

Poll
Plug-in Guide
4.2.1-1

Table of Contents

- [1 Introduction.....](#) **3**
- [1.1 Getting Started.....](#) **3**
- [2 Installation.....](#) **4**
- [2.1 Install Poll.....](#) **4**
- [2.2 Verify The Installation.....](#) **4**
- [2.3 Configure The Poll Plug-in.....](#) **5**
- [2.3.1 Set MentometerManager Properties.....](#) **5**
- [2.3.2 Set PollSchedule Properties.....](#) **6**
- [2.3.3 Set PollRedirectUrlValidator and SubmitActionHelper Properties.....](#) **6**
- [3 Adding Polling To a Publication.....](#) **8**
- [3.1 Creating a Poll Content Type.....](#) **8**
- [3.1.1 Displaying Poll Results in CUE.....](#) **9**
- [3.2 Poll REST service for vote submission.....](#) **9**
- [3.3 Poll REST service to fetch poll result.....](#) **10**

1 Introduction

The Poll plug-in allows content creators to easily add **polls** to CUE publications. A **poll** consists of a question and a set of alternative answers. Publication readers can take part in interactive polls by selecting one of the answers.

An CUE poll is a type of content item that contains a title, a question, and a series of possible answers. This makes it very easy for content creators to add polls to publications. To add a poll to a publication, a Content Studio user simply needs to:

1. Create a new poll content item.
2. Fill in the title, question and answer fields.
3. Save and publish the new content item.
4. Add the poll content item to the required section page, or add it to some other content item as related content.

In order to make use of this functionality, a template developer must create a set of one or more content types and associated templates that present polls in the required ways. Template developers can control:

- The overall look and feel of the polls in a publication
- Where polls appear in the publication
- Whether or not poll results are displayed with the poll or are hidden

The Poll plug-in is intended for lightweight, single-question polls. It is not suitable for purposes where secure, certain results are required, since it is not secure enough. Although it does contain some basic protection against multiple voting, this protection is cookie-based and easily circumvented. It is also not really suitable for multiple-question surveys, since it does not include any functionality for processing the results.

1.1 Getting Started

In order to get started with the Poll plug-in in you need to:

1. Install the plug-in. For instructions on how to do this, see [chapter 2](#).
2. Verify that the plug-in is correctly installed. For instructions on how to do this, see [section 2.2](#).

Once you have carried out these basic tasks and are sure that you have a fully-functioning plug-in, you can start adding polling functionality to your own publications. For instructions on how to do this, see [chapter 3](#).

2 Installation

The following preconditions must be met before you can install Poll 4.2.1-1:

- The Content Store is installed and in working order.

2.1 Install Poll

To install Poll on an Ubuntu or other Debian-based system, do the following on your **assembly-host** and on each of your **engine-hosts**:

Installing Poll involves the following steps:

1. Log in as **root**.
2. If necessary, add the CUE **apt** repository to your list of sources:

```
# echo "deb http://user:password@apt.escenic.com stable main non-free" >> /etc/  
apt/sources.list.d/escenic.list
```

where *user* and *password* are your CUE download credentials (the same ones you use to access the CUE Maven repository). If you do not have any download credentials, please contact [CUE support](#).

3. Enter the following commands:

```
# apt-get update  
# apt-get install escenic-poll
```

2.2 Verify The Installation

To verify the status of the Poll plug-in, open the CUE Admin web application (usually located at **http://server/escenic-admin**) and click on **View installed plugins**. The status of all currently installed plug-ins is shown here, and indicated as follows:

The plug-in is correctly installed.

The plug-in is not correctly installed.

So if the Poll plug-in is correctly installed, you should see something like this in the displayed plug-in list:

2.3 Configure The Poll Plug-in

Configuring the Poll plug-in involves a couple of simple tasks, described in the following sections. In these instructions, the placeholder *escenic-config* is used to represent the path of your CUE configuration, as defined with the `com.escenic.config` property in *escenic-home/assemblytool/assemble.properties*. If `com.escenic.config` is not defined, then *escenic-config* has a default definition of *escenic-home/localconfig*.

The general procedure for configuring the Poll plug-in is:

1. If the *escenic-config/com/escenic/poll* folder does not exist, create it.
2. If the *escenic-config/com/escenic/poll* folder does not contain `MentometerManager.properties` and `PollSchedule.properties` files, copy them from *escenic-home/engine/plugins/poll/misc/siteconfig/com/escenic/poll*.
3. Edit `MentometerManager.properties` and `PollSchedule.properties` to meet your requirements, as described in the following sections.

2.3.1 Set MentometerManager Properties

The configuration file *escenic-config/com/escenic/poll/MentometerManager.properties* can be used to set the following properties:

`checkCookieEnabled`

This property is set to `false` by default, which means that all poll visitors can vote an unlimited number of times. If you set `checkCookieEnabled` to `true`, then:

- Visitors are only allowed to vote if they have cookies enabled in their browser.
- Visitors are only allowed to vote once in any one poll.

It may be useful to leave this property set to `false` during development, but for production purposes you will usually want to set it to `true`.

To set this property to true, add the following line to `MentometerManager.properties`:

```
| checkCookieEnabled=true
```

`fieldNameAnswer`

You can use this property to set the prefix to be used for poll alternative field names. By default it is set to `option`, so that the alternative fields in your poll content types must be called `option1`, `option2`, `option3` etc. (see [section 3.1](#)). If you want to set the prefix to something that is more meaningful to you (`alt`, for example), add the following line to `MentometerManager.properties`:

```
| fieldNameAnswer=alt
```

`fieldNameSum`

You can use this property to set the name of the article field for saving the sum of the answers. By default it is set to `sum`. If you want to set the field name to something that is more meaningful to you (`total`, for example), add the following line to `MentometerManager.properties`:

```
| fieldNameSum=total
```

`fieldNameCorrectAnswer`

You can use this property to set the name of the article field for holding the correct answer for use in a competition. By default it is set to `correctAnswer`. If you want to set the field name to

something that is more meaningful to you (**rightAnswer**, for example), add the following line to **MentometerManager.properties**:

```
fieldNameCorrectAnswer=rightAnswer
```

howManyAlternatives

You can use this property to set the maximum number of poll alternative field names. The prefix of this field names should be **fieldNameAnswer** and the suffix for those should start with a number from 1 to the value of **howManyAlternatives**. By default it is set to 20. If you want to set another value (10, for example), add the following line to **MentometerManager.properties**:

```
howManyAlternatives=10
```

If **escenic-config/com/escenic/poll/MentometerManager.properties** does not already exist, then copy **MentometerManager.properties** from **escenic-home/engine/plugins/poll/misc/siteconfig/com/escenic/poll** to **escenic-config/com/escenic/poll/** and set the required properties.

2.3.2 Set PollSchedule Properties

The configuration file **escenic-config/com/escenic/poll/PollSchedule.properties** can be used to set the following properties:

serviceEnabled

This property is set to **false** by default, which means that poll-related data is not stored in the database. This means that all poll-related data will be lost whenever the CUE server is restarted. For production purposes, therefore, you must set this property to **true**. Note, however, that In a multi-server environment it must only be set to true on **one** server.

To set this property to true, add the following line to **PollSchedule.properties**:

```
serviceEnabled=true
```

If **escenic-config/com/escenic/poll/PollSchedule.properties** does not already exist, then copy **PollSchedule.properties** from **escenic-home/engine/plugins/poll/misc/siteconfig/com/escenic/poll** to **escenic-config/com/escenic/poll/** and set the required properties.

2.3.3 Set PollRedirectUrlValidator and SubmitActionHelper Properties

The configuration files **PollRedirectUrlValidator.properties** and **SubmitActionHelper.properties** are used to configure a whitelist of valid redirection targets for the Poll plug-in, in order to protect end-users from phishing attacks in which they are redirected to a malicious web site.

The configuration file **escenic-config/com/escenic/poll/PollRedirectUrlValidator.properties** has the following property:

validRedirectUrlHosts

A comma-separated list of host names that the Poll plug-in is allowed to redirect to. For example:

```
validRedirectUrlHosts=escenic.com,ccieurope.com
```

If `escenic-config/com/escenic/poll/PollRedirectUrlValidator.properties` does not already exist, then copy `PollRedirectUrlValidator.properties` from `escenic-home/engine/plugins/poll/misc/siteconfig/com/escenic/poll` to `escenic-config/com/escenic/poll/` and set the above property.

The configuration file `escenic-config/com/escenic/poll/SubmitActionHelper.properties` can contain one or more properties like this:

`redirectUrlValidator.publication-name`

where *publication-name* is the name of one of your publications. Create one copy of this property for each publication that makes use of the Poll plug-in and set it to point to your redirect URL validator component. For example:

```
redirectUrlValidator.dailynews=./PollRedirectUrlValidator
redirectUrlValidator.sportsweekly=./PollRedirectUrlValidator
```

If `escenic-config/com/escenic/poll/SubmitActionHelper.properties` does not already exist, then copy `SubmitActionHelper.properties` from `escenic-home/engine/plugins/poll/misc/siteconfig/com/escenic/poll` to `escenic-config/com/escenic/poll/` and set the above properties.

3 Adding Polling To a Publication

To add polling functionality to a publication, you need to:

- Create a poll content type (or add polling functionality to an existing content type).
- Create templates for displaying the poll questions and results in the publication.

Once you have done this you should be able to create poll content items

3.1 Creating a Poll Content Type

To create a poll content type, you must edit your publication's **content-type** resource, which is located in the publication's **META-INF/escenic/publication-resources** folder. The following example shows a simple poll content type similar to the one used in the **poll-demo** publication. It defines a field group containing a title field, a question field and three alternative answer fields. It then defines a content type which places this group of fields in the default panel.

```
<field-group name="poll">
  <field mime-type="text/plain" type="basic" name="title">
    <ui:label>Title</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="question">
    <ui:label>Question</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option1">
    <ui:label>Alternative 1</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option2">
    <ui:label>Alternative 2</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option3">
    <ui:label>Alternative 3</ui:label>
  </field>
</field-group>

<content-type name="poll">
  <panel name="default">
    <ref-field-group name="poll"/>
  </panel>
</content-type>
```

The result of adding this code to the **content-type** resource will be that a new content type will be added to the publication and made available to CUE users. Note the following:

- The above code is an example and deliberately kept short. A real polling content type would be likely to have a much larger number of alternative fields, in order to give CUE users the possibility of creating polls with many alternatives when necessary.
- You can give the field containing the poll question any name you like.
- The alternative fields, on the other hand, must (at least by default) have names starting with the prefix **option**. You can change this prefix to something else by editing the configuration file **MentometerManager.properties**. For further information about this, see [section 2.3.1](#).

3.1.1 Displaying Poll Results in CUE

You can very easily get CUE to display the results of polls by adding a single field to your `poll` content type. The field you add must have the following characteristic:

- It must be a **basic** field
- It must have a **mime-type** of **application/xhtml+xml**
- It must have a **result** sub-element with an **enabled** attribute value set to **true**. This element must belong to the namespace **http://xmlns.escenic.com/2014/poll**.

The following example shows such a `poll` with a result field (highlighted):

```
<field-group name="poll">
  <field mime-type="text/plain" type="basic" name="title">
    <ui:label>Title</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="question">
    <ui:label>Question</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option1">
    <ui:label>Alternative 1</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option2">
    <ui:label>Alternative 2</ui:label>
  </field>
  <field mime-type="text/plain" type="basic" name="option3">
    <ui:label>Alternative 3</ui:label>
  </field>
  <field name="pollresult" type="basic" mime-type="application/xhtml+xml">
    <poll:result enabled="true" />
  </field>
</field-group>

<content-type name="poll">
  <panel name="default">
    <ref-field-group name="poll"/>
  </panel>
</content-type>
```

3.2 Poll REST service for vote submission

Using poll REST voting service you can submit a poll. The vote should be submitted to the following URI.

http://your-host/poll-ws/poll/poll-id/vote

In above URI pattern you should use your host name and a valid poll id. This service requires only one form parameter named `option` which is the voting option. If poll is found and voting option is valid then vote will be submitted successfully and REST service will return HTTP status code 200. If poll is not found or voting option is invalid then it will return 404 status code.

Here voting restriction is not handled. Your REST client should handle that.

3.3 Poll REST service to fetch poll result

A REST service is available with poll plugin that will provide poll result details as **JSON** format. With poll plugin a web application named **poll-ws.war** is provided that contains the REST services. To get result for a specific poll you have to send request to the following URI.

http://your-host/poll-ws/poll/poll-id/status

In the above example you must use your host name where poll web service application is deployed and also must use a valid poll id. For a malformed id such as **55Y**, the service will return a 400 status code. If the poll is not found then a 404 status is returned. If the poll id is valid and content is found then it returns the poll title, other fields and options in JSON format. For example:

```
{
  "id": "2742",
  "title": "Euro 2012 quiz",
  "fields": {
    "description": "Count your vote",
    "question": "Who will win Euro 2012 ?"
  },
  "options": [
    {
      "option": option1,
      "value": "Spain",
      "vote": 2,
      "percentage": "25"
    },
    {
      "option": option2,
      "value": "Germany",
      "vote": 4,
      "percentage": "50"
    },
    {
      "option": option3,
      "value": "Portugal",
      "vote": 1,
      "percentage": "12"
    },
    {
      "option": option4,
      "value": "Italy",
      "vote": 1,
      "percentage": "12"
    }
  ]
}
```