**myproject项目总结**

1. **使用Maven原型创建项目**
   1. **使用maven原型opendaylight-startup-archetype的1.3.0-Carbon版本创建项目**

在用户目录下创建目录c14，在目录下输入以下命令

mvn archetype:generate \

-DarchetypeGroupId=org.opendaylight.controller \-DarchetypeArtifactId=opendaylight-startup-archetype \

-DarchetypeCatalog=local \

-DarchetypeVersion=1.3.0-Carbon

输入项目基本信息

Define value property ‘groupId’::com.ming.myproject

Define value for property ‘artifacId’::myproject

...

...

创建成功

* 1. **编译项目**

进入最顶层的目录（myproject/）,运行以下命令跳过测试编译项目

$mvn clean install -DskipTests

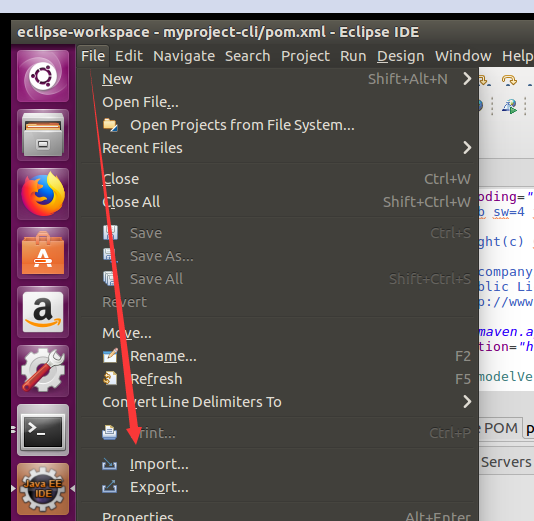
...

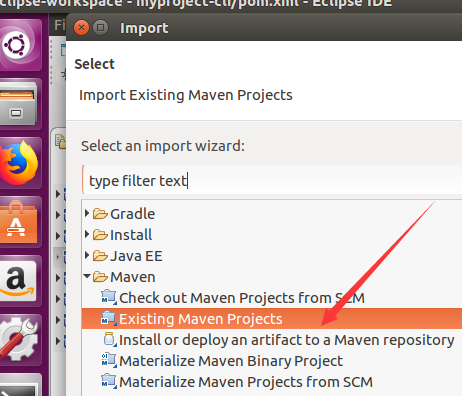
...

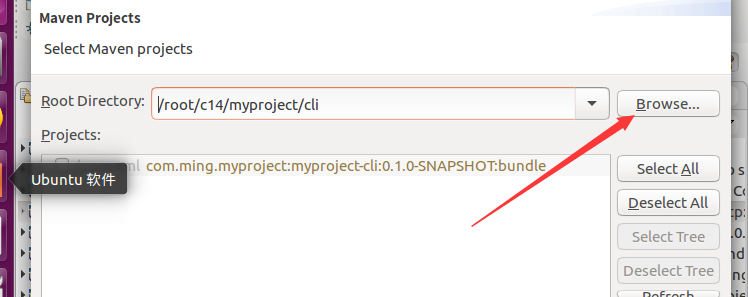
项目编译成功

* 1. **将项目导入eclipse IDE中**

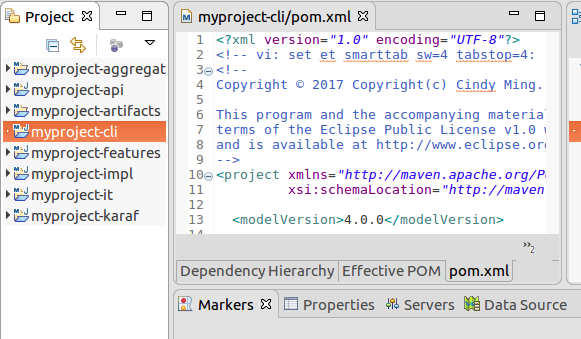
file→import





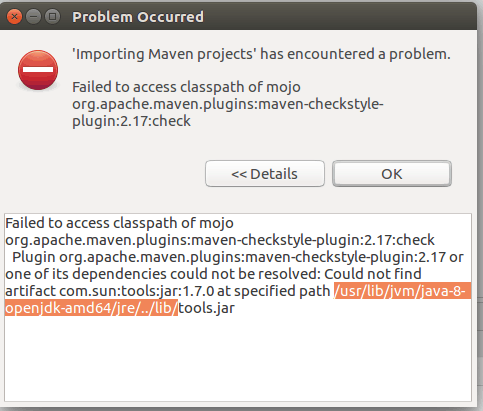


找到相应项目导入即可



* 1. **问题解决**

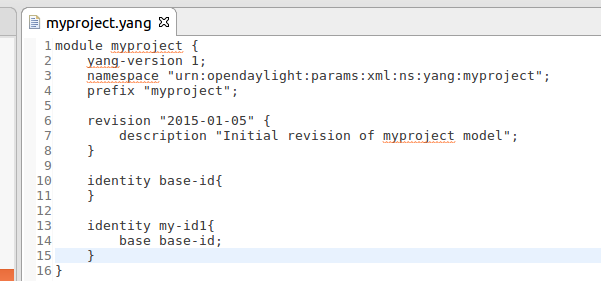
主要问题是



主要是tools.jar，它找不到。找到相应的tools.jar，然后复制到相对应的目录下就好了。可能在主机目录下不能找到上面所示的目录，缺少../lib，自己手动创建一个，把tools.jar放进去就好了

1. **YANG常用的定义及其自动转化的Java代码**

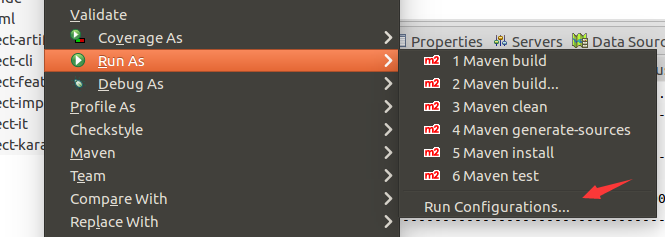
**2.1在模块myproject中定义identity声明，代码如下**

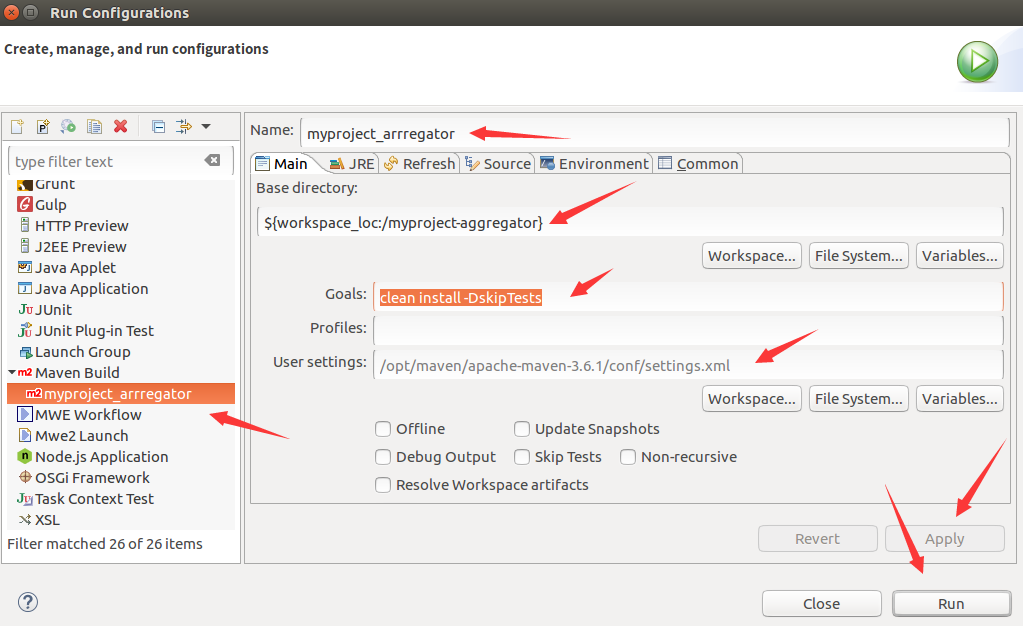


先编译后刷新

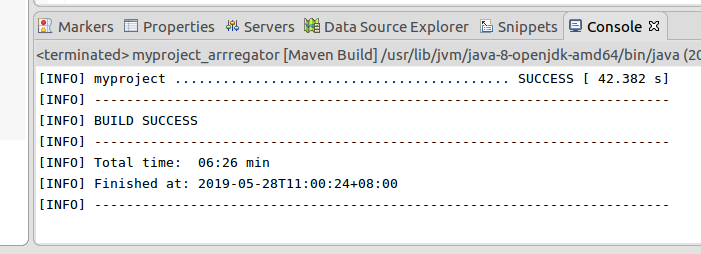
编译

单击顶层目录下的pom.xml文件，右击“Run Configurations...”,接着进行配置





配置好了之后点击apply ,run,等待一段时间

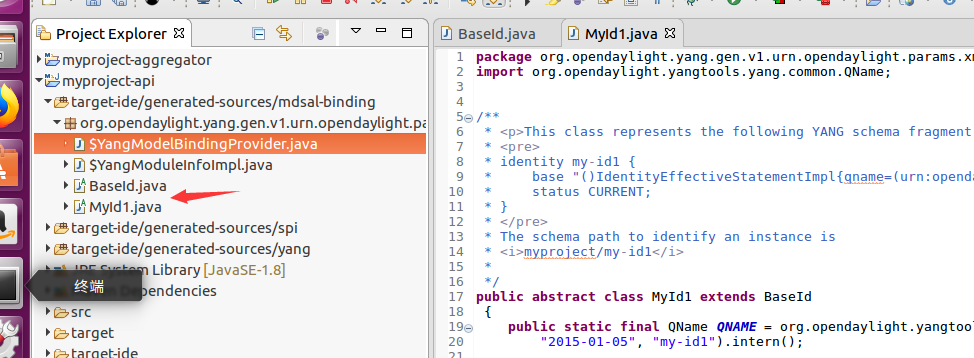


编译成功！！！

刷新：

在Package Explorer中选择所有目录，右击Refresh，刷新项目后出现新生成的2个映射的Java文件，BaseId.java和MyId.java

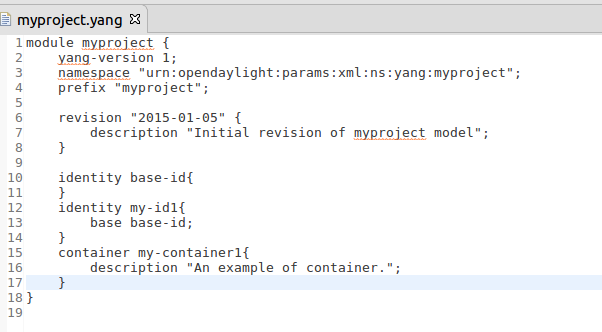
期间出现问题，刷新之后一直找不到这两个java文件，最后将项目文件从eclipse中全部删除，然后重新导入项目，再次刷新，就出现了这两个文件



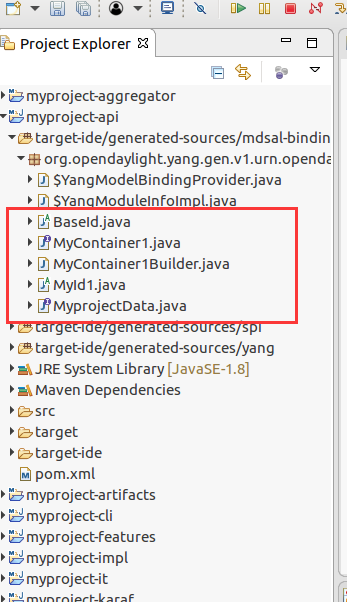
**2.2 container声明实例及其生成的Java文件**

**1.不带孩子声明的container声明**

在myproject.yang文件中加入一下代码

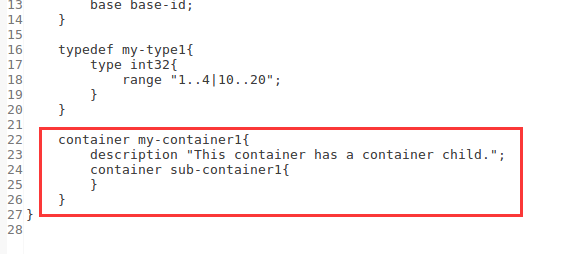


然后编译刷新（见2.1操作步骤）

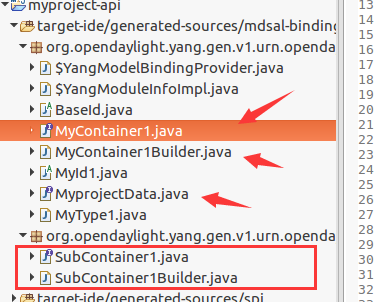


1. 带孩子的声明的container声明

接下来看一个带孩子元素的container的映射

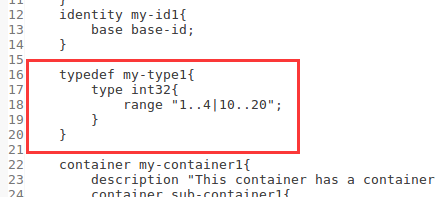


编译之后刷新

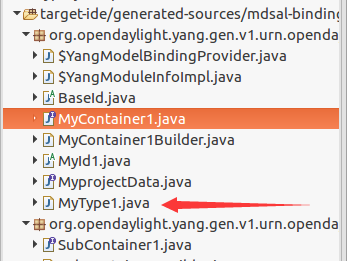


**2.3 typedef声明实例及其生成的java文件**

在模块myproject中定义2个typedef声明（注意：一个typedef声明实例只能包含一个typedef声明），代码如下

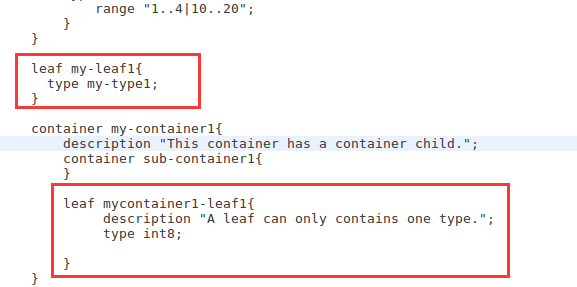


编译之后刷新



生成了一个映射文件java MyType1.java

**2.4 leaf声明实例及其生成的Java文件**

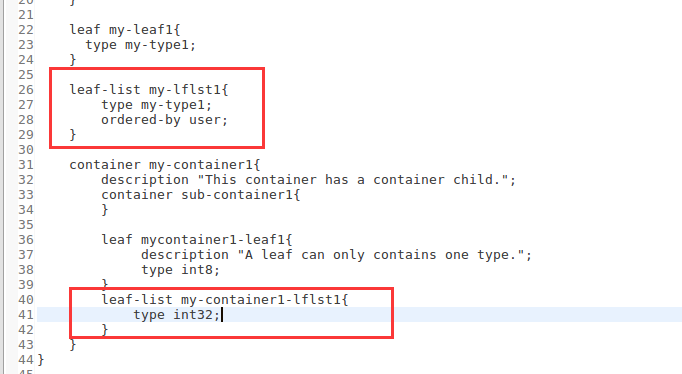


leaf不单独映射java文件，它只对现存文件造成改动

具体改动看书（p415），此处不再截图

**2.5 leaf-list 声明实例及其生成的Java文件**

一个leaf-list声明只能包含一个type子声明。

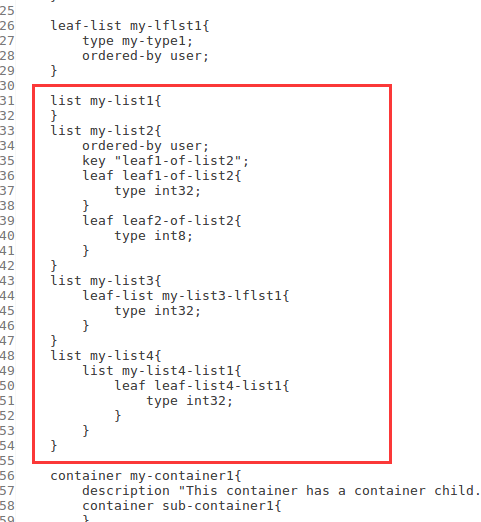


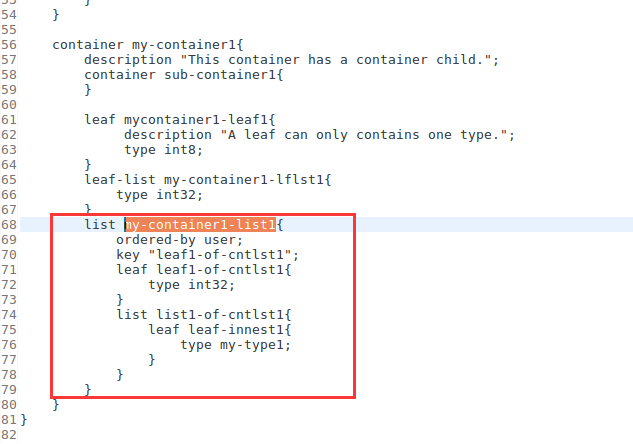
Leaf-list 不单独映射Java文件，它只对现存文件造成改动

具体改动不再演示

书（p419）

**2.6 list声明实例及其生成的Java文件**





编译之后刷新

由于篇幅较长，这里不再截图相应的增加包和java文件，详情看书（p425）

2.7 choice声明和case声明实例及他们生成的java文件

...

...

...

1. **YANG创建模型的一些实验**

在项目里打开odl控制器

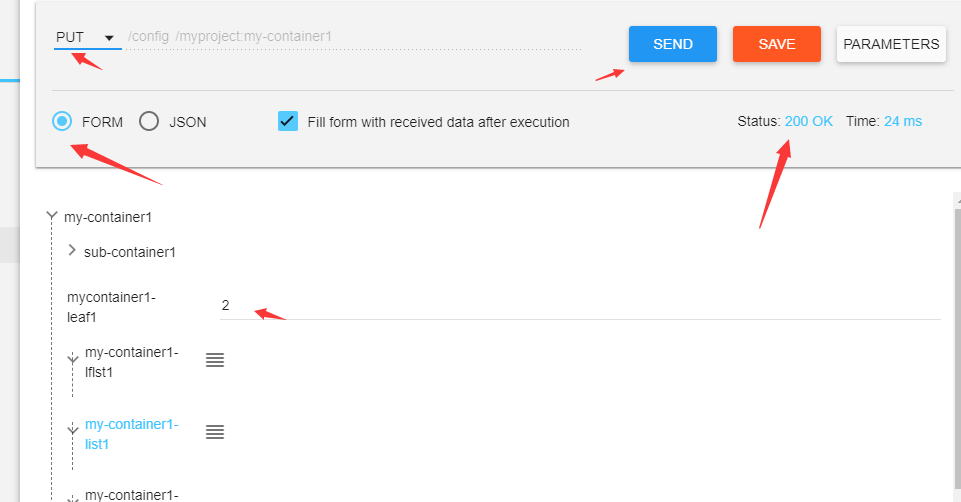
然后在浏览器上输入http://<Controller’IP Address>:8181/index.html#/yangui/index

由于现在最新用的是yangman，当没有出现yangui也不要意外（功能都一样）

**实验1：**

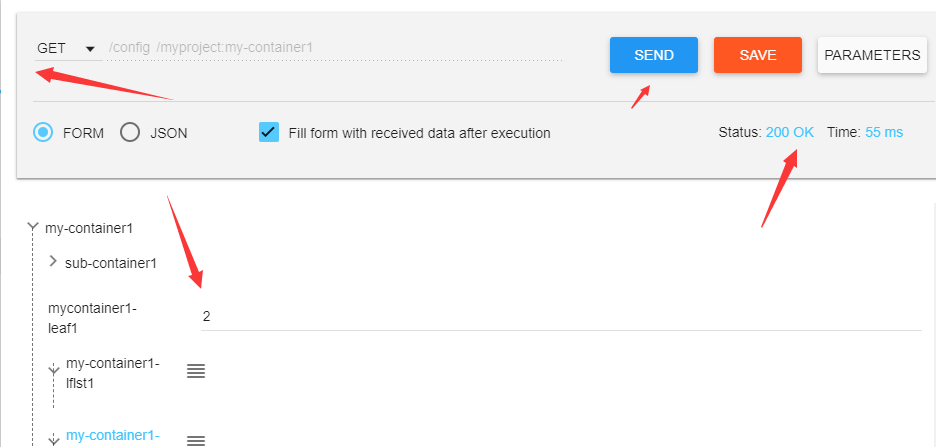
设定my-container1的叶子mycontainer1-leaf的值为2.

先PUT



Send成功后

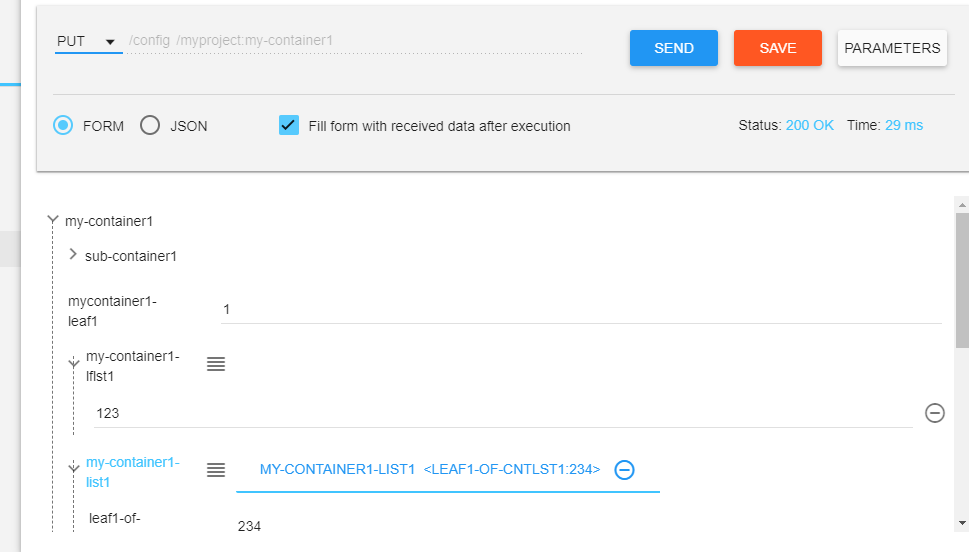
下面验证一下值是否设置成功，选择GET,点击send

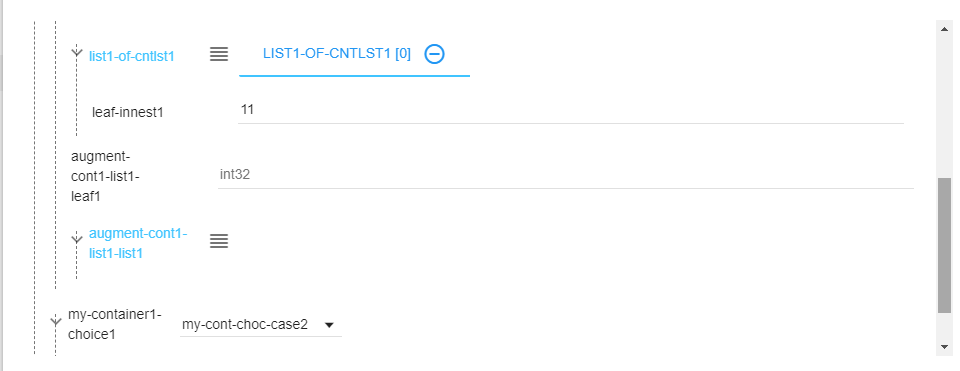


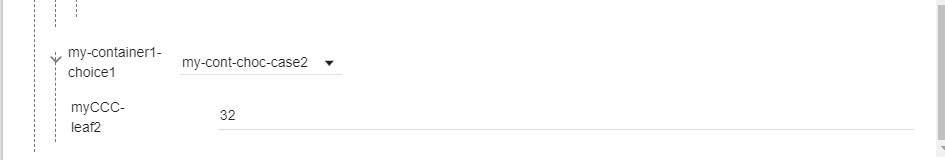
设置成功

**实验2**

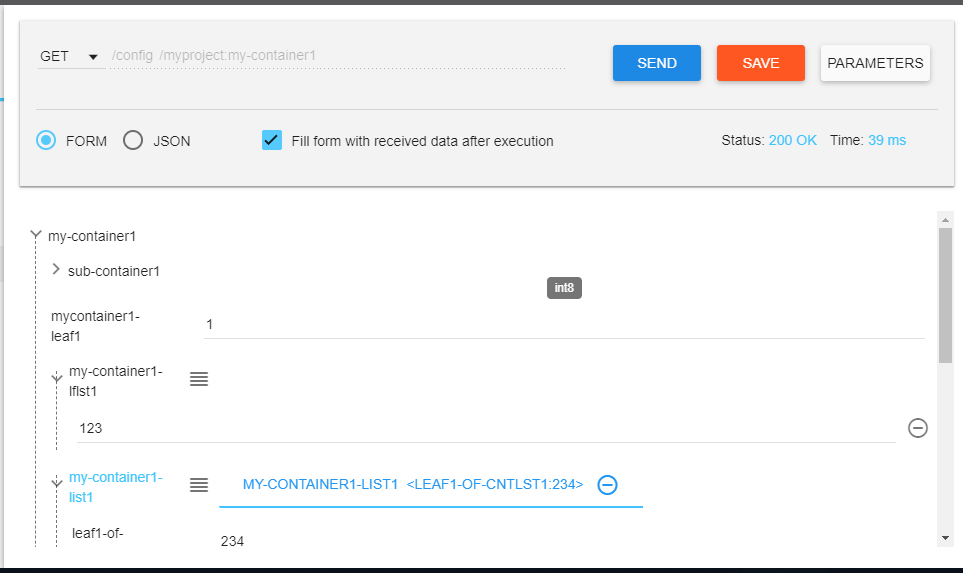
将my-container1的叶子mycontainer1-leaf1的值设置为1，leaf-list类型my-container1-lflst1的第一个子节点的值设置为123，将my-container1-list1的第一个子节点的自叶子节点leaf1-of-cntlst1的值设置为234、子列表节点list1-of-cnlst1的第一个元素的叶子节点设置为11，将case子节点my-container1-choice1的情况设置为my-cont-choc-case2,接着讲my-cont-choc-case2的子节点myCCC-leaf2设置为32，单机send按钮，成功上传数据，并且检查成功

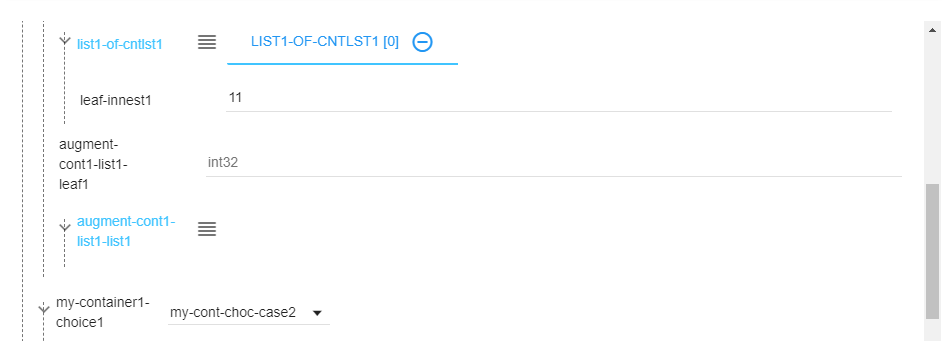






查询：

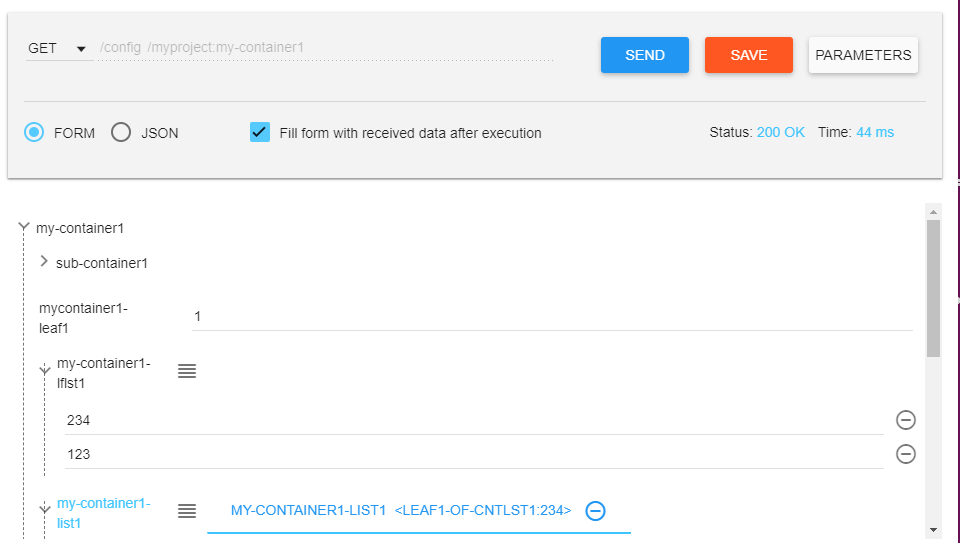


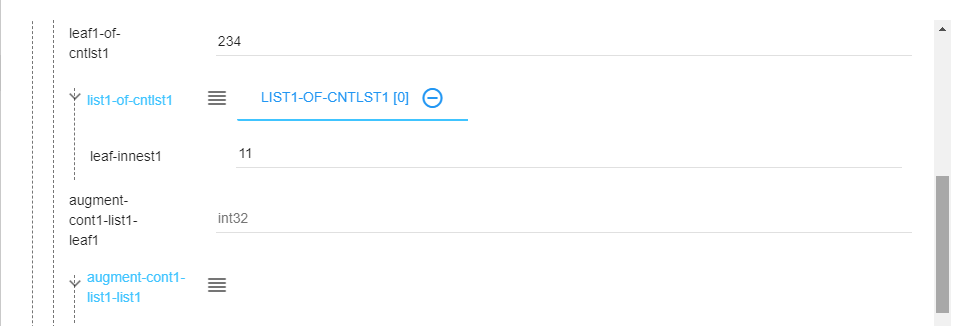




**实验3：**

将my-container1的叶子mycontainer1-leaf1的值设置为1，leaf-list类型my-container1-lflst1的第一个子节点的值设置为123，第二个子节点的值设置为234，将my-container1-list1的第一个子节点的叶子子节点leaf1-of-cntlst1的值设置为234、子列表家电list-of-cntlst1的第1个元素的子叶子节点设置为11，将case子节点my-container1-choice的情况设置为my-cont-choc-case1时的子节点myCCC-leaf1设置为string,单击Send按钮，成功上传数据，并检查成功

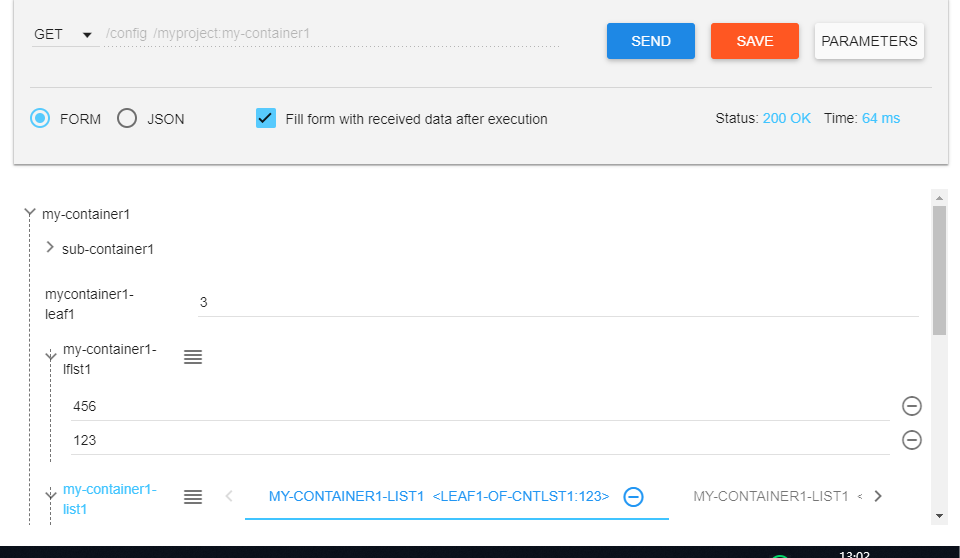


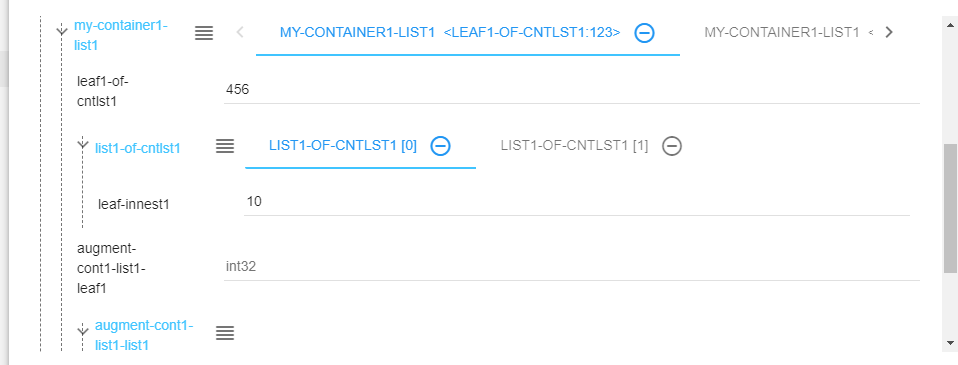




**实验4：**

将my-conainer1的叶子mycontainer1-leaf1的值设置为3，leaf-list类型my-container1-lflst1的第一个自系欸但的值设置为123，第二个子节点的值设置为456.将my-container1-list1的第一个子节点的子叶子节点leaf1-of-cntlst1的值设置为123，列表节点list1-of-cntlst1的第1个元素的叶子节点设置为1；第二个子节点的子叶子节点leaf1-of-cntlst1的值设置为456，子列表节点list1-of-cntlst1的第1个元素的叶子节点设置为10、第二个元素的叶子节点设置为11.将case子节点my-container1-coice1的情况设置为my-cont-choc-case2时的子节点myCCC-leaf设置为234，单击Send按钮，成功上传数据，并检查成功！







**实验5：**

列表my-list2的相关实验

将列表my-list2的第一个元素的叶子节点leaf-oflist2(同时也是键)的值设置为32，将叶子节点leaf2-of-list2的值设置为1，单击Send按钮，成功上传数据，并检查成功。

