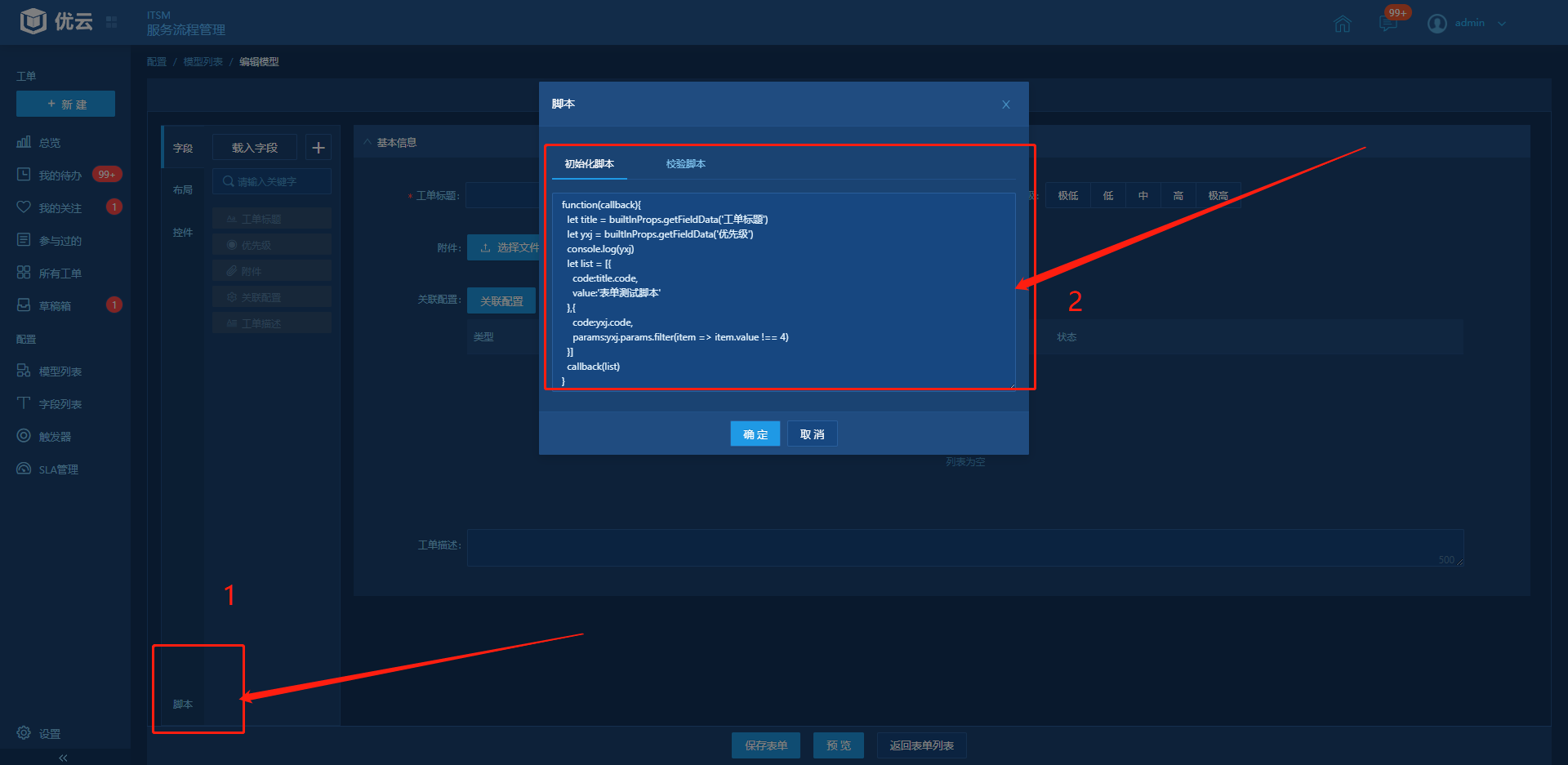
ITSM全局脚本指南

1. 配置入口

    R16版本增加全局配置脚本功能，全局脚本包含 初始化数据 和 提交前校验功能两部分

   为了编写方便提供了一个内置的对象来协助获取一些基本属性(请勿修改)

   设置脚本的时候需要了解 [ITSM字段基本属性](http://kb.uyunsoft.cn/pages/viewpage.action?pageId=80782950) ， 以保证传入的值是可解析的



1. 脚本规范

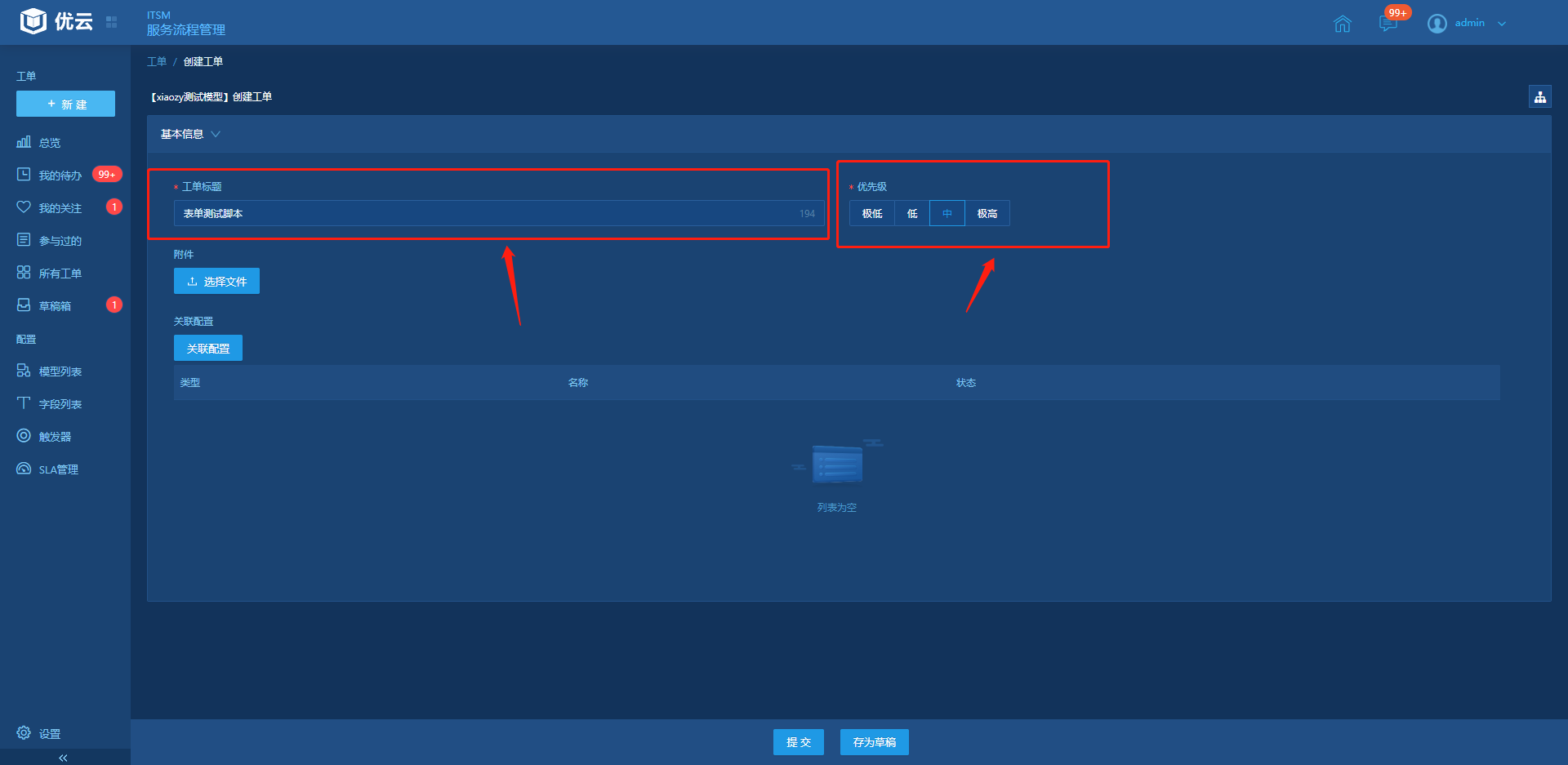
**脚本规范**

|  |
| --- |
| /\*\*   \*   \* @param {function} callback 回调函数   \* @param {id} id 回调函数对应的id   \*/  function(callback , id){    ...dosomething      const list = [{      code:'title',   //要设置字段的code      value:'测试',   //给字段设置的value值，不同字段不同类型      hidden:false,   //该字段是否隐藏      errorMes:'错误信息' //该字段不通过校验的错误信息（仅在校验脚本中进行解析）      params:[{        value:'',        label:''      }],            //下拉选项list ， label：下拉项名称 ， value：下拉项code    }]    callback(list , id)  } |

| **名称** | **类型** | **描述** | **入参** | **出参** |
| --- | --- | --- | --- | --- |
| builtInProps.values | object |  | 无 | /\*\* \* 当前表单的值 \* 如 ： 工单标题 ： 测试表单 ， 优先级 ： 低 \*/  {    title:'测试表单',    urgentLevel:'2' } |
| builtInProps.getFieldData | function |  | 类型： string  参数为字段的名称  如：'工单标题' | |  | | --- | | /\*\*   \*  当前字段的基本属性 示例为优先级字段属于单选类型   \*  {string} code ： 字段的编码   \*  {string} type ： 字段类型   \*  {array} params ： 字段的下拉项 （下拉，多选，单选，级联）,   \*   label:string 下拉名称  value:string 下拉字段的value   \*/  {    code:'urgentLevel',    type:'singleSel',    params:[{      label:'低',      value:'2'    }]  } | |
| builtInProps.getFieldsData | function |  | array  字段的名称列表  如：['工单标题' ，'优先级'] | |  | | --- | | /\*\*   \* {array} 入参字段列表的基本属性   \*/  [{    code:'title',    type:'singleRowText',    param:[]  },{     code:'urgentLevel',     type:'singleSel',     params:[{      label:'低',      value:'2'    }]  }] | |
| builtInProps.createConfirm | function | 当请求时间过长且在请求中不能操作表单的时候，可以调用该方法，出现一个”脚本正在执行，请耐心等待“弹框提示，阻止用户修改数据  请求完成之后脚本执行完毕会自动关闭弹框 | 无 | 无 |
| builtInProps.add | function | add方法可以坚守一个函数，用来监听message信息，并对数据进行处理 | function |  |

1. 初始化脚本
2. **案例1**

进入页面的时候设置工单标题为表单测试脚本，去掉优先级的高的选项，隐藏工单描述



**代码示例** 展开源码

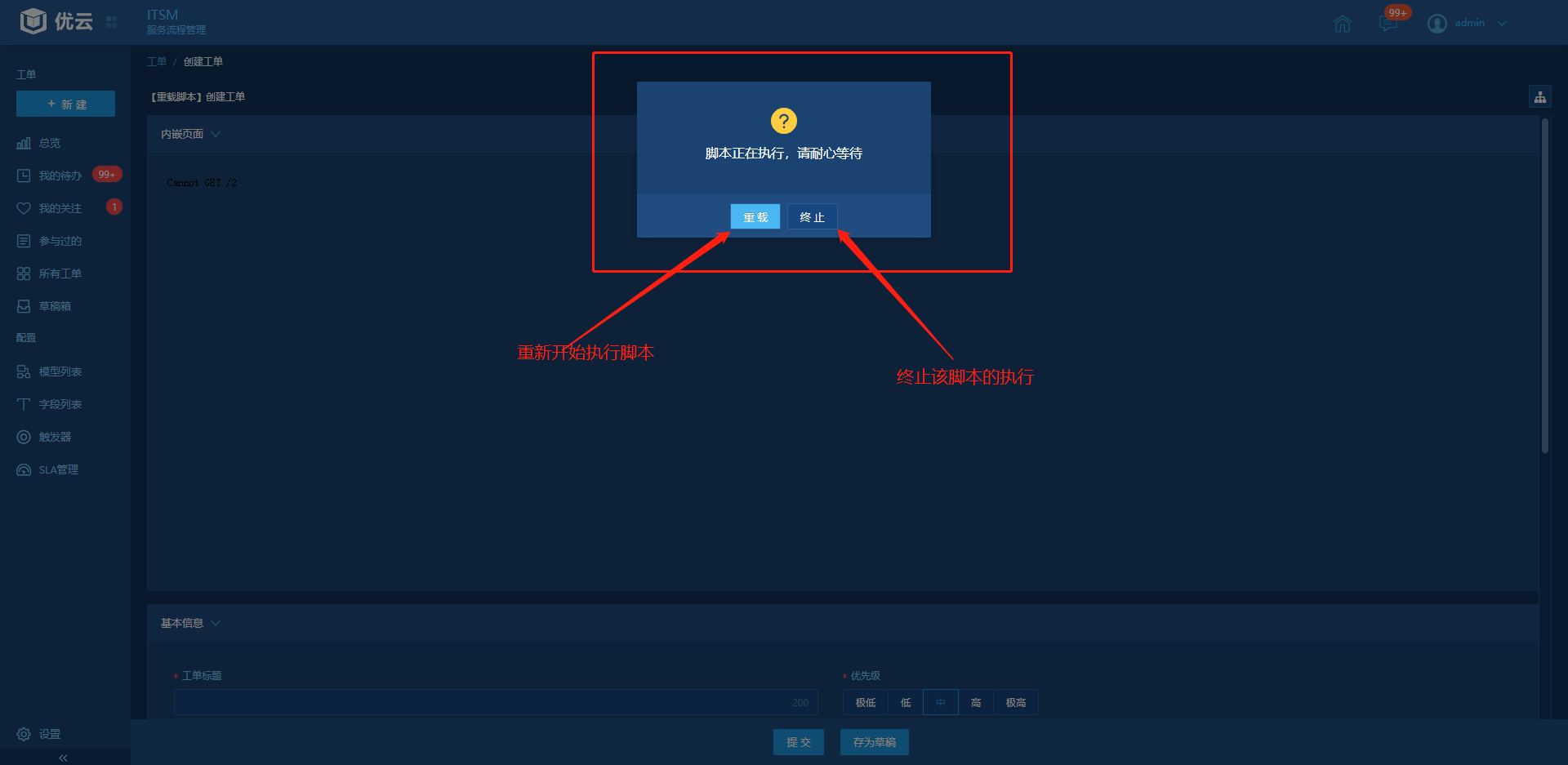
1. **案例2**

场景 ：进入工单详情页面调用ajax请求，获取返回的数据然后进行表单的修改，如果ajax返回时间过长可以调用 builtInProps.createConfirm() 弹框进行提示，调用callback后回自动关掉弹框

通过setTimeout进行场景模拟，5s以后修改工单标题为1111

重载：重新执行一次该脚本

终止：终止该脚本的执行



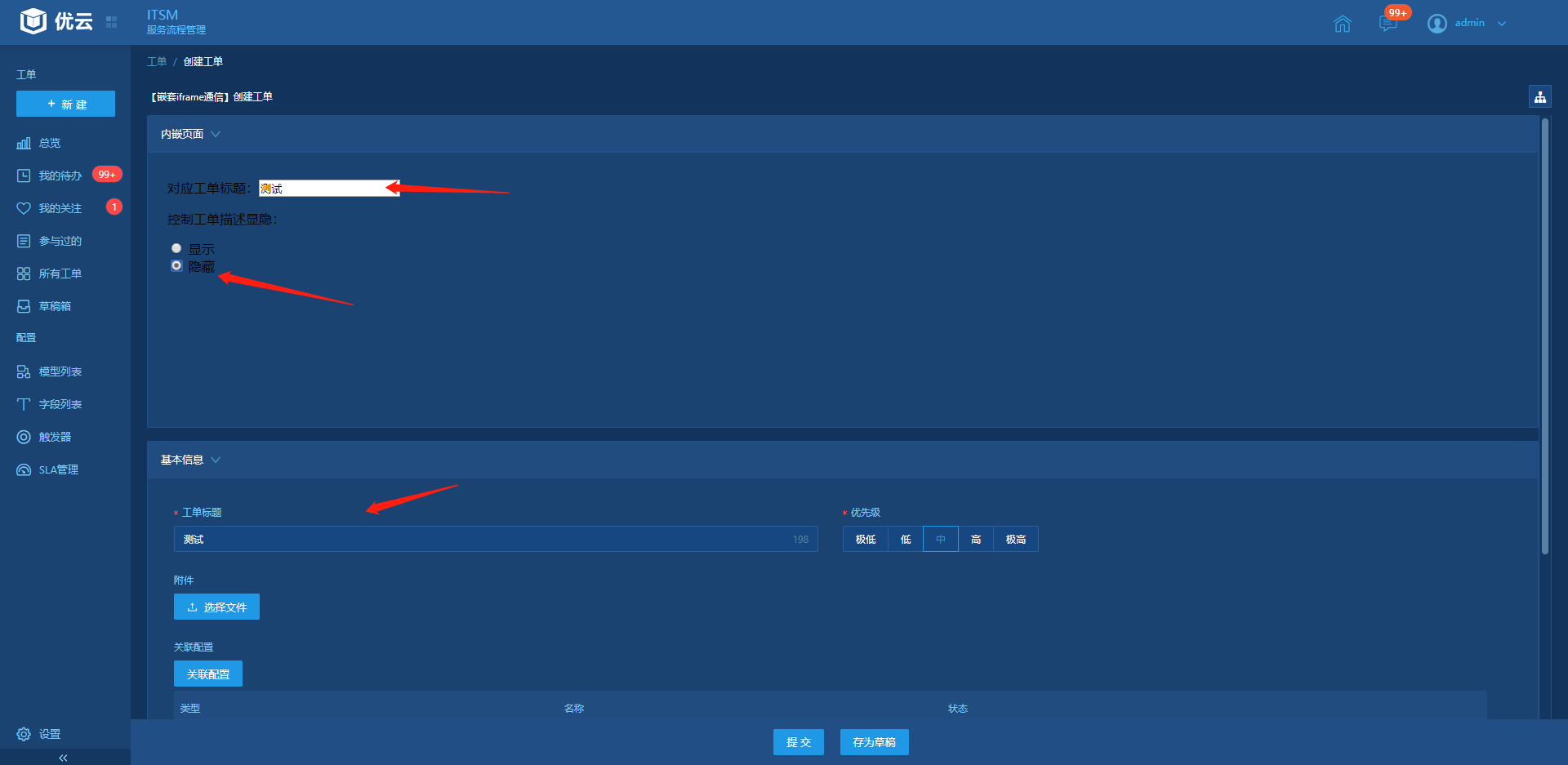
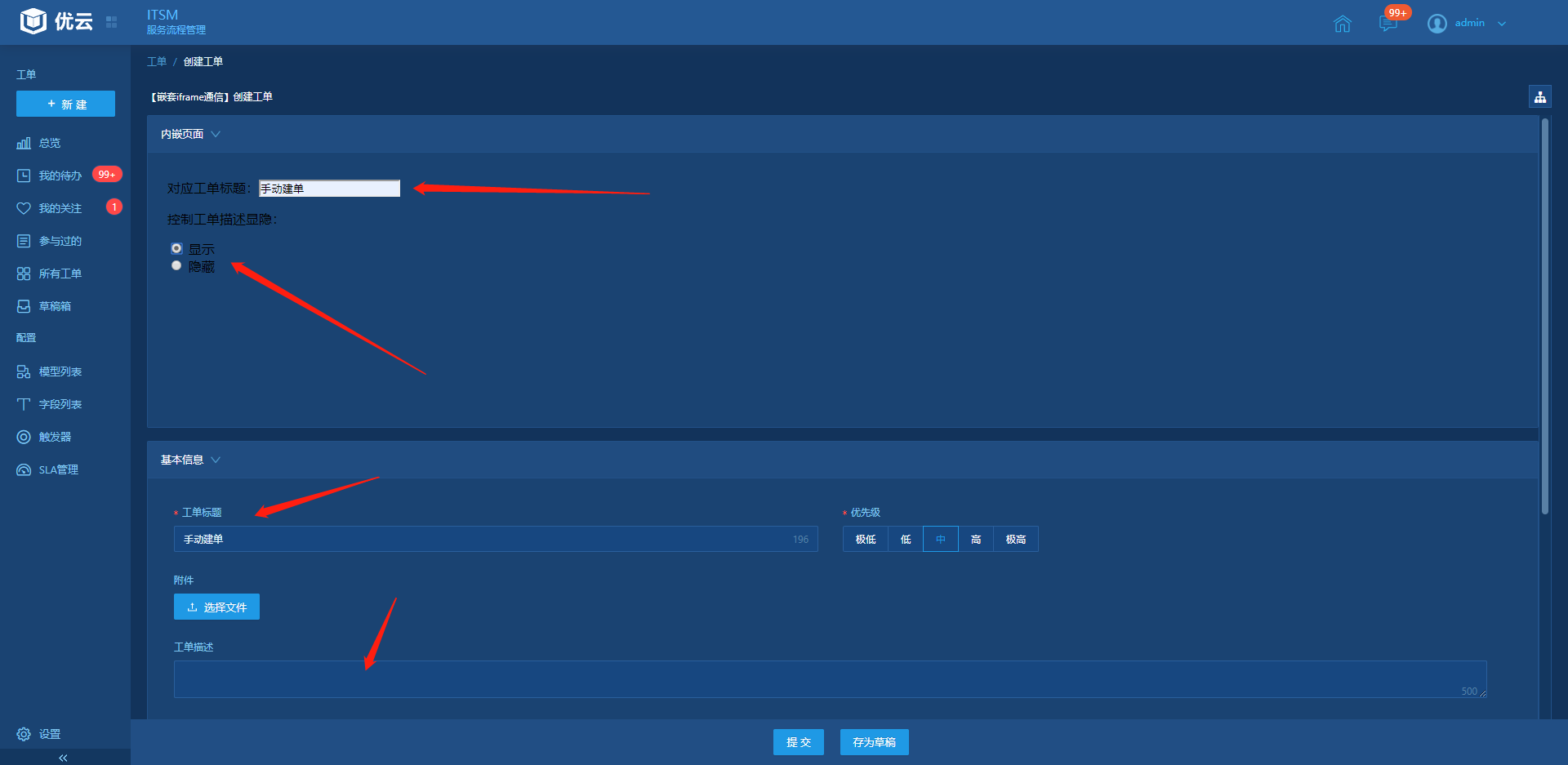
**代码示例** 折叠源码

|  |
| --- |
| function(callback , id){    builtInProps.createConfirm()    setTimeout(() => {      let list = [{        code:'title',        value:'1111',      }]      callback(list , id)    } , 5 \* 1000)  } |

1. **案例3**

监听iframe控件中嵌套的页面，iframe控件中的页面进行某些行为的时候，应该itsm表单中对应的值

iframe控件页面 demo  ： [index.html](http://kb.uyunsoft.cn/download/attachments/80780148/index.html?version=1&modificationDate=1564383126000&api=v2)

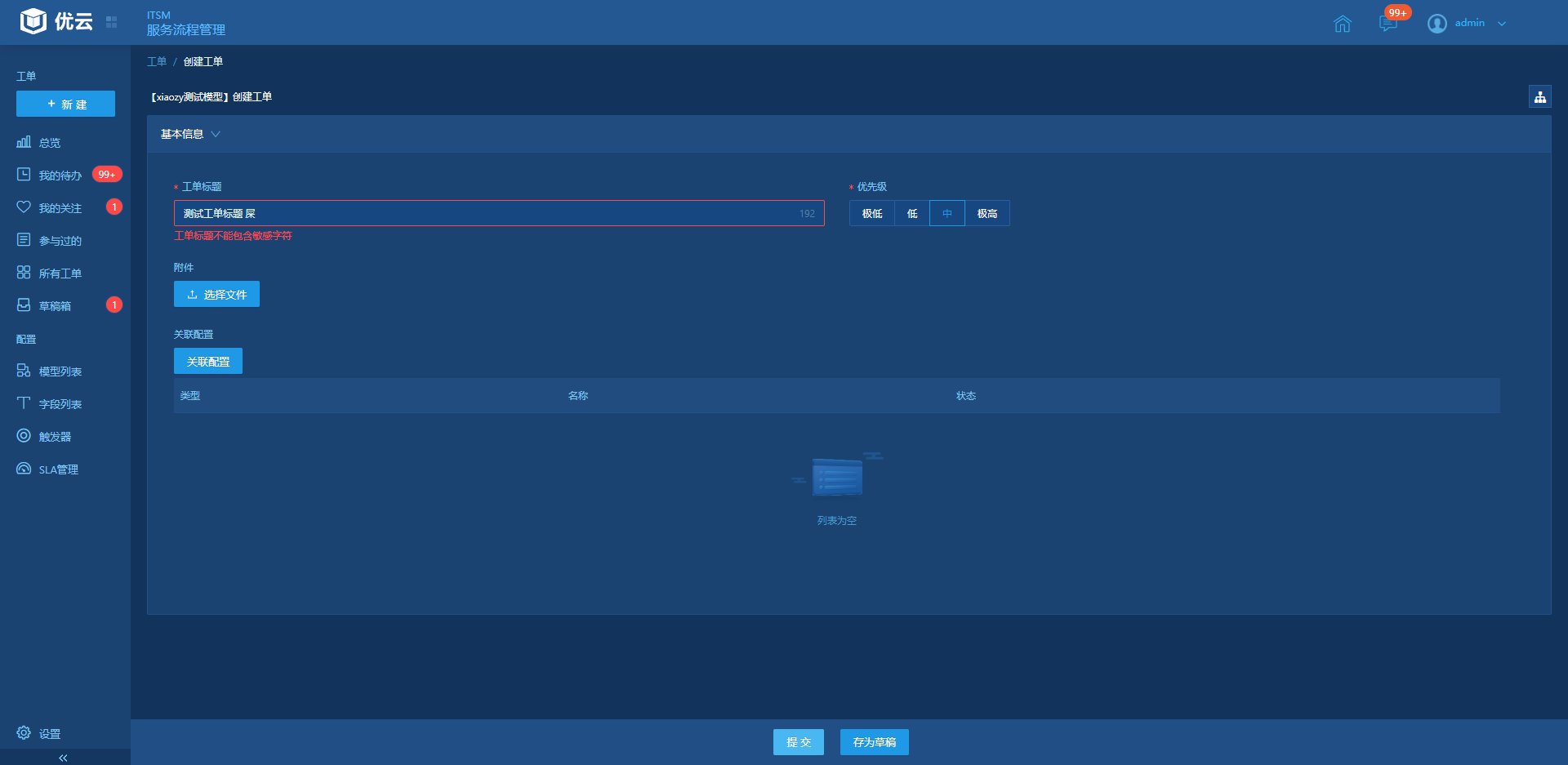


**代码示例** 折叠源码

|  |
| --- |
| function(callback , id){    /\*\*     \* 对监听到的数据进行处理，然后调用callback来修改itsm表单的值     \* 判断不一定要这样写，具体写法需要看ifrmae控件发送的信息格式     \* 我发送的信息格式为 ： window.parent.postMessage({source: 'itsm', action:"script", data:data} , '\*')     \*/    const setFields = (e) => {      if(e.data && e.data.source === 'itsm' && e.data.action === 'script'){        callback(e.data.data , id)      }    }    /\*\*     \* 注册监听事件     \* iframe控件需要发送信息 发送的方法为 window.parent.postMessage ，不懂请百度搜索postmessage通信     \*/    builtInProps.add(setFields)  } |

1. 校验脚本
2. **案例1**

工单标题不能包含 '屎'，'去死' ，’fuck‘ 不文明字符



**代码示例** 折叠源码

|  |
| --- |
| function (callback , id) {    let title = builtInProps.getFieldData('工单标题')    let list = ['屎','去死','fuck']    let falt = \_.some(list , item => {      return values[title.code].indexOf(item) !== -1    })    if(falt){      callback([{        code:'title',        errorMes:'工单标题不能包含敏感字符'      }] , id)    }  } |