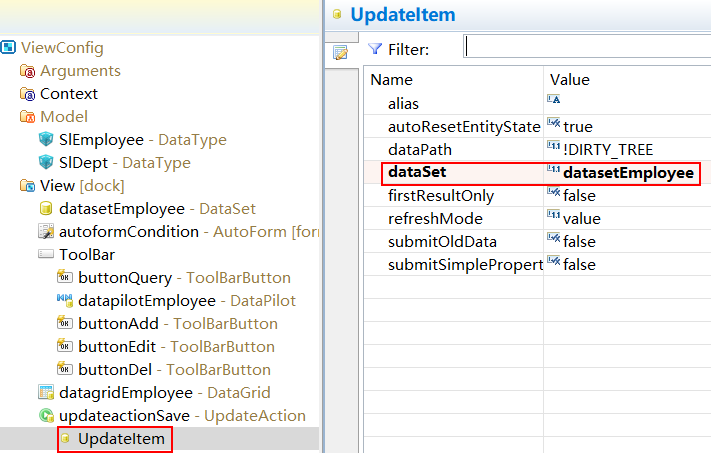
默认情况下通过UpdateAction提交数据时候，都是通过UpdateItem的dataPath定义哪些数据需要提交。



在一些特殊情况下如果只是根据dataPath设定提交数据的范围有一定的局限性，我们希望可以根据某种业务规则，选取数据集中的部分实体对象进行提交。这样更能应付复杂的情况。对于这种需求可以利用UpdateAction的onGetUpdateData事件。

onGetUpdateData定义为：当UpdateAction尝试获取每个UpdateItem对应的提交数据时触发的事件。如果我们不定义这个事件，则会根据UpdateItem的dataPath属性提取需要提交的实体对象，如果定义了这个事件，UpdateAction会尝试从方法体中的arg.data获取需要提交的数据。例如如果我们这么定义onGetUpdateData事件：

// @Bind #actionSave.onGetUpdateData

!function(arg){

arg.data = view.get("#dsProducts").getData("#");

//设定需要提交的实体对象为dsProducts的当前记录

}

最终这个UpdateAction执行的时候就只提交dsProducts中的当前记录。

另外如果UpdateAction中有多个UpdateItem，此事件在提交过程中会针对每一个UpdateItem触发一次。因此编写代码的时候就需要区别对待不同的UpdateItem：

// @Bind #actionSave.onGetUpdateData

!function(arg){

if (arg.updateItem.alias=="dataset1"){

arg.data = view.get("#dataset1").getData("#");

//设定需要提交的实体对象为dataset1当前记录

}

else if (arg.updateItem.alias=="dataset2"){

//什么都不做，默认根据UpdateItem的dataPath获取数据的操作

}

else if (arg.updateItem.alias=="dataset3"){

arg.data = view.get("#dataset3").getData();

//设定需要提交的实体对象为dataset3的所有数据

}

else if (arg.updateItem.alias=="dataset4"){

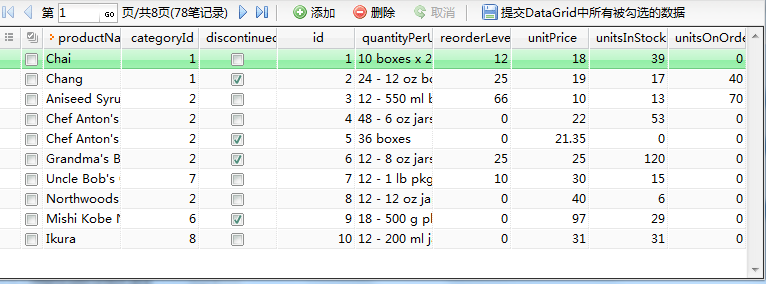
arg.data =

view.get("#dataset4").getData("[@.get(/"categoryName/").length>5]");//设定需要提交的实体对象为dataset4中所有产品类型名称长度大于5的数据

}

}

## 示例：



该页面中通过onGetUpdateData事件自定义提交数据的使用。

本例中将自定义提交的数据设计为Grid中选中的数据，则我们可以在onGetUpdateData事件中这么定义代码：

// @Bind #actionSave.onGetUpdateData

!function(arg) {

arg.data = view.get(“#dataGrid.selection”);

}

这样最终这个UpdateAction执行的时候就会将Grid中选中的行提交到服务器。