

- [Provider](#)
- -
 -
 -
 -
-

Provider

dec.provider.data.set

registerDataSetType(config)

configObjecttextvaluecardTypesshortcut

```
BI.config("dec.provider.data.set", function (provider) {  
  provider.registerDataSetType({  
    value: "plugin",  
    text: "",  
    cardType: "dec.data.set.type.plugin"  
  });  
});
```

```

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

    _store: function () {
      return BI.Models.getModel("dec.model.data.set.type.plugin", this.options);
    },

    render: function () {
      var self = this, o = this.options;
      return {};
    },

    /**
     *
     * datasetNameddatasetType
     * datasetData
     * @returns {}
     */
    getValue: function () {
      return {

      };
    },

    /**
     *
     * @returns {boolean}
     */
    validation: function () {
      return true;
    }
  });
  BI.shortcut("dec.data.set.type.plugin", Plugin);
})();

```

model

```

!(function () {
  var Model = BI.inherit(Fix.Model, {

    // dataSetNameableSave
    context: ["dataSetName", "ableSave"],

    state: function () {
      return {};
    },

    computed: {},

    actions: {}
  });
  BI.model("dec.model.data.set.type.plugin", Model);
})();

```

dec.data.set.previewcontextpreviewAble, previewedDataSet



1. BI.Services.getService("dec.service.data.set").isSupportDataSet(type);
2. BI.Services.getService("dec.service.data.set").getDefaultValueByType(type);
3. sql

```
// getParameters
Dec.Utils.getDataSetParameters(dataSet, function (res) {
    newParameters = BI.Services.getService("dec.service.data.set").getParameters(res.data,
oldParameters);
});
```

4. BI.Services.getService("dec.service.data.set").createParameterValueItem(param, cb);
param{type: "", value: "", name: ""}cb
5. BI.Services.getService("dec.service.data.set").showParametersPopover(parameters, cb)
parametersnamevaluecb