



Provider

dec.provider.schedule

registerTaskStep(config[, index])

configObjecttextvaluecardTypesshortcut

indexNumbersplice

```
//
//
BI.config("dec.provider.schedule", function (provider) {
  provider.registerTaskStep({
    text: BI.i18nText(""),
    value: "plugin_step",
    cardType: "dec.schedule.task.plugin"
  }, 1);
});
```

```

!(function () {
  var Plugin = BI.inherit(BI.Widget, {

    props: {
      baseCls: ""
    },

    render: function () {
      return {
        type: "bi.label",
        text: ""
      };
    },

    /**
     *
     * @param callback
     */
    validation: function (callback) {
      callback(true);
    },

    /**
     *
     * this.model.currTaskBI.extend
     * @returns {{}}
     */
    getValue: function () {
      return {};
    }
  });
  BI.shortcut("dec.schedule.task.plugin", Plugin);
})();

```

modelthis.model.currTask

☒ 基本设置 —
 ☒ 新增步骤 —
 ☐ 调度对象 —
 ☐ 调度周期 —
 ☐ 文件处理

[< 上一步](#) | [下一步 >](#)

新增步骤

registerDispatcher(dispatcher[, handings])

dispatcherObjecttextvaluecardTypesshortcut
handingsArrayregisterHandingWay

```
//
//
BI.config("dec.provider.schedule", function (provider) {
    provider.registerDispatcher({
        value: "plugin",
        text: "",
        cardType: "dec.schedule.task.dispatcher.plugin"
    }, []);
});
```

getValuevalidation



registerHandingWay(config, scopes)

configObjecttextvalue**actionName**cardTypesshortcutactionsactionselectedforceSelectedtrue
scopesArray()registerDispatcherdispatchervalue

```
//
//
BI.config("dec.provider.schedule", function (provider) {
    provider.registerHandingWay({
        text: "",
        value: "com.fr.xxxx", // actionName
        cardType: "dec.schedule.task.file.handling.plugin",
        actions: [] // action
    }, [DecCst.Schedule.TaskType.DEFAULT, DecCst.Schedule.TaskType.REPORT, DecCst.Schedule.TaskType.BI]);
});
```

runTyperunTyperunType

```

!(function () {
    var Plugin = BI.inherit(BI.Widget, {

        props: {
            baseCls: ""
        },

        render: function () {
            return {
                type: "bi.label",
                text: ""
            };
        },

        /**
         *
         * @returns {boolean}
         */
        validation: function () {
            return true;
        },

        /**
         *
         * outputActionList
         * @returns {{}}
         */
        getValue: function () {
            return {};
        }
    });
    BI.shortcut("dec.schedule.task.file.handling.plugin", Plugin);
})();

```

registerTaskAttached(config, scopes)

configObjecttextvalue

scopesArrayregisterDispatcherdispatchervalue

```

//
//
BI.config("dec.provider.schedule", function (provider) {
    provider.registerTaskAttached({
        value: 11,
        text: "pluginText"
    }, [DecCst.Schedule.TaskType.REPORT, DecCst.Schedule.TaskType.BI]);
});

```

2019.10.10provider

2019.11.12