一、以订单管理为例：

第一步，创建订单对象

```
import java.util.Date;

/**
 * Created by CHS on 1/10/18.
 */
@URIANT(pathname = "bean")
@TableANT(name="TB_BEAN")
public class Order {

    @ComponentANT(name = "A",title = "id",inputType = "input",inputSubtype = "hidden")
    @URIANT(pathname = "A")
    @ColumnANT(name="A",type = "varchar",length=30,isKey = true,nullable = false,unique = true)
    private String id;

    @ComponentANT(name = "B",title = "code",inputType = "input",inputSubtype = "text",rights = 2)
    @URIANT(pathname = "B")
    @ColumnANT(name="B",type = "varchar",length=50,nullable = false,unique = true)
    private String code;

    @ComponentANT(name = "C",title = "amount",inputType = "input",inputSubtype = "text",rights = 2)
    @URIANT(pathname = "C")
    @ColumnANT(name="C",type = "integer")
    private int amount;

    @ComponentANT(name = "D",title = "total",inputType = "input",inputSubtype = "text",rights = 2)
    @URIANT(pathname = "D")
    @ColumnANT(name="D",type = "integer")
    private int total;

    @ComponentANT(name = "E",title = "createTime",dateFormat = "yyyy-MM-dd HH:mm")
    @URIANT(pathname = "E")
    @ColumnANT(name="E",type = "timestamp")
```

第二步，创建服务 ACTION 对象

```
import java.util.ArrayList;
import java.util.List;

/**
 * Created by CHS on 1/10/18.
 */
@ActionANT(bean=Order.class)
public class OrderAction extends AbstractAction {

    public OrderAction() {
        super(Order.class,"N100");
        init();
    }

    private void init(){
        defaultJsonFieldSelector.selectIndex("1,2,3,4,5");
        defaultMetaComponentSelector.selectNames("A,B,C,D,E");
    }


    @HttpANT(hid="11",name="bean.11",METHOD = "get" ,URI = "/#bean")
    public void createOrderInit(){
        List<ComponentANT> var1=viewMeta.getWritableCmpMeta();
        int var2=var1.size();
        List<Component> var3=new ArrayList<>();
        for(int i=0;i<var2;i++){
            ComponentANT var4=var1.get(i);
            Component var5=Component.create(var4);
            if(var5 instanceof Input){
                Input var6=(Input)var5;
                var6.setSize(70);
                var6.setMaxlength(400);
```

```
            ComponentANT var4=var1.get(i);
            Component var5=Component.create(var4);
            if(var5 instanceof Input){
                Input var6=(Input)var5;
                var6.setSize(70);
                var6.setMaxlength(400);
            }
            var3.add(var5);
        }
        FormResponser formResponser=new FormResponser("f11",var3,"createOrder");
        actionContextAware.setResponseBody(formResponser);
    }

    @HttpANT(hid="12",name="bean.12",METHOD = "post" ,URI = "/#bean")
    public void createOrder() { create(); }

    @HttpANT(hid="13",name="bean.13",METHOD = "get" ,URI = "/#bean/id/[\\d\\.]+")
    public void getOrder() { get(); }

    @HttpANT(hid="14",name = "bean.14",METHOD = "get" ,URI = "/#bean@page")
    public void searchOrder(){
        String acceptContentType=actionContextAware.getRequestAcceptContentType();
        boolean _bJSON=acceptContentType.equalsIgnoreCase("json");
        if(_bJSON){
            actionContextAware.setJsonFieldSelector(defaultJsonFieldSelectorID,defaultJsonFieldSelector);
        }else{
            viewContextAware.setContextPathGet("/#bean");
            viewContextAware.setContextPathList("/#bean");
        }
        search();
    }
```

第三步，Deployment

二、主要组件的用法

组件名称	组件类型	用途	使用方式
TableANT	注解	与持久化相关，标记某个 BEAN 为持久化 BEAN。	作用于 BEAN
URIANT	注解	与访问地址相关，配置 BEAN 或者 BEAN 的属性在 URI 中的符号以及在前端的名称。这将影响地址映射以及影响 BEAN 序列化到 JSON 的输出结果。	作用于 BEAN 以及 FIELD
ColumnANT	注解	配置 BEAN 的某个属性映射为数据库表的某个字段。	作用于 FIELD

ComponentANT	注解	与 UI 相关，配置 BEAN 的某个属性对应的前端页面组件及展示样式，设置页面组件的行为及状态。	作用于 FIELD
ActionANT	注解	标记某个类为 ACTION 类	作用于 CLASS
HttpANT	注解	标记某个方法为 HTTP SERVICE	作用于 METHOD

三、ComponentANT 的配置说明

属性	含义	取值范围	备注
name			字符型
title	通常对应 HTML 的 title		字符型
inputType	UI 组件类型	INPUT,SELECT,TEXTAREA,CHECKBOX 等	字符型
inputSubtype	UI 组件的子类型	取值跟随上面的 inputType 属性，比如 INPUT 组件下含 TEXT,HIDDEN 子类型	字符型
rights	Present read-only or writable	1—read-only(default) 2--writable	整型

四、HttpANT 的配置说明

属性	含义	取值范围	备注
hid	唯一标识		hid 需确保在整个应用内唯一。
name	名称		字符型
METHOD	对应 HTTP 协议的 METHOD	GET/POST/PUT/DELETE 等	字符型
URI	代表 HTTP 服务的 URI，支持正则表达式		字符型