How to develop a Form Submission Statistics Generator (MySQL)

- 1. What is the problem?
- 2. What is your idea to solve the problem?
- 3. What is the input needed for your plugin?
- 4. What is the output and expected outcome of your plugin?
- 5. Is there any resources/API that can be reuse?
- 6. Prepare your development environment
- 7. Just code it!
    - a. Extending the abstract class of a plugin type
    - b. Implement all the abstract methods
    - c. Manage the dependency libraries of your plugin
    - d. Make your plugin internationalization (i18n) ready
    - e. Register your plugin to Felix Framework
    - f. Build it and testing
- 8. Take a step further, share it or sell it

In this tutorial, we will following the guideline of developing a plugin to develop our Form Submission Statistics Generator. Please also refer to the very first tutorial How to develop a Bean Shell Hash Variable for more details steps.

1. What is the problem?

We have 2 SQL Chart menus (the queries used are dependent on MySQL) to show form data submission statistics and this 2 menus always need to replicate for different form.

2. What is your idea to solve the problem?

We can develop a Generator Plugin to ease the process to replicate the 2 SQL Chart menus for other forms.

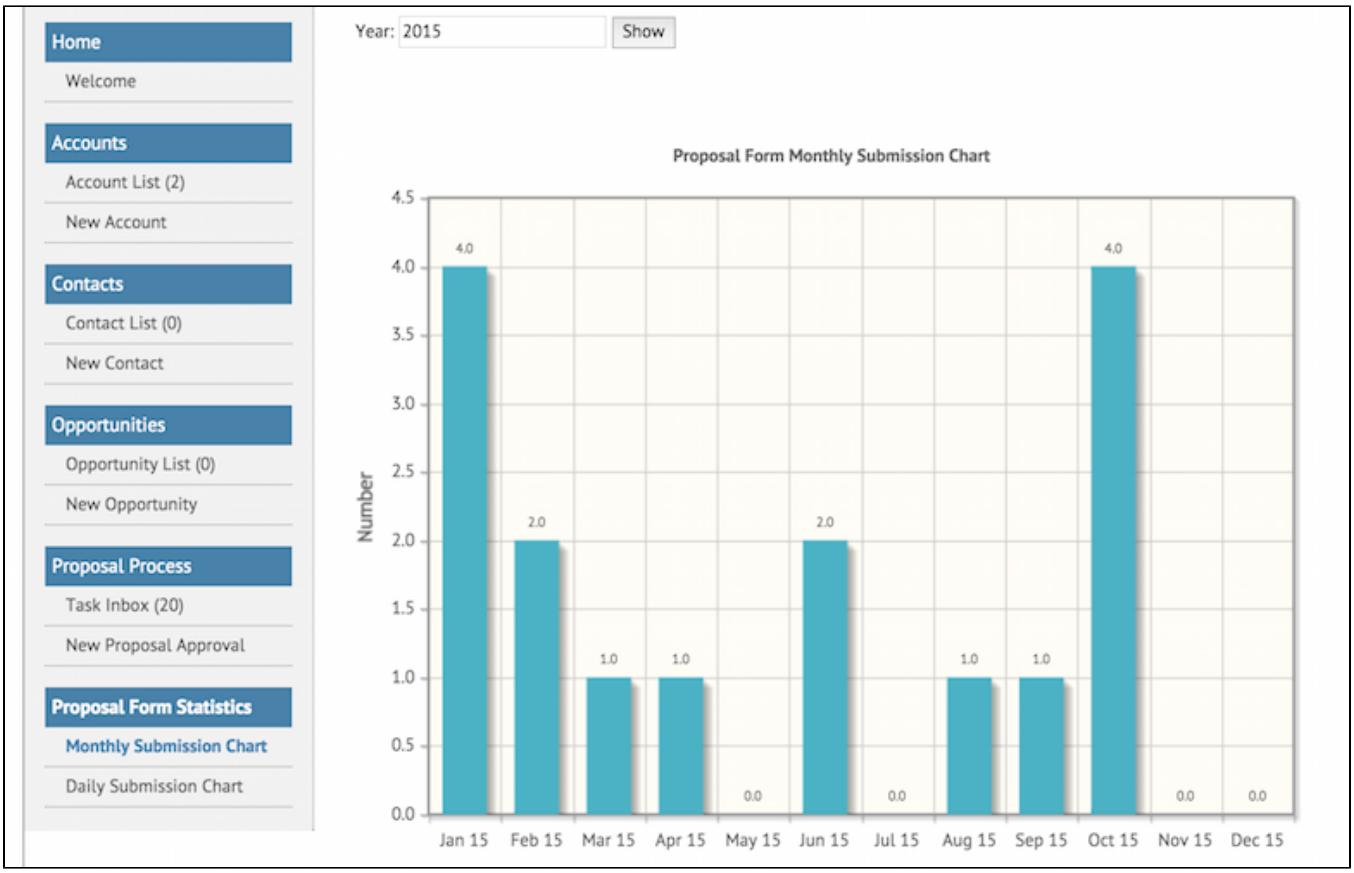
3. What is the input needed for your plugin?

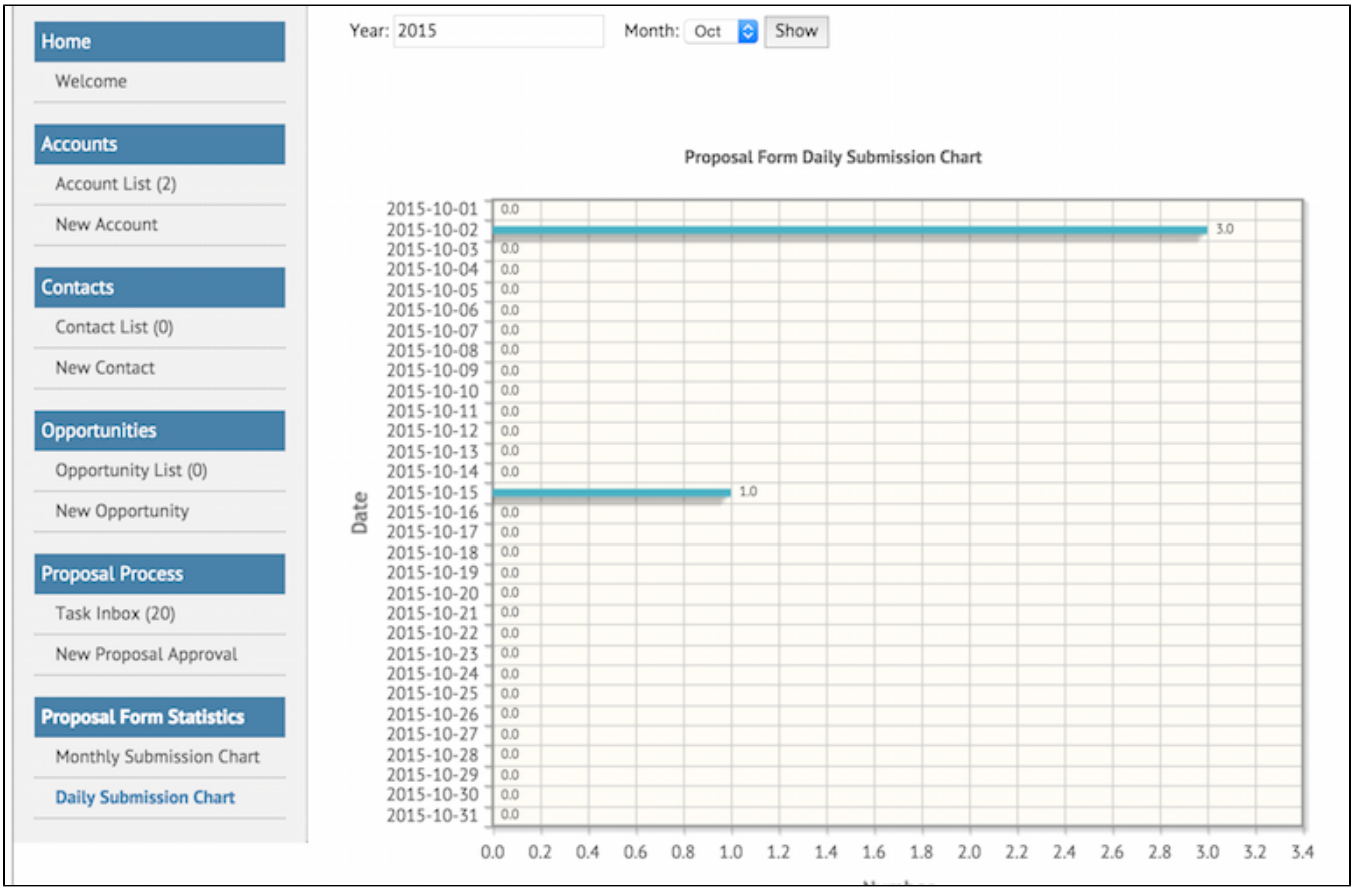
To develop a Generator Plugin for our 2 SQL Chart menus, we can consider to provide the following as input.

1. Userview ID : Which userview to add this 2 SQL menus
2. Some option for label change in Category label, Menu label, and Chart label

4. What is the output and expected outcome of your plugin?

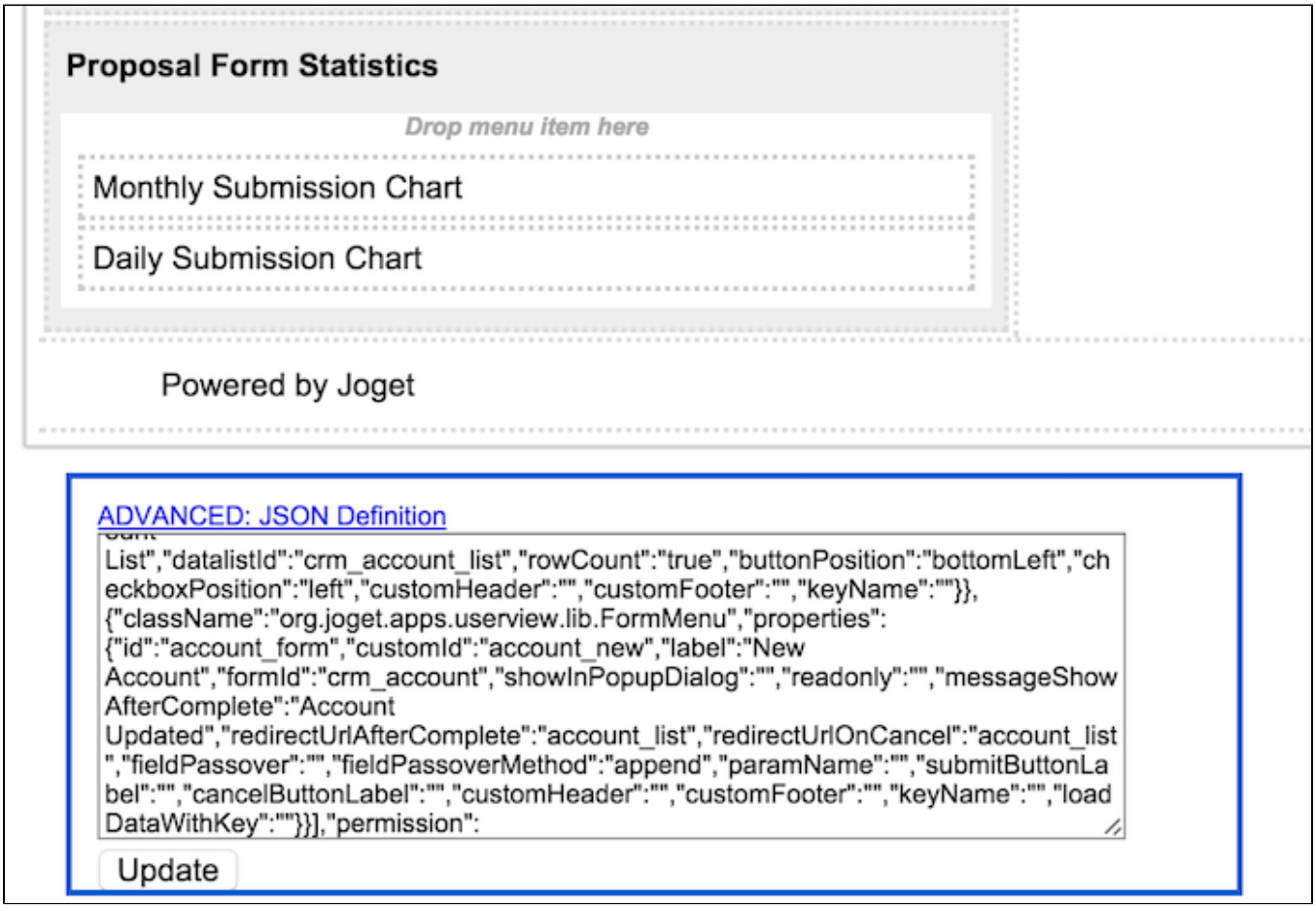
2 SQL Chart menus will be added to the selected userview under a new category. One of the menu will show the monthly submission chart and another one will show the daily submission chart based on the year and month filter.


Home
Welcome
Accounts
Account List (2)
New Account
Contacts
Contact List (0)
New Contact
Opportunities
Opportunity List (0)
New Opportunity
Proposal Process
Task Inbox (20)
New Proposal Approval
Proposal Form Statistics
Monthly Submission Chart
Daily Submission Chart
Year: 2015
Show
Proposal Form Monthly Submission Chart
Number
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
4.0
2.0
1.0
1.0
0.0
2.0
0.0
1.0
1.0
4.0
0.0
0.0
Jan 15
Feb 15
Mar 15
Apr 15
May 15
Jun 15
Jul 15
Aug 15
Sep 15
Oct 15
Nov 15
Dec 15


Home
Welcome
Accounts
Account List (2)
New Account
Contacts
Contact List (0)
New Contact
Opportunities
Opportunity List (0)
New Opportunity
Proposal Process
Task Inbox (20)
New Proposal Approval
Proposal Form Statistics
Monthly Submission Chart
Daily Submission Chart
Year: 2015
Month: Oct
Show
Proposal Form Daily Submission Chart
Date
2015-10-01
2015-10-02
2015-10-03
2015-10-04
2015-10-05
2015-10-06
2015-10-07
2015-10-08
2015-10-09
2015-10-10
2015-10-11
2015-10-12
2015-10-13
2015-10-14
2015-10-15
2015-10-16
2015-10-17
2015-10-18
2015-10-19
2015-10-20
2015-10-21
2015-10-22
2015-10-23
2015-10-24
2015-10-25
2015-10-26
2015-10-27
2015-10-28
2015-10-29
2015-10-30
2015-10-31
3.0
1.0
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0 3.2 3.4

5. Is there any resources/API that can be reuse?

First, we can do build our 2 SQL Chart menus in one of the existing userview. Then, copy the JSON definition of the category which contains our 2 SQL Chart menus form the "ADVANCED: JSON Definition" on the bottom of the UI Builder.



We will get our JSON definition of the category as following. Please note that the queries used in the 2 SQL Chart menus are dependent on MySQL database.

```
{
    "className": "org.joget.apps.userview.model.UserviewCategory",
    "properties": {
        "id": "category-8722A52FFBB64D058E2CD41174922807",
        "label": "Proposal Form Statistics"
    },
    "menus": [{
        "className": "org.joget.plugin.enterprise.SqlChartMenu",
        "properties": {
            "id": "807F165BFA9C4BB589E5B52E4C071250",
            "customId": "crm_proposal_monthly",
            "label": "Monthly Submission Chart",
            "chartType": "bar",
            "title": "Proposal Form Monthly Submission Chart",
            "categoryAxisLabel": "Month",
            "xAxisDisplayAS": "",
            "valueAxisLabel": "Number",
            "yaxisPrefix": "",
            "showLegend": "",
            "showValueLabel": "true",
            "stack": "",
            "horizontal": "",
            "chartWidth": "100%",
            "chartHeight": "80%",
            "colors": "",
            "query": "SELECT DATE_FORMAT(STR_TO_DATE(m.monthYear, '%c-%Y'),'%b %y') as monthYear, COUNT(fd.
dateCreated) AS 'Number'\nFROM \n(\n    SELECT my.month, CONCAT(my.month, '-', IF(('#requestParam.year?sql#'
```

```
REGEXP '^[0-9]{4}$'), '#requestParam.year?sql#' , '#date.yyyy?sql#')) AS monthYear FROM (\n        SELECT 1 AS
month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8
UNION SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12\n    ) my\n) m\nLEFT JOIN \n(\n\t  select
dateCreated FROM app_fd_crm_proposal\n) fd\nON m.monthYear=DATE_FORMAT(fd.dateCreated,'%c-%Y')\nGROUP BY
monthYear\nORDER BY m.month",
            "customHeader": "<div class=\"filter\">\n    <form action=\"?\" method=\"GET\">\n
<label>Year: <\/label><input name=\"year\" value=\"#requestParam.year?html#\"\/>\n        <input type=\"
submit\" value=\"Show\"\/>\n    <\/form>\n<\/div>\n<script>\n    $(function(){\n        if ($(\"[name='year']
\").val() === \"\") {\n            $(\"[name='year']\").val(\"#date.yyyy#\");\n        }\n    });\n<\
/script>\n<br\/>\n<br\/>",
            "customFooter": "",
            "datasource": "default",
            "keyName": ""
        }
    }, {
        "className": "org.joget.plugin.enterprise.SqlChartMenu",
        "properties": {
            "id": "39E2163319D84FD693D164D92FA93C06",
            "customId": "crm_proposal_daily",
            "label": "Daily Submission Chart",
            "chartType": "bar",
            "title": "Proposal Form Daily Submission Chart",
            "categoryAxisLabel": "Date",
            "xAxisDisplayAS": "",
            "valueAxisLabel": "Number",
            "yaxisPrefix": "",
            "showLegend": "",
            "showValueLabel": "true",
            "stack": "",
            "horizontal": "true",
            "chartWidth": "100%",
            "chartHeight": "80%",
            "colors": "",
            "query": "SELECT d.date_field, COUNT(fd.dateCreated) AS 'Number'\nFROM\n(\n    SELECT\n
MAKEDATE(IF(('#requestParam.year?sql#' REGEXP '^[0-9]{4}$'), '#requestParam.year?sql#' , '#date.yyyy?sql#'),1)
+\n        INTERVAL (IF(('#requestParam.month?sql#' REGEXP '^[0-9]{2}$'), '#requestParam.month?sql#' , '#date.
MM?sql#') -1) MONTH +\n        INTERVAL daynum DAY date_field\n    FROM\n    (\n        SELECT t*10+u
daynum\n        FROM\n            (SELECT 0 t UNION SELECT 1 UNION SELECT 2 UNION SELECT 3) A,\n
(SELECT 0 u UNION SELECT 1 UNION SELECT 2 UNION SELECT 3\n            UNION SELECT 4 UNION SELECT 5 UNION
SELECT 6 UNION SELECT 7\n            UNION SELECT 8 UNION SELECT 9) B\n        ORDER BY daynum\n    ) AA\n)
d\nLEFT JOIN \n(\n\t  select dateCreated FROM app_fd_crm_proposal\n) fd\nON d.date_field=DATE_FORMAT(fd.
dateCreated,'%Y-%m-%d')\nWHERE DATE_FORMAT(d.date_field,'%m') = IF(('#requestParam.month?sql#' REGEXP '^[0-9]
{2}$'), '#requestParam.month?sql#' , '#date.MM?sql#')\nGROUP BY d.date_field\nORDER BY d.date_field desc",
            "customHeader": "<div class=\"filter\">\n    <form action=\"?\" method=\"GET\">\n
<label>Year: <\/label><input name=\"year\" value=\"#requestParam.year?html#\"\/>   \n
<label>Month: <\/label><select name=\"month\"\/>\n            <option value=\"01\">Jan<\/option>\n
<option value=\"02\">Feb<\/option>\n            <option value=\"03\">Mar<\/option>\n            <option value=\"
04\">Apr<\/option>\n            <option value=\"05\">May<\/option>\n            <option value=\"06\">Jun<\
/option>\n            <option value=\"07\">Jul<\/option>\n            <option value=\"08\">Aug<\
/option>\n            <option value=\"09\">Sep<\/option>\n            <option value=\"10\">Oct<\
/option>\n            <option value=\"11\">Nov<\/option>\n            <option value=\"12\">Dec<\
/option>\n        <\/select>\n        <input type=\"submit\" value=\"Show\"\/>\n    <\/form>\n<\
/div>\n<script>\n    $(function(){\n        if ($(\"[name='year']\").val() === \"\") {\n            $(\"
[name='year']\").val(\"#date.yyyy#\");\n        }\n        if ($(\"[name='month']\").val() !== \"#requestParam.
month?javascript#\" \n            && '#requestParam.month?javascript#' !== \"\"\n            && $(\"
[name='month'] option[value='#requestParam.month?javascript#']\").length > 0 ) {\n            $(\"[name='month']
\").val('#requestParam.month?javascript#');\n        } else {\n            $(\"[name='month']\").val(\"#date.
MM#\");\n        }\n    });\n<\/script>\n<br\/>\n<br\/>",
            "customFooter": "",
            "datasource": "default",
            "keyName": ""
        }
    }]
}
```

After that, we can utilize the GeneratorUtil to add the category JSON definition to our selected userview JSON definition.

6. Prepare your development environment

We need to always have our Joget Workflow Source Code ready and builded by following this guideline.

The following of this tutorial is prepared with a Macbook Pro and Joget Source Code version 5.0.0. Please refer to Guideline for Developing a Plugin for other platform command.

Let said our folder directory as following.

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

The "plugins" directory is the folder we will create and store all our plugins and the "jw-community" directory is where the Joget Workflow Source code stored.

Run the following command to create a maven project in "plugins" directory.

```
cd joget/plugins/
~/joget/jw-community/wflow-plugin-archetype/create-plugin.sh org.joget.tutorial
form_submission_statistics_generator 8.0-Snapshot
```

Then, the shell script will ask us to key in a version for your plugin and ask us for confirmation before generate the maven project.

```
Define value for property 'version':  1.0-SNAPSHOT: : 8.0-Snapshot
[INFO] Using property: package = org.joget.tutorial
Confirm properties configuration:
groupId: org.joget.tutorial
artifactId: form_submission_statistics_generator
version: 5.0.0
package: org.joget.tutorial
Y: : y
```

We should get "BUILD SUCCESS" message shown in our terminal and a "form_submission_statistics_generator" folder created in "plugins" folder.

Open the maven project with your favour IDE. I will be using NetBeans.

7. Just code it!

a. Extending the abstract class of a plugin type

Create a "FormSubmissionStatisticsGenerator" class under "org.joget.tutorial" package. Then, extend the class with **org.joget.apps.generator.model.GeneratorPlugin** abstract class. Please refer to Generator Plugin.

b. Implement all the abstract methods

As usual, we have to implement all the abstract methods. We will using AppPluginUtil.getMessage method to support i18n and using constant variable MESSAGE_PATH for message resource bundle directory.

Implementation of all basic abstract methods

```
package org.joget.tutorial;
import org.joget.apps.app.service.AppPluginUtil;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.generator.model.GeneratorPlugin;
import org.joget.apps.generator.model.GeneratorResult;
public class FormSubmissionStatisticsGenerator extends GeneratorPlugin {

    private final static String MESSAGE_PATH = "messages/FormSubmissionStatisticsGenerator";

    public String getName() {
        return "File Link Datalist Formatter";
    }

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

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

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

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

    public String getPropertyOptions() {
        return AppUtil.readPluginResource(getClassName(), "/properties/formSubmissionStatisticsGenerator.json",
null, true, MESSAGE_PATH);
    }
    @Override
    public String getExplanation() {
        //support i18n
        return AppPluginUtil.getMessage("generator.formSubmissionStatistics.explanation", getClassName(),
MESSAGE_PATH);
    }
    @Override
    public GeneratorResult generate() {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

Then, we have to do a UI for admin user to provide inputs for our plugin. In getPropertyOptions method, we already specify our Plugin Properties Options definition file is locate at "/properties /formSubmissionStatisticsGenerator.json". Let us create a directory "resources/properties" under "form_submission_statistics_generator/src/main" directory. After create the directory, create a file named "formSubmissionStatisticsGenerator.json" in the "properties" folder.

In the properties definition options file, we will need to provide options as below. Please note that we can use "@@message.key@@" syntax to support i18n in our properties options.

```
[{
    title : '@@generator.formSubmissionStatistics.config@@',
    properties : [
    {
        name : 'userviewId',
        label : '@@generator.formSubmissionStatistics.userview@@',
        type : 'selectbox',
        value: '[default_userviewId]',
        options_ajax : '[CONTEXT_PATH]/web/json/console/app[APP_PATH]/userview/options'
    }]
},
```

```
{
    title : '@@generator.formSubmissionStatistics.advanced@@',
    properties : [
    {
        label : '@@generator.formSubmissionStatistics.label.options@@',
        type : 'header'
    },
    {
        name : 'monthlyChartTitle',
        label : '@@generator.formSubmissionStatistics.monthlyChartTitle@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.monthlyChartTitle.value@@'
    },
    {
        name : 'monthlyXAxisLabel',
        label : '@@generator.formSubmissionStatistics.monthlyXAxisLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.monthlyXAxisLabel.value@@'
    },
    {
        name : 'dailyChartTitle',
        label : '@@generator.formSubmissionStatistics.dailyChartTitle@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.dailyChartTitle.value@@'
    },
    {
        name : 'dailyXAxisLabel',
        label : '@@generator.formSubmissionStatistics.dailyXAxisLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.dailyXAxisLabel.value@@'
    },
    {
        name : 'yAxisLabel',
        label : '@@generator.formSubmissionStatistics.yAxisLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.yAxisLabel.value@@'
    },
    {
        name : 'yearLabel',
        label : '@@generator.formSubmissionStatistics.yearLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.yearLabel.value@@'
    },
    {
        name : 'monthLabel',
        label : '@@generator.formSubmissionStatistics.monthLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.monthLabel.value@@'
    },
    {
        name : 'showLabel',
        label : '@@generator.formSubmissionStatistics.showLabel@@',
        type : 'textfield',
        required : 'true',
        value : '@@generator.formSubmissionStatistics.showLabel.value@@'
    },
    {
        label : '@@generator.formSubmissionStatistics.useriewMenu.options@@',
        type : 'header'
    },
    {
        name : 'categoryLabel',
        label : '@@generator.formSubmissionStatistics.categoryLabel@@',
        type : 'textfield',
```

```
    required : 'true',
    value : '@@generator.formSubmissionStatistics.categoryLabel.value@@'
},
{
    name : 'monthlyMenuId',
    label : '@@generator.formSubmissionStatistics.monthlyMenuId@@',
    type : 'textfield',
    required : 'true',
    regex_validation : '^[a-zA-Z0-9_]+$',
    validation_message : '@@generator.formSubmissionStatistics.menuId.invalidId@@',
    value : '[formId]_monthly'
},
{
    name : 'monthlyMenuLabel',
    label : '@@generator.formSubmissionStatistics.monthlyMenuLabel@@',
    type : 'textfield',
    required : 'true',
    value : '@@generator.formSubmissionStatistics.monthlyMenuLabel.value@@'
},
{
    name : 'dailyMenuId',
    label : '@@generator.formSubmissionStatistics.dailyMenuId@@',
    type : 'textfield',
    required : 'true',
    regex_validation : '^[a-zA-Z0-9_]+$',
    validation_message : '@@generator.formSubmissionStatistics.menuId.invalidId@@',
    value : '[formId]_daily'
},
{
    name : 'dailyMenuLabel',
    label : '@@generator.formSubmissionStatistics.dailyMenuLabel@@',
    type : 'textfield',
    required : 'true',
    value : '@@generator.formSubmissionStatistics.dailyMenuLabel.value@@'
},
{
    label : '@@generator.formSubmissionStatistics.createUserviewOptions@@',
    type : 'header',
    control_field: 'userviewId',
    control_value: '',
    control_use_regex: 'false',
},
{
    name : 'userviewNewId',
    label : '@@generator.formSubmissionStatistics.userview.id@@',
    type : 'textfield',
    required : 'true',
    value : '@@generator.formSubmissionStatistics.userview.id.value@@',
    regex_validation : '^[a-zA-Z0-9_]+$',
    validation_message : '@@generator.formSubmissionStatistics.userview.id.invalidId@@',
    control_field: 'userviewId',
    control_value: '',
    control_use_regex: 'false'
},
{
    name : 'userviewName',
    label : '@@generator.formSubmissionStatistics.userview.name@@',
    type : 'textfield',
    required : 'true',
    value : '@@generator.formSubmissionStatistics.userview.name.value@@',
    control_field: 'userviewId',
    control_value: '',
    control_use_regex: 'false'
},
{
    name : 'userviewDesc',
    label : '@@generator.formSubmissionStatistics.userview.description@@',
    type : 'textarea',
    rows : "3",
    control_field: 'userviewId',
    control_value: '',
```

```
        control_use_regex: 'false'
    }]
}]
```

In the plugin properties option, you may notice we are using "[default_userviewId]" in "userviewId" property. We will use the "[formName]" and "[formId]" in some properties as well. So, we will need to modify the getPropertyOptions method to cater for this values.

```
    public String getPropertyOptions() {
        String options = AppUtil.readPluginResource(getClassName(), "/properties
/formSubmissionStatisticsGenerator.json", null, true, MESSAGE_PATH);

        //populate value like [formName] and [formId]
        options = GeneratorUtil.populateFormMeta(options, getFormId(), getAppDefinition());

        //populate value of [default_userviewId]
        options = options.replace("[default_userviewId]", GeneratorUtil.getFirstAvailableUserviewId
(getAppDefinition()));

        return options;
    }
```

Once we done the properties option to collect input, we can work on the main method of the plugin which is format method.

```
    @Override
    public GeneratorResult generate() {
        GeneratorResult result = new GeneratorResult();
        AppDefinition appDef = getAppDefinition();

        try {
            //get userview
            UserviewDefinitionDao userviewDefinitionDao = (UserviewDefinitionDao) AppUtil.
getApplicationContext().getBean("userviewDefinitionDao");
            UserviewDefinition userviewDef = null;
            String json = null;
            String userviewId = getPropertyString("userviewId");
            String userviewName;
            String userviewDesc;
            if (userviewId != null && !userviewId.isEmpty()) {
                userviewDef = userviewDefinitionDao.loadById(userviewId, appDef);
            }
            if (userviewDef != null) {
                userviewName = userviewDef.getName();
                userviewDesc = userviewDef.getDescription();
                json = userviewDef.getJson();
            } else {
                userviewId = getPropertyString("userviewNewId");
                int count = 0;
                while (isExist(userviewId, userviewDefinitionDao)) {
                    count++;
                    userviewId = userviewId + count;
                }
                userviewName = getPropertyString("userviewName");
                userviewDesc = getPropertyString("userviewDesc");
            }
            if (json == null || json.isEmpty()) {
                //create a new userview json
                json = GeneratorUtil.createNewUserviewJson(userviewId, userviewName, userviewDesc);
            }

            //add the category json to userview json
```

```
            json = GeneratorUtil.addCategoryJsonToUserviewJson(getCategoryJson(appDef), json);

            //Store the userview json
            if (userviewDef != null) {
                userviewDef.setJson(json);
                userviewDefinitionDao.update(userviewDef);
            }else {
                userviewDef = new UserviewDefinition();
                userviewDef.setJson(json);
                userviewDef.setId(userviewId);
                userviewDef.setName(userviewName);
                userviewDef.setAppDefinition(appDef);
                userviewDefinitionDao.add(userviewDef);

                //Set current published version
                final AppDefinition currentAppDef = appDef;
                TransactionTemplate transactionTemplate = (TransactionTemplate)AppUtil.getApplicationContext().
getBean("transactionTemplate");
                transactionTemplate.execute(new TransactionCallback<Object>() {
                    public Object doInTransaction(TransactionStatus ts) {
                        AppService appService = (AppService)AppUtil.getApplicationContext().getBean
("appService");
                        appService.publishApp(currentAppDef.getId(), currentAppDef.getVersion().toString());
                        return null;
                    }
                });
            }

            //show message to continue edit
            String editLink = WorkflowUtil.getHttpServletRequest().getContextPath() + "/web/console/app/"
+appDef.getAppId()+"/"+appDef.getVersion()+"/userview/builder/"+userviewId;
            String msg = AppPluginUtil.getMessage("generator.formSubmissionStatistics.msg.success",
getClassName(), MESSAGE_PATH);
            msg = msg.replace("[url]", editLink);

            result.setMessage(msg);
        } catch (Exception e) {
            result.setError(true);
            result.setMessage(AppPluginUtil.getMessage("generator.formSubmissionStatistics.msg.error",
getClassName(), MESSAGE_PATH));
            LogUtil.error(getClassName(), e, "Not able to generate the menus");
        }

        return result;
    }

    /**
     * Retrieves the category JSON definition
     * @param appDef
     * @return
     */
    protected String getCategoryJson(AppDefinition appDef) {
        Collection<String> args = new ArrayList<String>();

        AppService appService = (AppService)AppUtil.getApplicationContext().getBean("appService");
        String tabelName = appService.getFormTableName(appDef, getFormId());

        args.add(UuidGenerator.getInstance().getUuid()); //category id
        args.add(StringUtil.escapeString(getPropertyString("categoryLabel"), StringUtil.TYPE_JSON, null));
//category label
        args.add(UuidGenerator.getInstance().getUuid()); //monthly menu uuid
        args.add(StringUtil.escapeString(getPropertyString("monthlyMenuId"), StringUtil.TYPE_JSON, null));
//monthly menu custom id
        args.add(StringUtil.escapeString(getPropertyString("monthlyMenuLabel"), StringUtil.TYPE_JSON, null));
//monthly menu label
        args.add(StringUtil.escapeString(getPropertyString("monthlyChartTitle"), StringUtil.TYPE_JSON, null));
//monthly chart title
        args.add(StringUtil.escapeString(getPropertyString("monthlyXAxisLabel"), StringUtil.TYPE_JSON, null));
//monthly x-axis label
        args.add(StringUtil.escapeString(getPropertyString("yAxisLabel"), StringUtil.TYPE_JSON, null));
//monthly y-axis label
```

```
        args.add(StringUtil.escapeString(tabelName, StringUtil.TYPE_JSON, null)); //monthly form table name
        args.add(StringUtil.escapeString(getPropertyString("yearLabel"), StringUtil.TYPE_JSON, null));
//monthly year label
        args.add(StringUtil.escapeString(getPropertyString("showLabel"), StringUtil.TYPE_JSON, null));
//monthly show button label
        args.add(UuidGenerator.getInstance().getUuid()); //daily menu uuid
        args.add(StringUtil.escapeString(getPropertyString("dailyMenuId"), StringUtil.TYPE_JSON, null));
//daily menu custom id
        args.add(StringUtil.escapeString(getPropertyString("dailyMenuLabel"), StringUtil.TYPE_JSON, null));
//daily menu label
        args.add(StringUtil.escapeString(getPropertyString("dailyChartTitle"), StringUtil.TYPE_JSON, null));
//daily chart title
        args.add(StringUtil.escapeString(getPropertyString("dailyXAxisLabel"), StringUtil.TYPE_JSON, null));
//daily x-axis label
        args.add(StringUtil.escapeString(getPropertyString("yAxisLabel"), StringUtil.TYPE_JSON, null)); //daily
y-axis label
        args.add(StringUtil.escapeString(tabelName, StringUtil.TYPE_JSON, null)); //daily form table name
        args.add(StringUtil.escapeString(getPropertyString("yearLabel"), StringUtil.TYPE_JSON, null)); //daily
year label
        args.add(StringUtil.escapeString(getPropertyString("monthLabel"), StringUtil.TYPE_JSON, null)); //daily
month label
        args.add(StringUtil.escapeString(getPropertyString("showLabel"), StringUtil.TYPE_JSON, null)); //daily
show button label

        String json = AppUtil.readPluginResource(getClass().getName(), "/resources/category.json", args.
toArray(), true, null);

        return json;
    }

    /**
     * Checks for a userview is already exist
     * @param id
     * @param userviewDefinitionDao
     * @return
     */
    protected boolean isExist(String id, UserviewDefinitionDao userviewDefinitionDao) {
        Long count = userviewDefinitionDao.count("AND id = ?", new String[]{id}, getAppDefinition());
        return count > 0;
    }
```

In the getCategoryJson method, we will retrieve the category JSON definition form "/resources/category.json" file. Let us create a directory "resources/resources" under "form_submission_statistics_generator/src/main" directory. After create the directory, create a file named "category.json" in the "resources" folder. Then, copy the category JSON definition we created previously and paste it inside this file. We will need to replace some of the hardcoded value as variable and remember to escape those existing "%" to "%%" as the AppUtil.readPluginResource using String.format for injects value into the file.

```
{
    "className": "org.joget.apps.userview.model.UserviewCategory",
    "properties": {
        "id": "category-%s",
        "label": "%s"
    },
    "menus": [{
        "className": "org.joget.plugin.enterprise.SqlChartMenu",
        "properties": {
            "id": "%s",
            "customId": "%s",
            "label": "%s",
            "chartType": "bar",
            "title": "%s",
            "categoryAxisLabel": "%s",
            "xAxisDisplayAS": "",
            "valueAxisLabel": "%s",
            "yaxisPrefix": "",
            "showLegend": "",
            "showValueLabel": "true",
            "stack": "",
            "horizontal": "",
            "chartWidth": "100%%",
            "chartHeight": "80%%",
```

```
            "colors": "",
            "query": "SELECT DATE_FORMAT(STR_TO_DATE(m.monthYear, '%%c-%%Y'),'%%b %%y') as monthYear, COUNT(fd.
dateCreated) AS 'Number'\nFROM \n(\n    SELECT my.month, CONCAT(my.month, '-', IF(('#requestParam.year?sql#'
REGEXP '^[0-9]{4}$'), '#requestParam.year?sql#' , '#date.yyyy?sql#')) AS monthYear FROM (\n        SELECT 1 AS
month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8
UNION SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12\n    ) my\n) m\nLEFT JOIN \n(\n\t  select
dateCreated FROM app_fd_%s\n) fd\nON m.monthYear=DATE_FORMAT(fd.dateCreated,'%%c-%%Y')\nGROUP BY
monthYear\nORDER BY m.month",
            "customHeader": "<div class=\"filter\">\n    <form action=\"?\" method=\"GET\">\n        <label>%s:
<\/label><input name=\"year\" value=\"#requestParam.year?html#\"\/>\n        <input type=\"submit\" value=\"%s\"
\/>\n    <\/form>\n<\/div>\n<script>\n    $(function(){\n        if ($(\"[name='year']\").val() === \"\")
{\n            $(\"[name='year']\").val(\"#date.yyyy#\");\n        }\n    });\n<\/script>\n<br\/>\n<br\/>",
            "customFooter": "",
            "datasource": "default",
            "keyName": ""
        }
    }, {
        "className": "org.joget.plugin.enterprise.SqlChartMenu",
        "properties": {
            "id": "%s",
            "customId": "%s",
            "label": "%s",
            "chartType": "bar",
            "title": "%s",
            "categoryAxisLabel": "%s",
            "xAxisDisplayAS": "",
            "valueAxisLabel": "%s",
            "yaxisPrefix": "",
            "showLegend": "",
            "showValueLabel": "true",
            "stack": "",
            "horizontal": "true",
            "chartWidth": "100%%",
            "chartHeight": "80%%",
            "colors": "",
            "query": "SELECT d.date_field, COUNT(fd.dateCreated) AS 'Number'\nFROM\n(\n    SELECT\n
MAKEDATE(IF(('#requestParam.year?sql#' REGEXP '^[0-9]{4}$'), '#requestParam.year?sql#' , '#date.yyyy?sql#'),1)
+\n        INTERVAL (IF(('#requestParam.month?sql#' REGEXP '^[0-9]{2}$'), '#requestParam.month?sql#' , '#date.
MM?sql#') -1) MONTH +\n        INTERVAL daynum DAY date_field\n    FROM\n    (\n        SELECT t*10+u
daynum\n        FROM\n            (SELECT 0 t UNION SELECT 1 UNION SELECT 2 UNION SELECT 3) A,\n
(SELECT 0 u UNION SELECT 1 UNION SELECT 2 UNION SELECT 3\n            UNION SELECT 4 UNION SELECT 5 UNION
SELECT 6 UNION SELECT 7\n            UNION SELECT 8 UNION SELECT 9) B\n        ORDER BY daynum\n    ) AA\n)
d\nLEFT JOIN \n(\n\t  select dateCreated FROM app_fd_%s\n) fd\nON d.date_field=DATE_FORMAT(fd.dateCreated,'%%Y-
%%m-%%d')\nWHERE DATE_FORMAT(d.date_field,'%%m') = IF(('#requestParam.month?sql#' REGEXP '^[0-9]{2}$'),
'#requestParam.month?sql#' , '#date.MM?sql#')\nGROUP BY d.date_field\nORDER BY d.date_field desc",
            "customHeader": "<div class=\"filter\">\n    <form action=\"?\" method=\"GET\">\n        <label>%s:
<\/label><input name=\"year\" value=\"#requestParam.year?html#\"\/>   \n        <label>%s: <\
/label><select name=\"month\"\/>\n            <option value=\"01\">Jan<\/option>\n            <option value=\"
02\">Feb<\/option>\n            <option value=\"03\">Mar<\/option>\n            <option value=\"04\">Apr<\
/option>\n            <option value=\"05\">May<\/option>\n            <option value=\"06\">Jun<\
/option>\n            <option value=\"07\">Jul<\/option>\n            <option value=\"08\">Aug<\
/option>\n            <option value=\"09\">Sep<\/option>\n            <option value=\"10\">Oct<\
/option>\n            <option value=\"11\">Nov<\/option>\n            <option value=\"12\">Dec<\
/option>\n        <\/select>\n        <input type=\"submit\" value=\"%s\"\/>\n    <\/form>\n<\
/div>\n<script>\n    $(function(){\n        if ($(\"[name='year']\").val() === \"\") {\n            $(\"
[name='year']\").val(\"#date.yyyy#\");\n        }\n        if ($(\"[name='month']\").val() !== \"#requestParam.
month?javascript#\" \n            && '#requestParam.month?javascript#' !== \"\"\n            && $(\"
[name='month'] option[value='#requestParam.month?javascript#']\").length > 0 ) {\n            $(\"[name='month']
\").val('#requestParam.month?javascript#');\n        } else {\n            $(\"[name='month']\").val(\"#date.
MM#\");\n        }\n    });\n<\/script>\n<br\/>\n<br\/>",
            "customFooter": "",
            "datasource": "default",
            "keyName": ""
        }
    }]
}
```

c. Manage the dependency libraries of your plugin

Our plugin is using javax.servlet.http.HttpServletRequest and javax.servlet.http.HttpServletResponse class, so we will need to add jsp-api library to our POM file.

```
<!-- Change plugin specific dependencies here -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.0</version>
</dependency>
<!-- End change plugin specific dependencies here -->
```

d. Make your plugin internationalization (i18n) ready

We are using i18n message key in getLabel and getDescription method. We also used i18n message key in our properties options definition as well. So, we will need to create a message resource bundle properties file for our plugin.Create directory "resources/messages" under "form_submission_statistics_generator/src/main" directory. Then, create a "FormSubmissionStatisticsGenerator.properties" file in the folder. In the properties file, let add all the message keys and its label as below.

```
org.joget.tutorial.FormSubmissionStatisticsGenerator.pluginLabel=Generate Form Submission Statistics
org.joget.tutorial.FormSubmissionStatisticsGenerator.pluginDesc=Generate 2 SQL Chart Menus for form submission
statistics
generator.formSubmissionStatistics.explanation=Create 2 SQL Chart Menus based on the current form for
submission statistics. Note: This is dependent for MySQL database.
generator.formSubmissionStatistics.config=Options
generator.formSubmissionStatistics.userview=Userview
generator.formSubmissionStatistics.advanced=Advanced
generator.formSubmissionStatistics.label.options=Label Options
generator.formSubmissionStatistics.categoryLabel=Category Label
generator.formSubmissionStatistics.categoryLabel.value=[formName] Statistics
generator.formSubmissionStatistics.menuId.invalidId=Only alpha-numeric and underscore characters allowed
generator.formSubmissionStatistics.monthlyMenuId=Monthly Menu id
generator.formSubmissionStatistics.monthlyMenuLabel=Montly Menu Label
generator.formSubmissionStatistics.monthlyMenuLabel.value=Monthly Submission Chart
generator.formSubmissionStatistics.monthlyChartTitle=Montly Chart Title
generator.formSubmissionStatistics.monthlyChartTitle.value=[formName] Monthly Submission Chart
generator.formSubmissionStatistics.monthlyXAxisLabel=Montly X-axis Label
generator.formSubmissionStatistics.monthlyXAxisLabel.value=Month
generator.formSubmissionStatistics.dailyMenuId=Daily Menu Id
generator.formSubmissionStatistics.dailyMenuLabel=Daily Menu Label
generator.formSubmissionStatistics.dailyMenuLabel.value=Daily Submission Chart
generator.formSubmissionStatistics.dailyChartTitle=Daily Chart Title
generator.formSubmissionStatistics.dailyChartTitle.value=[formName] Daily Submission Chart
generator.formSubmissionStatistics.dailyXAxisLabel=Daily X-axis Label
generator.formSubmissionStatistics.dailyXAxisLabel.value=Date
generator.formSubmissionStatistics.yAxisLabel=Y-axis Label
generator.formSubmissionStatistics.yAxisLabel.value=Number
generator.formSubmissionStatistics.yearLabel=Year Label
generator.formSubmissionStatistics.yearLabel.value=Year
generator.formSubmissionStatistics.monthLabel=Month Label
generator.formSubmissionStatistics.monthLabel.value=Month
generator.formSubmissionStatistics.showLabel=Show Label
generator.formSubmissionStatistics.showLabel.value=Show
generator.formSubmissionStatistics.useriewMenu.options=Userview Label Options
generator.formSubmissionStatistics.createUserviewOptions=Create New Userview Options
generator.formSubmissionStatistics.userview.id=Userview ID
generator.formSubmissionStatistics.userview.id.value=v
generator.formSubmissionStatistics.userview.id.invalidId=Only alpha-numeric and underscore characters allowed
generator.formSubmissionStatistics.userview.name=Userview Name
generator.formSubmissionStatistics.userview.name.value=[appName]
generator.formSubmissionStatistics.userview.description=Userview Description
generator.formSubmissionStatistics.msg.success=Menus generated. Click <a href="[url]" target="_blank">here</a>
to continue edit in Userview Builder.
generator.formSubmissionStatistics.msg.error=Error during generating Form Submission Statistics Menus!
```

e. Register your plugin to Felix Framework

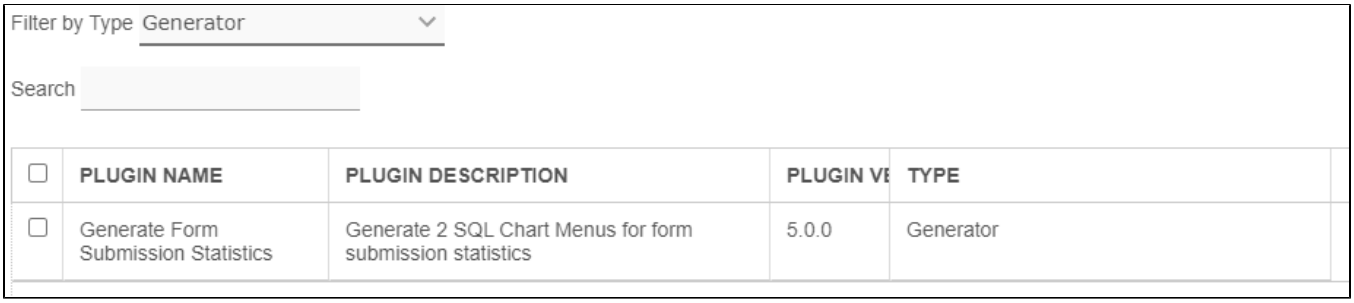
We will have to register our plugin class in Activator class (Auto generated in the same class package) to tell Felix Framework that this is a plugin.

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

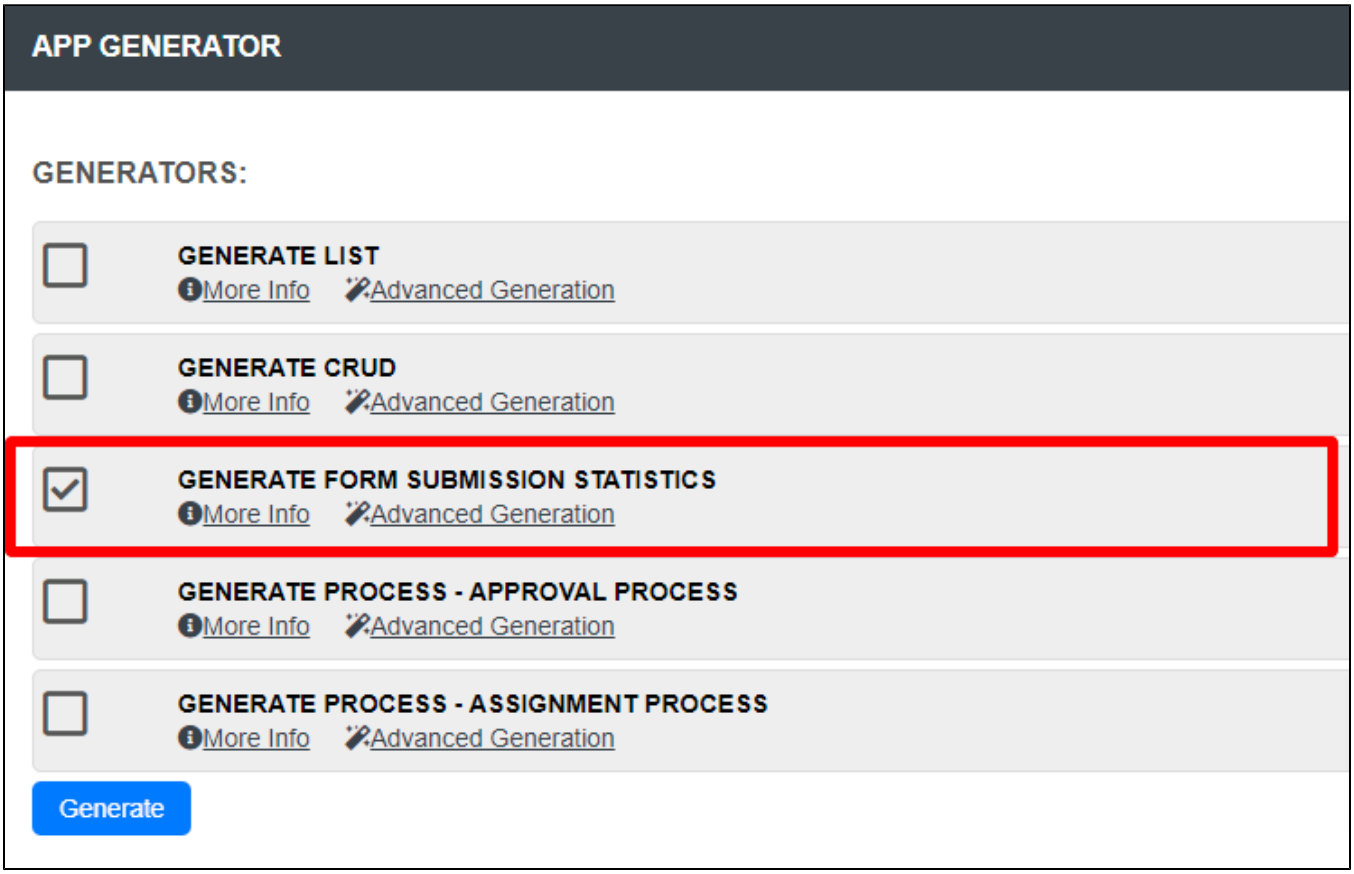
f. Build it and testing

Let build our plugin. Once the building process is done, we will found a "form_submission_statistics_generator-5.0.0.jar" file is created under "form_submission_statistics_generator/target" directory.

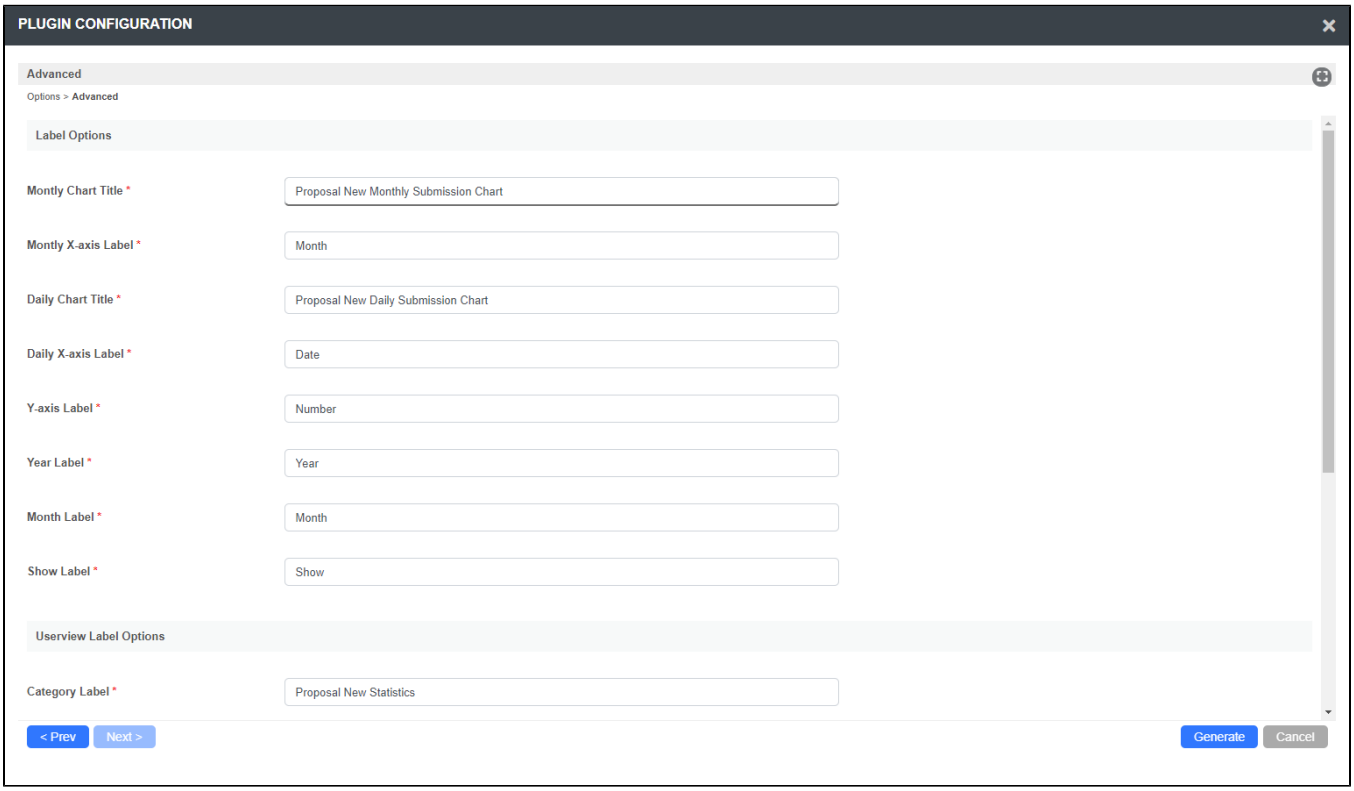
Then, let upload the plugin jar to Manage Plugins. After upload the jar file, double check the plugin is uploaded and activated correctly.



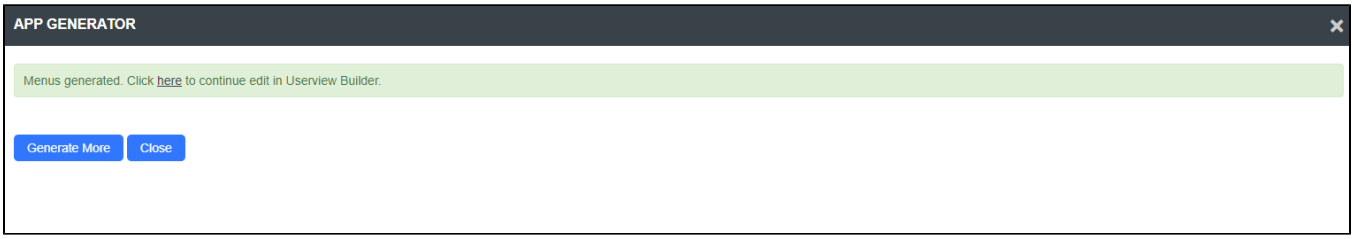
Now, we can open one for our form in Form Builder for testing the generator. Click on the "Generate App" button on right top of the Form Builder. Please refer to Generate App.



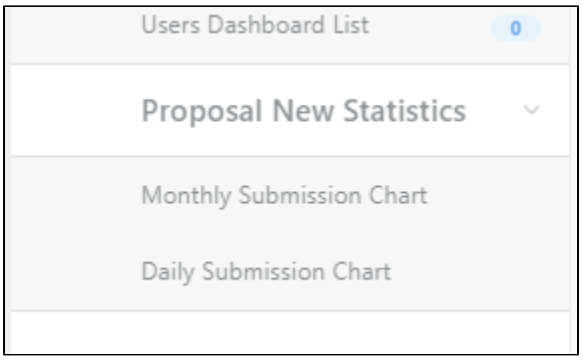
Let us check the properties page for "Advanced Generator".



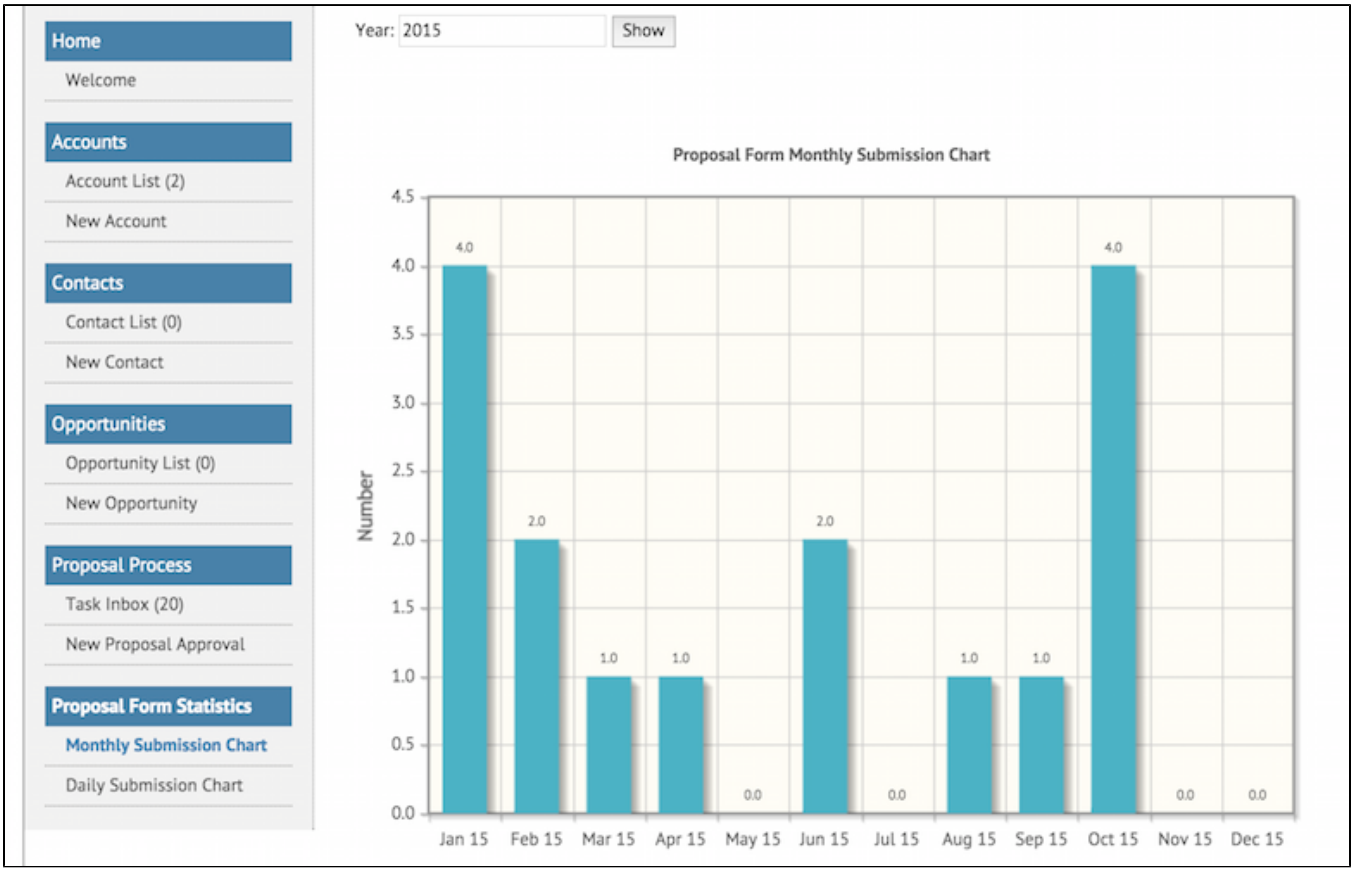
After the generation process is done.

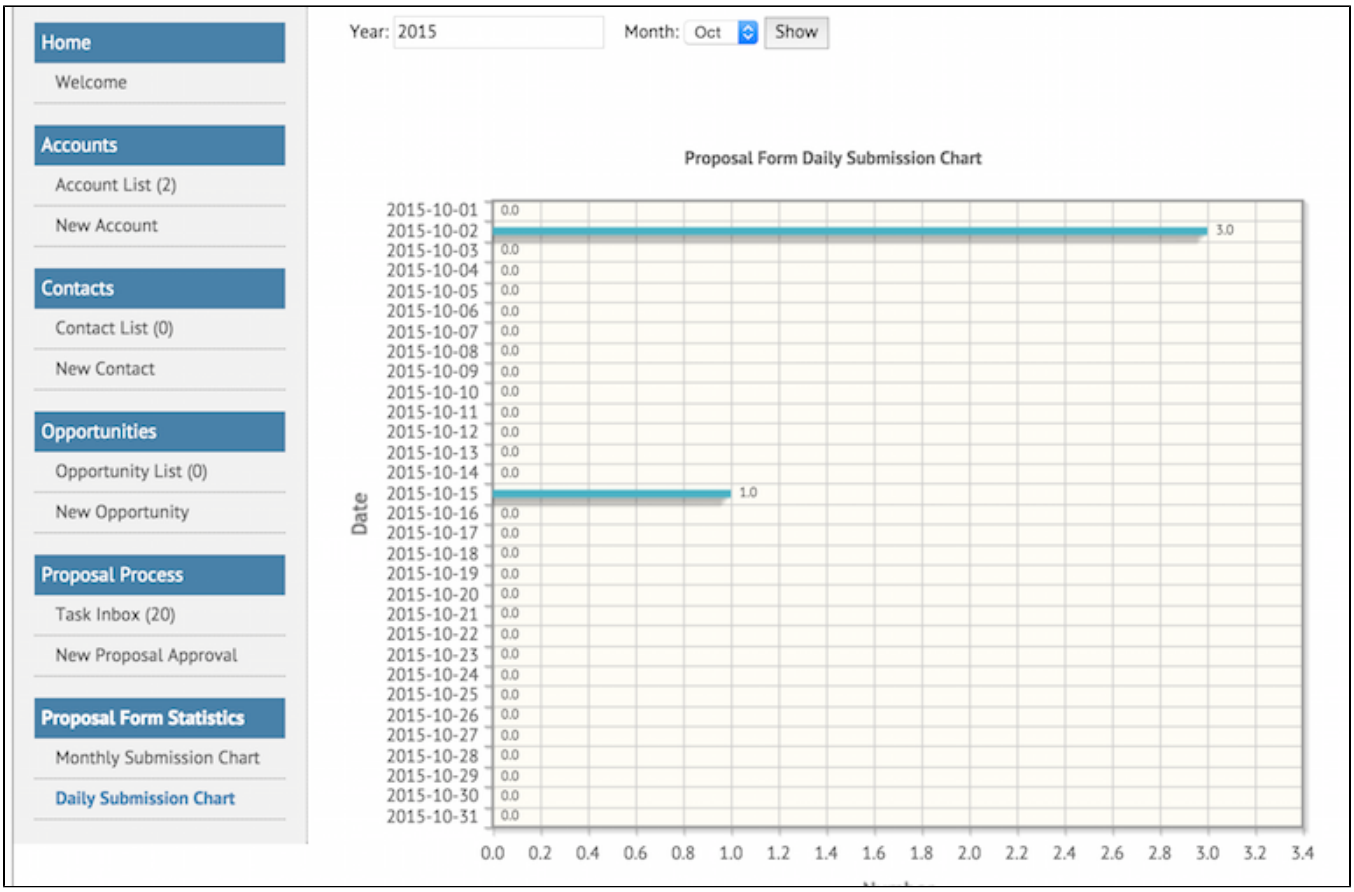


A new category is added to our userview.



The SQL chart menus will shown as following.





8. Take a step further, share it or sell it

You can download the source code from form_submission_statistics_generator.zip.