- 1.
- 3.
- 4.
- 5./ API
- 6.
- 7. !
    - a.
    - b.
    - c.
    - d.
    - e. Felix
    - f.
- 8.

Bean Shell

1.

All (18) | Published (17) | Draft (1) | Trash (2)

2.

Datalist Filter Type

3.

1.
2.

4.

All (18) | Published (17) | Draft (1) | Trash (2)

5./ API

6.

Joget Workflow Source Code

Macbook ProJoget5.0.1

```
- Home
  - joget
    - plugins
    - jw-community
      -5.0.1
```

“”“jw-community”Joget Workflow

“plugins”maven

```
cd joget/plugins/
~/joget/jw-community/5.0.1/wflow-plugin-archetype/create-plugin.sh org.joget hyperlink_options_filter 5.0.1
```

shellmaven

```
Define value for property 'version':  1.0-SNAPSHOT: : 5.0.0
[INFO] Using property: package = org.joget
Confirm properties configuration:
groupId: org.joget
artifactId: hyperlink_options_filter
version: 5.0.0
package: org.joget
Y: : y
```

“BUILD SUCCESS”“plugins”“hyperlink_options_filter”

IDEmaven NetBeans

7. !

a.

“org.joget”“HyperlinkOptionsFilter”**org.joget.apps.datalist.model.DataListFilterTypeDefault** Datalist Filter Type

b.

AppPluginUtil.getMessagei18nMESSAGE_PATH

Implementation of all basic abstract methods

```
package org.joget;

import org.joget.apps.app.service.AppPluginUtil;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.datalist.model.DataListFilterTypeDefault;

public class HyperlinkOptionsFilter extends DataListFilterTypeDefault {
    private final static String MESSAGE_PATH = "message/HyperlinkOptionsFilter";

    public String getName() {
        return "Hyperlink Options Filter Type";
    }

    public String getVersion() {
        return "5.0.0";
    }

    @Override
    public String getLabel() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.HyperlinkOptionsFilter.pluginLabel", getClassName(),
MESSAGE_PATH);
    }

    @Override
    public String getDescription() {
        //support i18n
        return AppPluginUtil.getMessage("org.joget.HyperlinkOptionsFilter.pluginDesc", getClassName(),
MESSAGE_PATH);
    }

    public String getClassName() {
        return this.getClass().getName();
    }

    public String getPropertyOptions() {
        return AppUtil.readPluginResource(getClass().getName(), "/properties/hyperlinkOptionsFilter.json",
null, true, MESSAGE_PATH);
    }
}
```

UlgetPropertyOptions “/properties/hyperlinkOptionsFilter.json”“hyperlink_options_filter / src / main”“resources / properties”“properties”“hyperlinkOptionsFilter.json”

“@@ message.key @@”i18n

```
[{
    title : '@@HyperlinkOptionsFilter.config@@',
    properties : [{
        name:'defaultValue',
        label:'@@HyperlinkOptionsFilter.defaultValue@@',
        type:'textfield'
    },
    {
        name : 'showLabel',
        label : '@@HyperlinkOptionsFilter.showLabel@@',
        type : 'checkbox',
        options : [{
            value : 'true',
            label : ''
        }]
    },
    {
        name : 'displayFull',
        label : '@@HyperlinkOptionsFilter.displayFull@@',
        type : 'checkbox',
        value : 'true',
        options : [{
            value : 'true',
            label : ''
        }]
    },
    {
        name : 'showCount',
        label : '@@HyperlinkOptionsFilter.showCount@@',
        type : 'checkbox',
        value : '',
        options : [{
            value : 'true',
            label : ''
        }]
    },
    {
        name : 'options',
        label : '@@HyperlinkOptionsFilter.options@@',
        type : 'grid',
        columns : [{
            key : 'value',
            label : '@@HyperlinkOptionsFilter.value@@'
        },
        {
            key : 'label',
            label : '@@HyperlinkOptionsFilter.label@@'
        }]
    }]
},
{
    title : '@@HyperlinkOptionsFilter.chooseOptionsBinder@@',
    properties : [{
        name : 'optionsBinder',
        label : '@@HyperlinkOptionsFilter.optionsBinder@@',
        type : 'elementselect',
        options_ajax : '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.
FormLoadOptionsBinder',
        url : '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
    }]
}]
```

getTemplategetQueryObjectgetTemplate

```
    public String getTemplate(DataList datalist, String name, String label) {
        PluginManager pluginManager = (PluginManager) AppUtil.getApplicationContext().getBean("pluginManager");
        Map dataModel = new HashMap();
```

```
        dataModel.put("element", this);
        dataModel.put("name", datalist.getDataListEncodedParamName(DataList.PARAMETER_FILTER_PREFIX+name));
        dataModel.put("label", label);

        Map<String, String> options = getOptionMap();
        if ("true".equalsIgnoreCase(getPropertyString("showCount"))) {
            DataListBinder binder = datalist.getBinder();
            for (String key : options.keySet()) {
                DataListFilterQueryObject filter = getQueryObject(datalist, name, key);
                int count = 0;
                if (binder != null) {
                    if (filter != null) {
                        count = binder.getDataTotalRowCount(datalist, binder.getProperties(), new
DataListFilterQueryObject[]{filter});
                    } else {
                        count = binder.getDataTotalRowCount(datalist, binder.getProperties(), new
DataListFilterQueryObject[]{});
                    }
                }

                options.put(key, options.get(key) + " (" + count + ")");
            }
        }

        String value = getValue(datalist, name, getPropertyString("defaultValue"));
        dataModel.put("value", value);
        dataModel.put("options", options);

        return pluginManager.getPluginFreeMarkerTemplate(dataModel, getClassName(), "/templates
/hyperlinkOptionsFilter.ftl", null);
    }

    protected Map<String, String> getOptionMap() {
        Map<String, String> optionMap = new ListOrderedMap();

        // load from "options" property
        Object[] options = (Object[]) getProperty(FormUtil.PROPERTY_OPTIONS);
        for (Object o : options) {
            Map option = (HashMap) o;
            Object value = option.get(FormUtil.PROPERTY_VALUE);
            Object label = option.get(FormUtil.PROPERTY_LABEL);
            if (value != null && label != null) {
                optionMap.put(value.toString(), label.toString());
            }
        }
        // load from binder if available
        Map optionsBinderProperties = (Map) getProperty("optionsBinder");
        if (optionsBinderProperties != null && optionsBinderProperties.get("className") != null && !
optionsBinderProperties.get("className").toString().isEmpty()) {
            PluginManager pluginManager = (PluginManager) AppUtil.getApplicationContext().getBean
("pluginManager");
            FormBinder optionBinder = (FormBinder) pluginManager.getPlugin(optionsBinderProperties.get
("className").toString());
            if (optionBinder != null) {
                optionBinder.setProperties((Map) optionsBinderProperties.get("properties"));
                FormRowSet rowSet = ((FormLoadBinder) optionBinder).load(null, null, null);
                if (rowSet != null) {
                    optionMap = new ListOrderedMap();
                    for (FormRow row : rowSet) {
                        Iterator<String> it = row.stringPropertyNames().iterator();
                        // get the key based on the "value" property
                        String value = row.getProperty(FormUtil.PROPERTY_VALUE);
                        if (value == null) {
                            // no "value" property, use first property instead
                            String key = it.next();
                            value = row.getProperty(key);
                        }
                        // get the label based on the "label" property
                        String label = row.getProperty(FormUtil.PROPERTY_LABEL);
                        if (label == null) {
                            // no "label" property, use next property instead
```

```
                            String key = it.next();
                            label = row.getProperty(key);
                        }
                        optionMap.put(value, label);
                    }
                }
            }
        }

        if (!optionMap.containsKey("")) {
            Map<String, String> tempOptionMap = new ListOrderedMap();
            tempOptionMap.put("", AppPluginUtil.getMessage("HyperlinkOptionsFilter.all", getClassName(),
MESSAGE_PATH));
            tempOptionMap.putAll(optionMap);
            optionMap = tempOptionMap;
        }

        return optionMap;
    }

    protected DataListFilterQueryObject getQueryObject(DataList datalist, String name, String value) {
        DataListFilterQueryObject queryObject = new DataListFilterQueryObject();
        if (datalist != null && datalist.getBinder() != null && value != null && !value.isEmpty()) {
            String columnName = datalist.getBinder().getColumnName(name);
            List<String> valuesList = new ArrayList<String>();
            String query = "("+columnName+" = ? or "+columnName+" like ? or "+columnName+" like ? or
"+columnName+" like ?)";
            valuesList.add(value);
            valuesList.add(value + ";%");
            valuesList.add("%;" + value + ";%");
            valuesList.add("%;" + value);
            queryObject.setOperator(DataListFilter.OPERATOR_AND);
            queryObject.setQuery(query);
            queryObject.setValues(valuesList.toArray(new String[0]));
            return queryObject;
        }
        return null;
    }

    public DataListFilterQueryObject getQueryObject(DataList datalist, String name) {
        String value = getValue(datalist, name, getPropertyString("defaultValue"));
        return getQueryObject(datalist, name, value);
    }
```

getTemplate“hyperlinkOptionsFilter.ftl”“hyperlink_options_filter / src / main / resources / templates” FreeMaker

```
<div id="${name!}_container" style="display:none;margin:5px 0;">
    <input id="${name!}" name="${name!}" type="hidden" value="${value!?html}" />
    <#if element.properties.showLabel! == "true" >
        <label><strong>${label!?html} :</strong></label>  
    </#if>
    <#list options?keys as key>
        <a ref="${key?html}" href="${key?html}" class="<#if value! == key >active</#if>"><span><#if value! ==
key ><strong></#if>${options[key]!?html}<#if value! == key ></strong></#if></span></a>  
    </#list>
    <script type="text/javascript">
        $(document).ready(function(){
            <#if element.properties.displayFull! == "true" >
                $('#${name!}_container').insertBefore($('#${name!}_container').closest(".filters"));
            </#if>
            $('#${name!}_container').show();
            $('#${name!}_container a').click(function(){
                var value = $(this).attr("ref");
                $(this).parent().find("input").val(value);
                $(this).closest("form").submit();
                return false;
            });
        });
    </script>
</div>
```

c.

POM“commons-collections”

```
        <!-- Change plugin specific dependencies here -->
        <dependency>
            <groupId>commons-collections</groupId>
            <artifactId>commons-collections</artifactId>
            <version>3.2.1</version>
        </dependency>
        <!-- End change plugin specific dependencies here -->
```

d.

getLabelgetDescriptioni18ni18n

“hyperlink_options_filter / src / main”“resources / message”“HyperlinkOptionsFilter.properties”

```
org.joget.HyperlinkOptionsFilter.pluginLabel=Hyperlink Options
org.joget.HyperlinkOptionsFilter.pluginDesc=Show options as Hyperlink to perform filter.
HyperlinkOptionsFilter.all=All
HyperlinkOptionsFilter.config=Configure Hyperlink Options Filter
HyperlinkOptionsFilter.options=Options
HyperlinkOptionsFilter.value=Value
HyperlinkOptionsFilter.label=Label
HyperlinkOptionsFilter.chooseOptionsBinder=Choose Options Binder
HyperlinkOptionsFilter.optionsBinder=Options Binder
HyperlinkOptionsFilter.defaultValue=Default Value
HyperlinkOptionsFilter.showCount=Show Data Count?
HyperlinkOptionsFilter.displayFull=Display in full width (Above other filters)
HyperlinkOptionsFilter.showLabel=Show label?
```

e. Felix

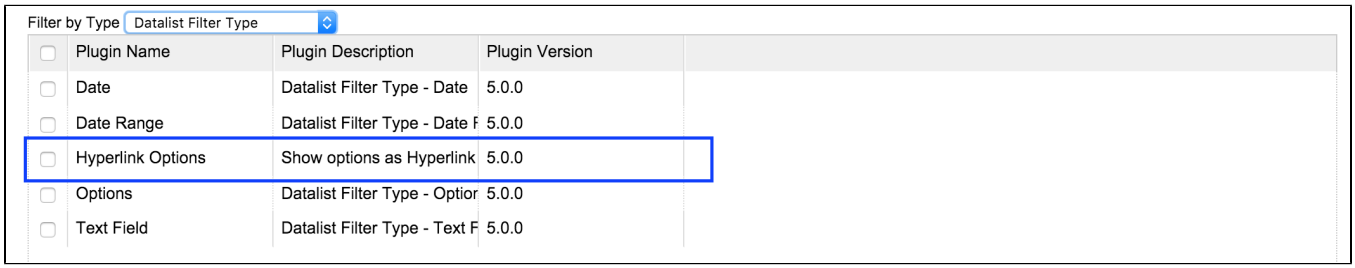
ActivatorFelix

```
    public void start(BundleContext context) {
        registrationList = new ArrayList<ServiceRegistration>();
        //Register plugin here
        registrationList.add(context.registerService(HyperlinkOptionsFilter.class.getName(), new
HyperlinkOptionsFilter(), null));
    }
```

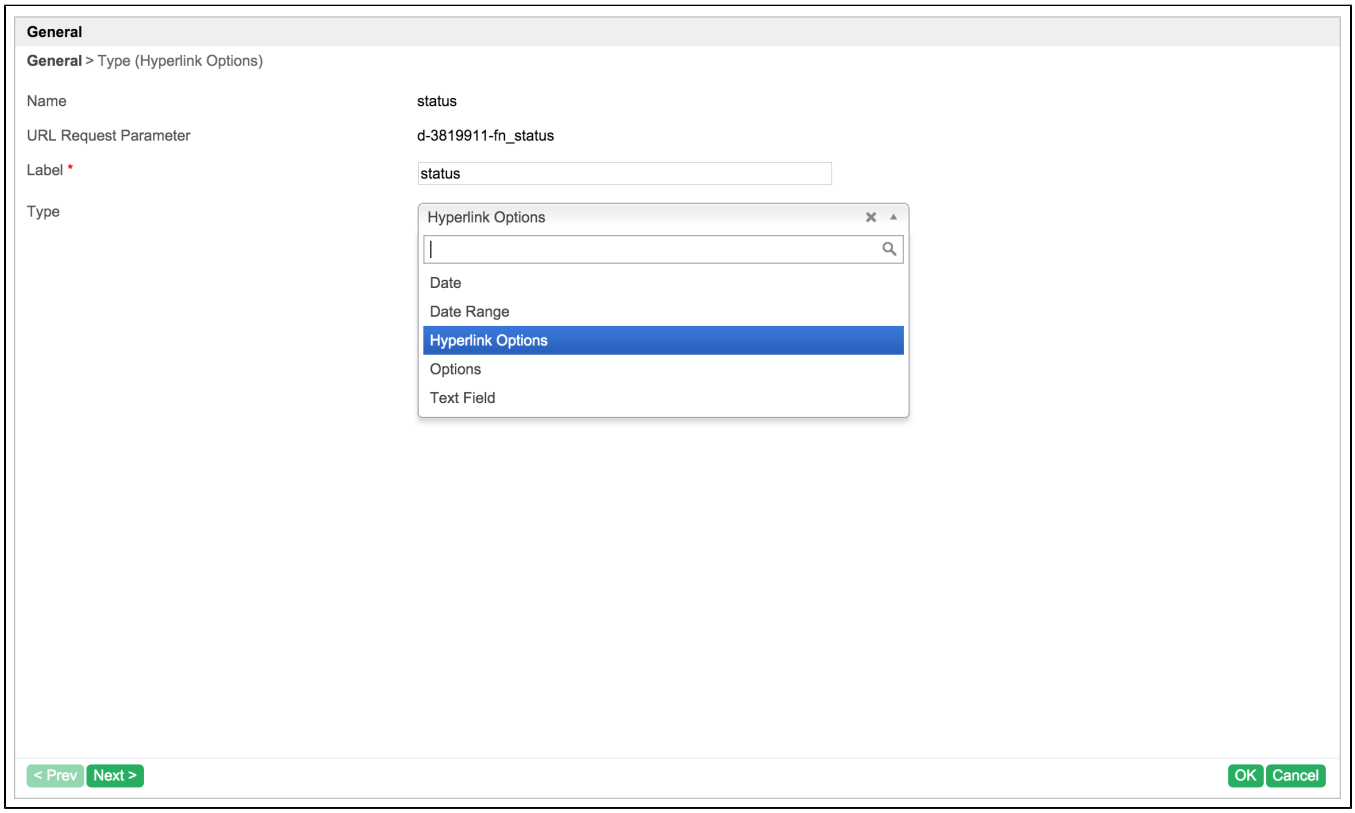
f.

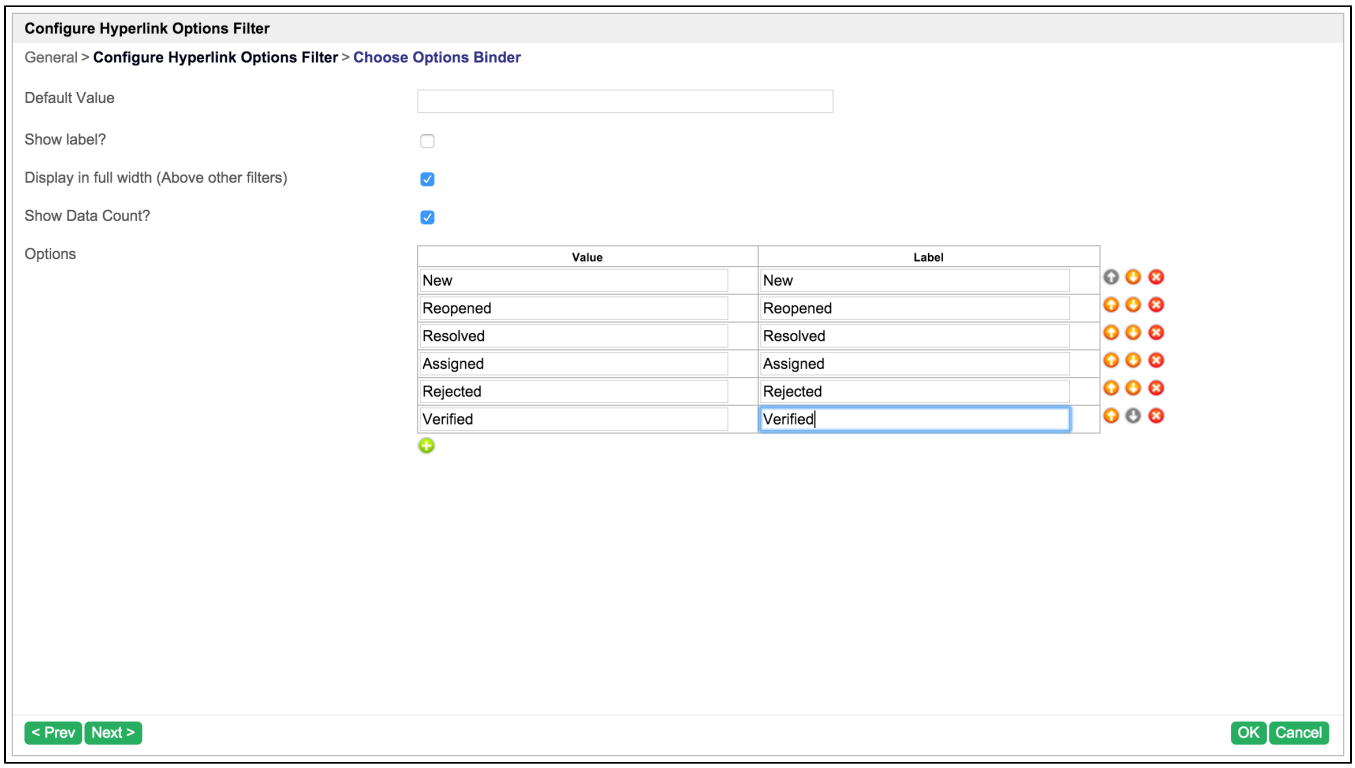
“hyperlink_options_filter / target”“hyperlink_options_filter-5.0.0.jar”

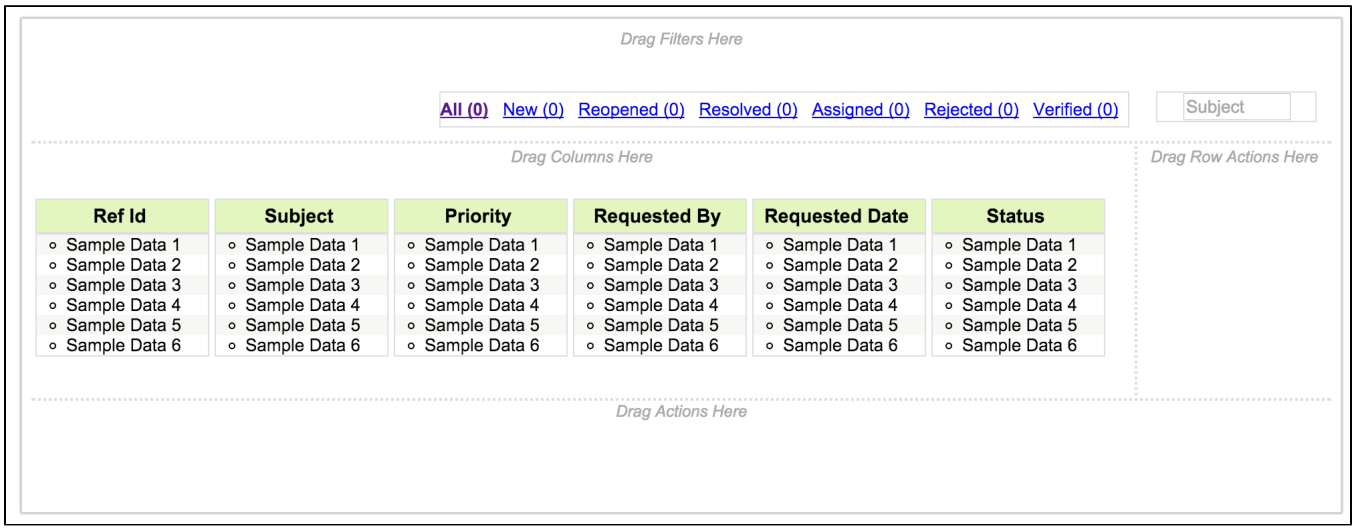
jar jar



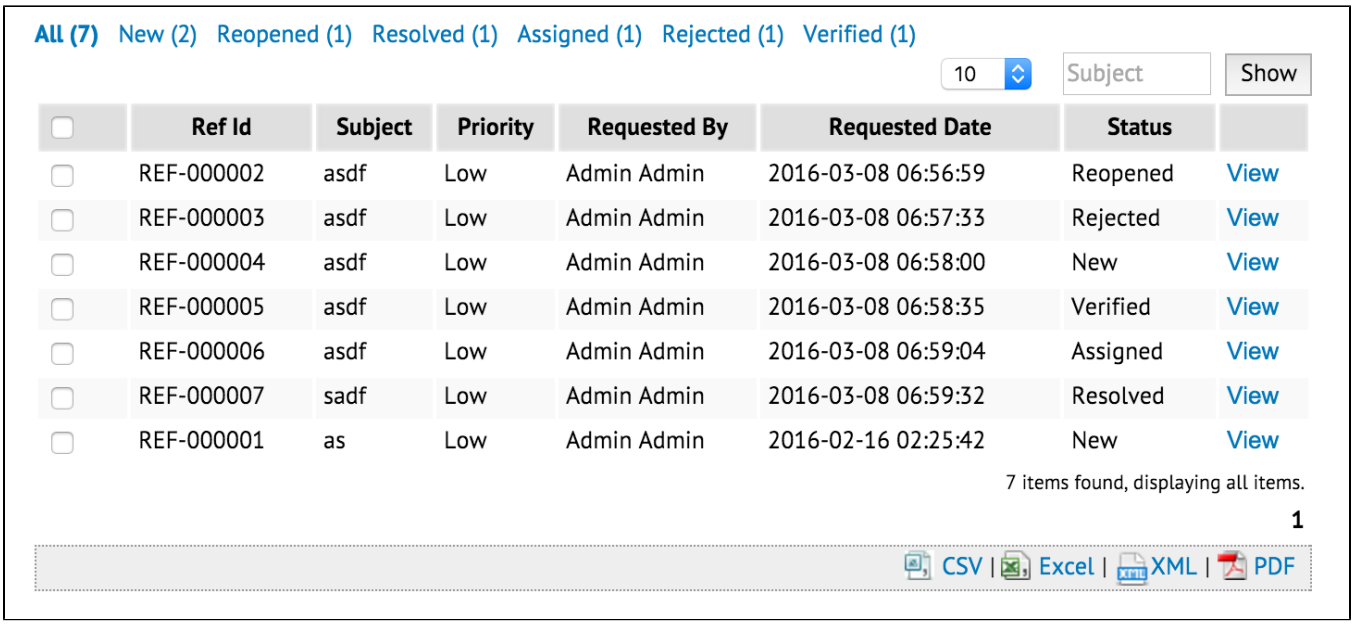
“”“.”




Configure Hyperlink Options Filter
General > Configure Hyperlink Options Filter > Choose Options Binder
Default Value
Show label?
Display in full width (Above other filters)
Show Data Count?
Options
Value
Label
New
New
Reopened
Reopened
Resolved
Resolved
Assigned
Assigned
Rejected
Rejected
Verified
Verified
< Prev
Next >
OK
Cancel


Drag Filters Here
All (0)
New (0)
Reopened (0)
Resolved (0)
Assigned (0)
Rejected (0)
Verified (0)
Subject
Drag Columns Here
Drag Row Actions Here
Ref Id
Subject
Priority
Requested By
Requested Date
Status
Sample Data 1
Sample Data 2
Sample Data 3
Sample Data 4
Sample Data 5
Sample Data 6
Drag Actions Here

datalist



8.

hyperlink_options_filter_src.zip

jarhttp://marketplace.joget.org/