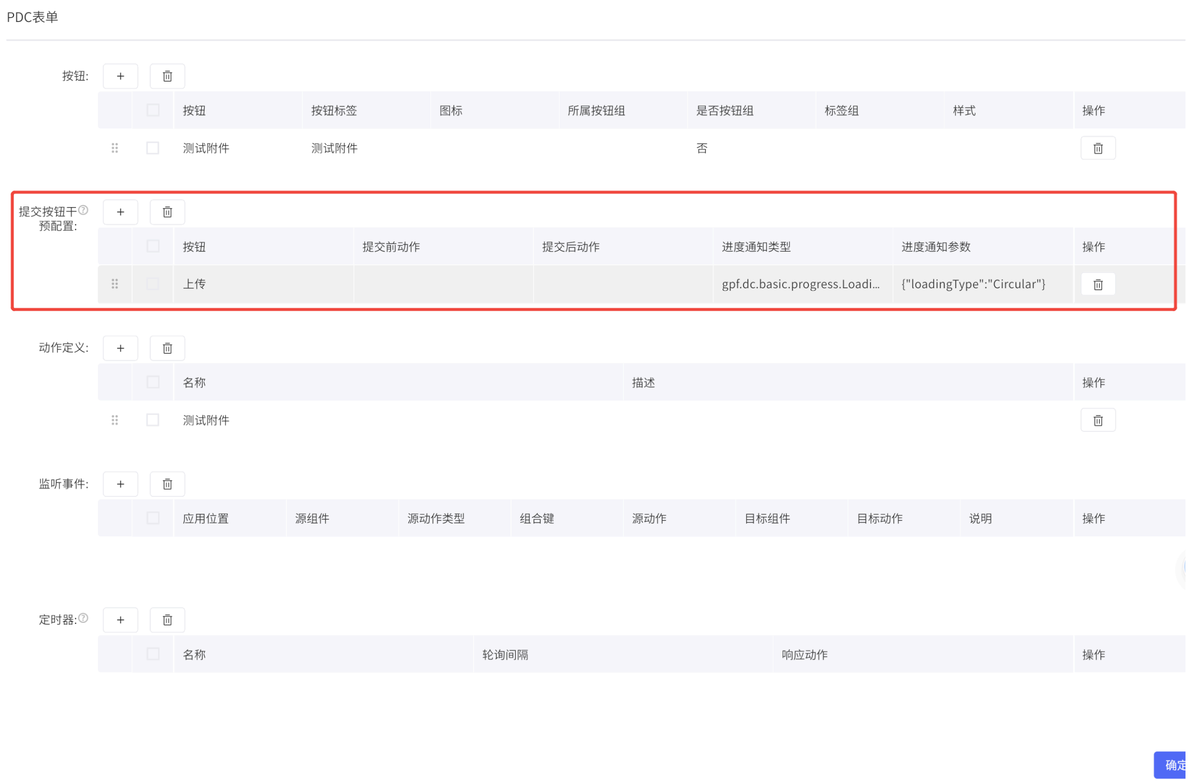
表单提交按钮进度干预

表单提交按钮进度干预，可使用平台带的通知对象，或自定义构建进度通知对象实现表单提交过程中的交互信息。

使用说明



PDC表单

按钮:

	□	按钮	按钮标签	图标	所属按钮组	是否按钮组	标签组	样式	操作
⠿	□	测试附件	测试附件			否			🗑

提交按钮干预配置:

	□	按钮	提交前动作	提交后动作	进度通知类型	进度通知参数	操作
⠿	□	上传			gpf.dc.basic.progress.Loadi...	{"loadingType":"Circular"}	🗑

动作定义:

	□	名称	描述	操作
⠿	□	测试附件		🗑

监听事件:

	□	应用位置	源组件	源动作类型	组合键	源动作	目标组件	目标动作	说明	操作

定时器:

	□	名称	轮询间隔	响应动作	操作

动作使用实例:

```
package cell.study.progress;

import java.util.LinkedHashMap;
import java.util.Map;

import bap.cells.Cells;
import cell.CellIntf;
import cmn.anotation.ClassDeclare;
import cmn.dto.Progress;
import cmn.enums.ProgressConfirmOperation;
import cmn.enums.ProgressMessageType;
import cmn.util.ProgressUtil;
import gpf.dc.action.intf.BaseActionIntf;
import gpf.dc.action.param.BaseActionParameter;
@ClassDeclare(label = "进度通知代码样例"
,what="进度通知代码样例"
, why = ""
, how = ""
,developer="陈晓斌"
,version = "1.0"
,createTime = "205-01-17"
,updateTime = "205-01-17")
public interface IActionStudyProgress<T extends BaseActionParameter> extends CellIntf,BaseActionIntf<T>
{

    public static IActionStudyProgress get(){ return Cells.get(IActionStudyProgress.class);}

    @Override
    default Object execute(T input) throws Exception {
        Progress prog = input.getRtx().getProgress();
        //发送消息
        ProgressUtil.setMessage(prog, "发送一条进度消息", true);
        ProgressUtil.sendProcess(prog, 10, "发送一条带进度的消息", true);
        //弹出确认框,注意要考虑流程纯后台提交，上下文没有携带进度通知对象的处理场景，默认null是返回ProgressConfirmOperation.YES的处理结果
        int option = ProgressUtil.showConfirmDialog(prog,"请确认是否继续进行？", "请选择", ProgressConfirmOperation.YES);
        if(option == ProgressConfirmOperation.YES.getValue()) {
            //弹出消息框
            ProgressUtil.showMessageDialog(prog,"你选择了是！\n你选择了是！\n你选择了是！\n你选择了是！\n你选择了是！\n你选择了是！\n你选择了是！\n你选择了是！\n", "", ProgressMessageType.success);
        }else {
            ProgressUtil.showMessageDialog(prog,"你选择了否！", "", ProgressMessageType.info);
        }
        //弹出确认框
        option = ProgressUtil.showConfirmDialog(prog,"请确认是否继续进行？", "请选择1", ProgressConfirmOperation.YES,ProgressMessageType.info);
        if(option == ProgressConfirmOperation.YES.getValue()) {
            //弹出消息框
            ProgressUtil.showMessageDialog(prog,"你选择了是！", "", ProgressMessageType.warning);
        }else {
            //弹出消息框
            ProgressUtil.showMessageDialog(prog,"你选择了否！", "", ProgressMessageType.error);
        }
        //构建自定义
        Map<String,Object> userObject = new LinkedHashMap<>();
        userObject.put("选项", option);
        //发送自定义对象到进度通知
        ProgressUtil.sendDataFrame(prog,userObject);
        //断言用户是否执行取消操作，确认取消将抛出UserCancelException
        ProgressUtil.assertCancel(prog);
        return null;
    }

}
```

自定义进度通知实现

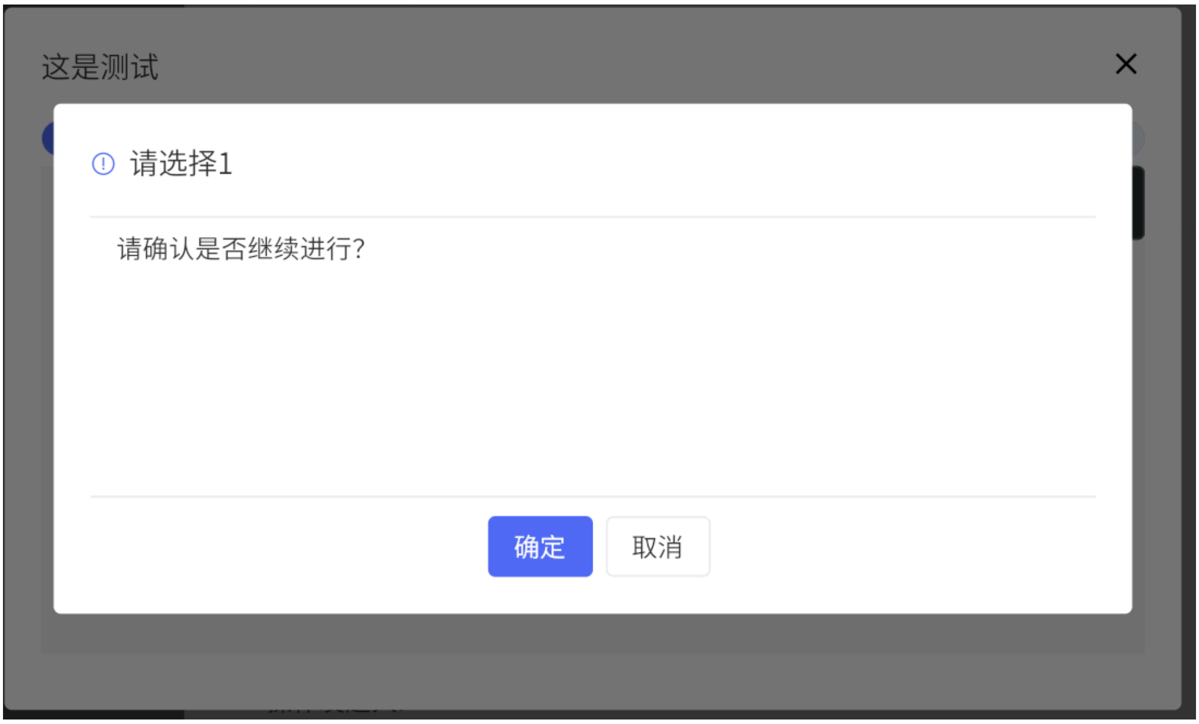
自定义通知实现需要继承 AbsProgressBuilder 实现buildProgress和buildEditPanel方法

buildProgress为进度通知对象的构建接口

buildEditPanel为进度通知参数的面板配置接口

具体示例可见：TextAreaProgressBuilder、LoadingProgressBuilder

效果实例图



这是测试

请选择1

请确认是否继续进行?