PanelCollection API

- GeneralPanelOptions
- CssRule
- ListOptions
- ParameterValidation
- ParameterDisplayRule
- ParameterValueLoaderclass
- DynamicParameterOptionsParameterValueLoader
- FieldObjectclass

GeneralPanel Options

GeneralPanelOptions class

public void setSaveButton(boolean saveButton) public void setSaveButtonOptions(Map<String, Object> opts) public void setSaveText(String saveText)	**setSaveButton**true**setSaveButtonOptions**type BUTTON**setSaveText**
public void setExpandable(boolean expandable) public void setExpanded(boolean expandable)	
public void setShowName(boolean showTitle)	true
public void setCssRules(Set<CssRule> cssRules)	htmlCSS

CSSRule

PanelCollection APICssRuleYellowfin**CssRuleImpl**CSS

```
div.styleExampleCell {
    border: none;
    color: #666666;
}
```

CSSYellowfin

CssDeclaration

```
border: none;
```

Yellowfin**CssDeclarationImpl**

```
Parameter inputField = new ParameterImpl();
inputField.setName("Example Param");
inputField.setProperty("PARAM_PROPERTY");
inputField.setInputType(InputType.TEXTBOX);

CssRule cssRule = new CssRuleImpl("input", false);
cssRule.addDeclaration(new CssDeclarationImpl("height", "21px"));
cssRule.addDeclaration(new CssDeclarationImpl("padding", "5px"));
cssRule.addDeclaration(new CssDeclarationImpl("font-size", "16px"));
cssRule.addDeclaration(new CssDeclarationImpl("resize", "none"));
cssRule.addDeclaration(new CssDeclarationImpl("color", "#666666"));
cssRule.addDeclaration(new CssDeclarationImpl("border", "1px solid #e4e4e4"));
Set<CssRule> cssRules = new HashSet<>();
cssRules.add(cssRule);
inputField.setCssRules(cssRules);
```

CSS

```
input {
    height: 21px;
    padding: 5px;
    font-size: 16px;
    resize: none;
    color: #666666;
    border: 1px solid #e4e4e4;
}
```

ListOptions

classTEXTBOX

javadoc

ParameterValidation

class

javadoc

ParameterDisplayRule

//ParameterDisplayRule

1

SOURCE0nullTABLE_NAME

```
Parameter p = new ParameterImpl();
p.setName("Table Name");
p.setProperty("TABLE_NAME");
p.InputType(InputType.SELECT);
p.addDisplayRule(new ParameterDisplayRule("AND", "SOURCE", new Object[] { null, 0 }, true, null));
```

```
if(SOURCE != null && SOURCE != 0) showParameter();
```

2

SOURCEnull0TABLE_NAMEbooleanfalse

```
p.addDisplayRule(new ParameterDisplayRule("AND", "SOURCE", new Object[] { null, 0 }, false, null));
```

```
if(SOURCE == null || SOURCE == 0) showParameter();
```

class

public ParameterDisplayRule(String logic, String property, Object[] vals, boolean negative, PropertyLocation location)

- **logic**ParameterDisplayRuleLogicANDOR3logicAND
  Rule1Result && Rule2Result && Rule3Result
  Logic

- **property**SOURCE
- **vals**fOR
- **negative**vals
  If vals = {null, 0}

negative	
true	property != null && property != 0
false	property == null || property == 0

- **location**PropertyLocation/nullParamterDIsplayRule

ParameterValueLoader

ParameterValueLoaderclass

- p.setProperty("view");

- p.setEvent("viewChanged");

- p.addEventData("fieldsMap", allFieldsMap);
  p.addEventData("sourceId", getStepOption("SOURCE_ID"));

- getUpdatedPossibleValues()generateDynamicParameterYellowfin
  pvl.getDynamicParameterOptions.setValues(pvl.getUpdatedPossibleValues());
  pvl.generateDynamicParameters();

  ⚠ generateDynamicParameters()

-

ValueDependencies

ValueDependencies

```
datasourceParam.setProperty("datasource");
datasourceParam.setEvent("datasourceChanged");

viewParam.setProperty("view");
// Make view dependent on the data source
viewParam.addValueDependency("datasource");
```

dataSourceChangedYellowfin

3

1. setEvent()ParameterValueLoader
2. ID
3. APIValueDependent

EventData

```
cityParam.setProperty("city");
cityParam.setEvent("cityChanged");
cityParam.addEventData("fieldUUIDS", fieldUuids);

p.addEventParameter("country", new PropertyLocation("MyPanel", "addressSection"));
p.addEventParameter("region", new PropertyLocation("MyPanel", "addressSection"));
```

class

protected Map<String, Object> changedValues	changedValues

protected Map<String, Object> eventData	1changedValues
	
	{
	"datasource": 1
	}
	
	eventData
	
	{
	"view": objects set by the developer
	}

protected List<String> events	
	`if(events.contains("cityChanged")) {`
	`    // do stuff`
	`}`

protected DynamicParameterOptions response	ParameterValueLoaderDynamicParameterOptions

2

public void generateDynamicParameters()	DynamicParameterOptions
public Map<String, List<CustomValue<?>>> getUpdatedPossibleValues()	CustomValue

public final byte[] getFile(Integer fileId)	FileUploadParameterFileUploaderParameterYellowfin/
public final String getText(Integer textId)	YellowfinCLOBsgetFile

DynamicParameterOptions

ParameterValueLoader6

public void addPanel(String dynamicKey, ParameterPanel panelObject)	 dynamickey
public void addSection(String dKey, String panelKey, ParameterSection sec)	dKeypanelKey
public void addParameter(String dKey, String panelKey, String secKey, Parameter parameter)	/dKeypanelKeysecKey
public void addValue(String property, CustomValue<?> value)	DynamicParameterOptions
public void setValues(Map<String, List<CustomValue<?>>)	DynamicParameterOptions
public void addKeyToRemove(String key)	//

FiledObject

classclass

public FieldObject(String id, String keyType, String name, FieldObjectDataType dataType)

- **id**
- **keyType**
- **name**
- **dataType**FeildObjectDataTypeString