都输入了，才能点击product检查的按钮）：

通过检查空间的required的属性，然后判断它的值是否存在。







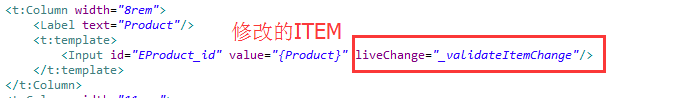
9、在createEntity view页面，实现Header和Item的修改可以定位到具体修改的字段：

这个功能是基于步骤8的，在必输项检查里面，当是编辑的时候，把修改的字段的ID保存到JSON model。

Header的修改：



Item的修改：





10、在createEntity view页面，实现cancel，或者是跳转到detail页面后，清除变量和跳转：



11、在createEntity view页面，实现创建和修改订单的传输，以及返回的message处理：

创建的保存：





编辑的保存：



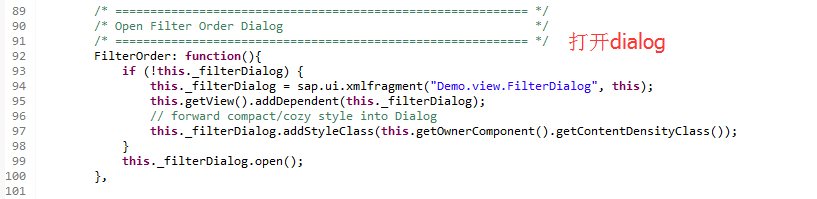




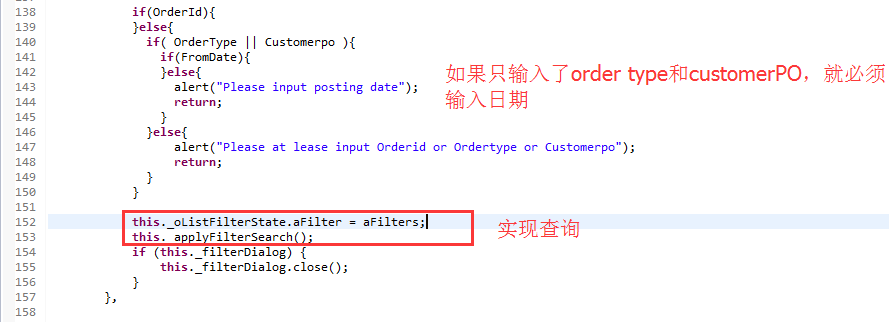
12、创建FilterDialog.fragment.xml文件：



13、在master view界面，添加一个搜索按钮，实现搜索：









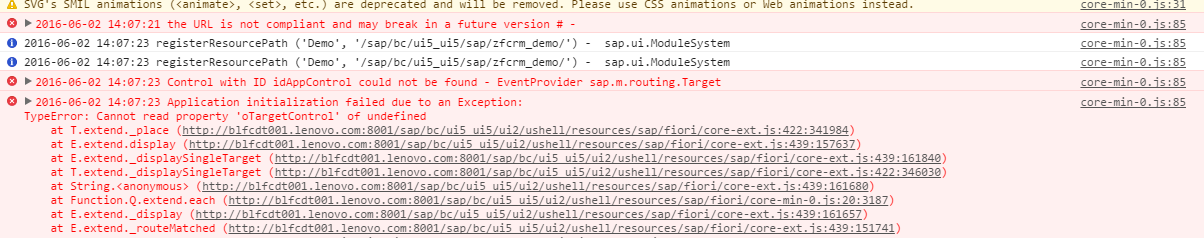
14、在每个页面实现连接后端的时候，出现busy的情况：

可能出现的错误：

错误1、share project到FCD的时候可能会出现这样的错误：

错误2、dialog的弹出失败：

错误3、跨应用可能出现这样的错误：



因为两个项目用的用一个路径：Demo，会导致它去load同一个component，所以这里把两个app的路径改成不一样的就好了。

错误4、