LG Polls v1.0.6

Add AJAX polling to ExpressionEngine entries. Now with Pie Charts!

Add polls to your site today!

After months of development and testing LG Polls is finally availableto the public. Purchase a commercial license for single domain (unlimited sub-domains) and recieve unlimited email support and contribute to independent ExpressionEngine development.

Only ~~$79.95~~ $39.95 (holiday special) per license. Register to purchase!

Enjoy LG Polls v1.0.6?
Bookmark and share it with others.

Features

LG Polls is an advanced ajax enabled polling module for ExpressionEngine.

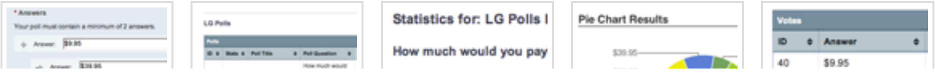
Included in the package is a module and two extensions which integrate LG Polls into the ExpressionEngine administration. Polls are created using the standard weblog entry publish field and can therefore be managed like all other weblog entries.

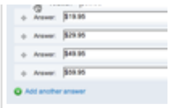
LG Polls is extremely customizable via templates and css styles and there are tons of options for you to choose to ensure that LG Polls runs the way you wanted.

Feature Summary:

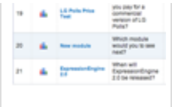
- Create polls as weblog entries allowing polls to be manipulated like standard weblog entries
- Create relationships between polls and other entries using ExpressionEngine relationships
- AJAX enabled poll submission (optional)
- Detailed poll information with powerfull sorting via the EE Control Panel.
- Customise poll templates and results using standard ExpressionEngine templates
- Restrict polls by member group
- Poll details stored in an easy to use custom field type

Screenshots

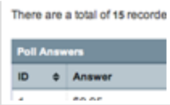




Poll publish form

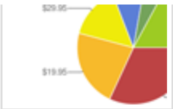


List of polls



There are a total of 15 records

Easy to digest stats



Display results as a bar or pie chart

39	$9.95
38	$19.95
37	$19.95
36	$19.95

Edit poll votes

Configuration

Simple module configuration

Per poll settings

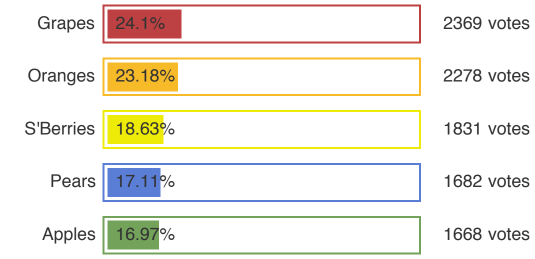
Demo

Actions speak louder than words so feel free to test LG Polls below. This demonstration uses AJAX for vote submission, allows non-members to vote and restricts all users to just one vote. The bar chart is created using standard CSS and XHTML while the pie chart is displayed using the Google Charts API.

Which is your favourite fruit?

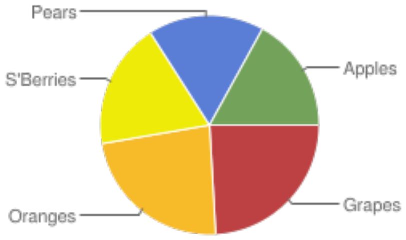
○ Apples ○ Pears ○ S'Berries ○ Oranges ○ Grapes **Submit**

Bar Chart Results



Total Votes: 9829

Pie Chart Results



Total Votes: 9829

Requirements

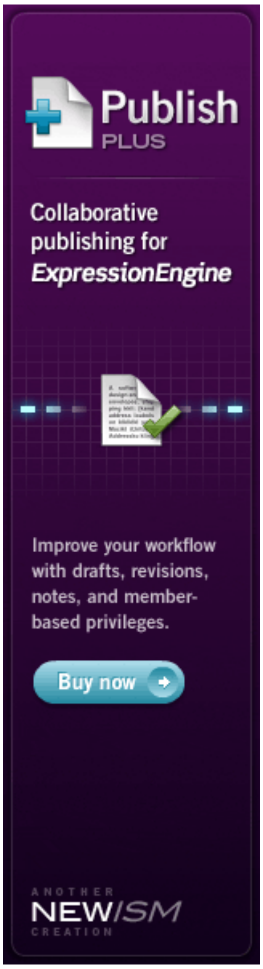
LG Polls is an ExpressionEngine module and extension which requires ExpressionEngine 1.6+.

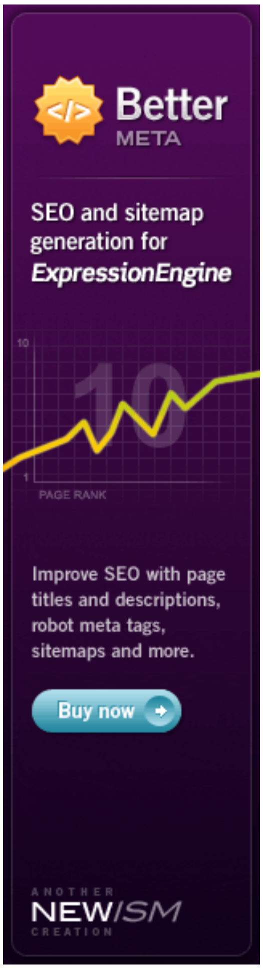
The administration of LG Polls also requires javascript to be enabled.

Installation

The LG Polls package contains a module and extension folders and a language file. To install the package follow the instructions below:

1. Purchase LG Polls and download the latest version of the extension.
2. Extract the .zip file to your desktop
3. **Open** the Extensions Manager
4. **Enable Extensions** if not already enabled
5. **Copy** the `extensions/ext.lg_polls_controller.php` file to your `/system/extensions` directory
6. **Copy** the `extensions/ext.lg_polls_question.php` file to your `/system/extensions` directory





Page created on: Dec 06, 2007
Last updated on: Sep 27, 2008

7. **Copy** the `language/english/lang.lg_polls.php` file to your `/system/languages/english` directory
8. **Copy** the `modules/lg_polls` folder to your `/system/modules` directory

Activation

The LG Polls module and extensions must first be activated before any polls can be created.

1. Open the Module Manager
2. **Install** the *LG Polls* module. This will also activate the extensions copied in the installation.

Setting Up

Now that the LG Polls module and extensions have been activated a dedicated poll weblog with two required custom fields is required. These custom fields store the poll question and the poll details (answers and per poll settings).

A new custom field group

1. Create a new custom field group called *LG Polls Custom Weblog Group*
2. Add the following custom fields to the new custom field group:
    1. Poll Question:
        - Field Name: poll_question
        - Field Label: Poll Question
        - Field Type: LG Polls: Poll Question
        - Required: Yes
        - Searchable: No
        - Show by default: Yes
        - Default Text Formatting for This Field: None
    2. Poll Details:
        - Field Name: poll_details
        - Field Label: Poll Details
        - Field Type: LG Polls: Poll Details
        - Required: No
        - Searchable: No
        - Show by default: Yes
        - Default Text Formatting for This Field: None

A new poll weblog

1. Create a new weblog called *LG Polls*
2. Assign the *LG Polls Custom Weblog Group* to the new weblog
3. Save the *LG Polls* weblog

Congratulations, the LG Polls module and extensions have now been successfully setup and are ready to be configured.

Module Configuration

LG Polls allows you to configure which member groups can vote, how to submit and when to display poll results and define poll and result templates.

The configuration has five sections:

Weblog - Assigning A polls weblog

Poll Weblog [required]

```
List of existing weblogs
```

Choose which weblog stores your poll entries (the weblog you created in the setup). This setting is used to create contextual links in the module administration.

Security - Who can vote and when

Allow users to vote more than once [new poll default]

```
yes | no
```

Allows users to vote more than once in a poll. Note: Duplicate votes may skew your results.

Check for duplicate votes against users [global setting]

```
Cookie | IP Address | Member ID
```

When a user votes LG Polls can check their cookies, IP address, and/or member id for duplicate votes.

Voting Groups [new poll default]

```
List of existing member groups
```

Voting in polls can be restricted to certain member groups. Set which groups will be allowed to vote in new polls by default.

Displaying Results

Display results in which order? [new poll default]

```
Least Votes First | Most Votes First | Same order as the answers | Random
```

Results can be displayed most votes first, least votes first, in the same order as the questions or in random order.

Show results [new poll default]

```
Before vote | After vote | After Poll | Never show results
```

Results can be shown before the user votes, after they have voted, after a poll has expired or never at all.

Vote Submission

Submit poll votes using AJAX (jQuery) [global setting]

```
yes | no
```

Submit votes using AJAX. LG Polls uses jQuery for all its ajaxy goodness. If you wish to use another library disable this setting and refer to the submitting votes using ajax section of this documentation.

Path to jQuery core library file [global setting]

```
/system/modules/lg_polls/scripts/jquery-1.2.1.pack.js
```

Set the path to the jQuery core library file. If your site uses jQuery elsewhere you can link to the existing jQuery file.

Auto Updates

Check for updates [required]

```
Yes|No
```

LG Polls can check leevigraham.com automatically for updates to the module. A short message will be displayed on your ExpressionEngine control panel homepage if an update is available.

Cache Refresh [required]

```
1200
```

The number of minutes between extensions update checks. This number should be quite large in order to speed up delays when loading the homepage.

Usage

Now that LG Polls has been installed and configured correctly in the ExpressionEngine administration it is ready for use.

Creating a new poll

A poll entry is just like any other weblog entry. Polls can have start and end dates, be categorized, have different statuses and be tagged. You can also attach associate a poll to another entry using relationships.

To create a new poll follow these simple steps:

1. *Browse* to the LG Polls module index
2. *Press* the **Create new poll** button. A new publish form will appear.
3. Fill in your poll details adding at least two answers
4. If necessary update the configuration for this poll
5. *Press* the **Save** button

Now that a new poll has appeared there will be a new poll in the LG Polls module index.

Displaying Polls

As previously mentioned in creating a new poll above a LG Poll is just a weblog entry. Polls must be displayed inside a `{exp:weblog:entries}` tag and use `{exp:lg_polls:poll}` to render the poll form and results.

The poll template must accommodate a wide range of conditions from banned users to previously submitted votes.

The example below loops over all the published polls including those that have expired and those that yet to start. The `{exp:lg_polls:poll}` tag accepts entry id of the weblog entry and retrieves the polls details and configuration options.

```
{exp:weblog:entries
  weblog="polls"
  show_future_entries="yes"
  show_expired="yes"
}
  {exp:lg_polls:poll
    entry_id="{entry_id}"
    precision="2"
    return="/"
  }
    <h1>{poll_question}</h1>
    {if can_vote}
      {poll_form}
        {if has_voted}<p>You have already voted in this poll, however you can vote again.</p>{/if}
        <ul class='lg-polls-answers'>
          {poll_answers}
          <li class='a-{answer_count}'>
            <label for='lg-polls-answer-{answer_id}'>{answer_input} <span class='answer'>{answer}</s
          </li>
          {/poll_answers}
        </ul>
        <input type='submit' value='submit' />
      {/poll_form}
    {if:else}
      {if has_voted}<p>Thanks for voting in this poll.</p>{/if}
      {if restricted}<p>Sorry, You are restricted from voting in this poll.</p>{/if}
      {if expired}<p>This poll ended on {expiration_date}.</p>{/if}
      {if yet_to_begin}<p>This poll is yet to begin. Voting opens on {entry_date}.</p>{/if}
    {/if}
    {if show_results}
      <div class='lg-poll-results' id='lg-poll-results-{entry_id}'>
        <ul class='lg-polls-answers'>
        {results_answers}
        <li class='a-{answer_count}'>
          <span class='answer'>{answer}</span>
          <span class='answer-total-votes'>{answer_total_votes} votes</span>
          <div class='percentage' style='width:{answer_percentage}%'><small>{answer_percentage}%</sm
        </li>
        {/results_answers}
        </ul>
        <div class='poll-total-votes'>Total Votes: {poll_total_votes}</div>
      </div>
    {if:else}
      {if show_results_after_poll && has_voted}The results of the poll will be made available on {ex
      {if never_show_results && has_voted}The results of this poll will be made public at a later da
    {/if}
  {/exp:lg_polls:poll}
{/exp:weblog:entries}
```

Submitting Polls with AJAX

Submitting polls with AJAX is easy with LG Polls although it does require a small change to your templates. There will be three templates needed to successfully submit the poll and get a quick AJAX response. They are:

- Poll template
- Main template
- AJAX template

Poll Template

The poll template is just a simple embedded template that will be rendered inside the main template and returned by AJAX in the AJAX Template.

The poll template is very similar to the template example in displaying polls but makes use of embedded variables to pass through the poll `entry_id` and any custom fields.

Below is an example of a Poll Template located in _components/_poll using embedded variables:

```
{exp:lg_polls:poll
  entry_id="{embed:entry_id}"
  precision="{embed:precision}"
  ajax_return="{embed:ajax_return}"
  return="{embed:return}"
}
  <div id='loading' style='display:none;'>Loading ... </div>
  <h1>{embed:poll_question}</h1>
  {if can_vote}
    {poll_form}
```

```
      {if has_voted}<p>You have already voted in this poll, however you can vote again.</p>{/if}
      <ul class='lg-polls-answers'>
        {poll_answers}
        <li class='a-{answer_count}'>
          <label for='lg-polls-answer-{answer_id}'>{answer_input} <span class='answer'>{answer}</spa
        </li>
        {/poll_answers}
      </ul>
      <input type='submit' value='submit' />
    {/poll_form}
  {if:else}
    {if has_voted}<p>Thanks for voting in this poll.</p>{/if}
    {if restricted}<p>Sorry, You are restricted from voting in this poll.</p>{/if}
    {if expired}<p>This poll ended on {embed:expiration_date}.</p>{/if}
    {if yet_to_begin}<p>This poll is yet to begin. Voting opens on {embed:entry_date}.</p>{/if}
  {/if}
  {if show_results}
    <div class='lg-poll-results' id='lg-poll-results-{embed:entry_id}'>

      <ul class='lg-polls-answers'>
        {results_answers}
        <li class='a-{answer_count}'>
          <span class='answer'>{answer}</span>
          <span class='answer-total-votes'>{answer_total_votes} votes</span>
          <div class='percentage' style='width:{answer_percentage}%'><small>{answer_percentage}%</sm
        </li>
        {/results_answers}
      </ul>
      <div class='poll-total-votes'>Total Votes: {poll_total_votes}</div>
    </div>
  {if:else}
    {if show_results_after_poll && has_voted}The results of the poll will be made available on {embe
    {if never_show_results && has_voted}The results of this poll will be made public at a later date
  {/if}
{/exp:lg_polls:poll}
```

Notice in the Poll template we have removed the `{exp:weblog_entries}` tag. This tag will still be in the main template. Also an additional #loading div has been added to the poll template to notify the user when a poll is being submitted using AJAX.

Main Template

The main template is the template that first displays your poll. It must include a `{weblog:entries}` and the embedded poll template.

The main template can look as simple as the example below, however it is most likely that you will have other information on your page.

```
<html>
<head>{exp:lg_polls:head}</head>
<body>
  {exp:weblog:entries
    weblog="polls"
    show_future_entries="yes"
    show_expired="yes"
  }
    {embed="_components/_poll"
      entry_id="{entry_id}"
      poll_question="{poll_question}"
      entry_date="{entry_date format="%D, %F %d, %Y - %g:%i:%s"}"
      expiration_date="{expiration_date format="%D, %F %d, %Y - %g:%i:%s"}"
      ajax_return="_components/_ajax_poll/{entry_id}"
      return=""
    }
  {/exp:weblog:entries}
</body>
</html>
```

In the example above the `ajax_return` parameter has been set to the AJAX template path which will render the returned content. The returned content will replace the poll form and results if they are shown.

AJAX Template

The AJAX template is a very basic template containing only one `{exp:weblog:entries}` tag and the embedded Poll Template.

The AJAX template should have a very restrictive set of `{exp:weblog:entries}` parameters to speed up the parsing of the poll entry.

`{post_entry_id}` is a special single variable only available in the AJAX template. It stores the poll entry_id of the posted vote. This variable is used to return the correct poll.

An example AJAX template is given below.

```
{exp:weblog:entries
  weblog="polls"
  entry_id="{segment_3}"
  show_future_entries="yes"
  show_expired="yes"
  limit="1"
  rdf="off"
  disable="categories|category_fields|member_data|pagination|trackbacks"
}
  {embed="_components/_poll"
    entry_id="{entry_id}"
    poll_question="{poll_question}"
    entry_date="{entry_date format="%D, %F %d, %Y - %g:%i:%s"}"
    expiration_date="{expiration_date format="%D, %F %d, %Y - %g:%i:%s"}"
    ajax_return="_components/_ajax_poll/{entry_id}"
  }
{/exp:weblog:entries}
```

Tag Reference

LG Polls: Poll Tag `{exp:lg_polls:poll}`

The `{exp:lg_polls:poll}` renders a single poll and/or poll results based on the poll status, existing votes and user. The output of this tag can be highly customised using tag parameters, special conditionals and single variables.

Parameters

A variety of parameters can be added to control which poll is displayed, how results are formatted and how forms are posted. Parameters are added to the opening partition of the tag. Here is a full example of the weblog tag with three parameters.

```
{exp:lg_polls:poll
  entry_id="{entry_id}"
  precision="2"
  return="/polls/thankyou"
  ajax_return="_components/_ajax_poll"
}
  ...
{/exp:lg_polls:poll}
```

The `{exp:lg_polls:poll}` tag also checks if the embedded value has been passed from any parent templates. If the value has not been passed it will unset the parameter so the `{embed:...}` tag will not be rendered. This feature is especially handy when embedding templates and submitting polls with AJAX.

The following parameters are available:

- entry_id="1"
- precision="3"
- return="/"
- ajax_return="_components/_ajax_poll"

entry_id= [required]

```
entry_id="1"
```

The entry id of the poll. Generally the {exp:lg_polls:poll} tag is wrapped inside a {exp:weblog_entries} tag.

precision= [optional]

```
precision="3"
```

The precision of the answer_percentage in decimal places. Defaults to 2 decimal places eg: 10.51%.

return= [optional]

```
return="/"
```

The return url of submitted poll. Defaults to the same page as where the poll was submitted from. Ignored if the poll is submitted using AJAX.

ajax_return= [optional]

```
return="_components/_ajax_poll"
```

If the poll has been submitted using AJAX the content returned to the browser will be retrieved from the `ajax_return` parameter. See Submitting Polls with AJAX for more information.

Conditional Variables

Conditional variables allow you to control what content is displayed to the user. The following conditional variables are available:

- {if can_vote}
- {if has_voted}
- {if restricted}
- {if expired}
- {if yet_to_begin}
- {if show_results}
- {if never_show_results}
- {if show_results_after_poll}

if can_vote

```
{if can_vote}
  Render Poll
{if:else}
  Notify User
{/if}
```

This condition allows you to display the poll template if the user can vote or alternative content if they cannot.

The ability to vote is determined by:

- The user is not banned
- The user is not from a banned nation
- The user has not been black listed
- The user is a member of one of the allowed voting member groups
- Has the poll started and not yet expired
- The poll has not expired
- The poll has begun
- The poll allows duplicate votes

If all the above conditions have been met the user is allowed to vote.

if has_voted

```
{if has_voted}
  You have already voted in the poll.
{/if}
```

```
{if has_voted and can_vote}
  You have already voted in this poll however duplicate votes are allowed.
{/if}
```

This conditional is used to show a custom message if the user has already voted in this poll.

if restricted

```
{if restricted}
  Sorry, you are not allowed to vote in this poll.
{/if}
```

This conditional is used to show a custom message if the user is restricted from voting in the poll.

The user is restricted if:

- The user is banned
- The user is from a banned nation
- The user has been black listed
- The user is not a member of one of the allowed voting member groups

if expired

```
{if expired}
    This poll closed on {expiration_date format="%D, %F %d, %Y - %g:%i:%s"}
{/if}
```

This conditional is used to show a custom message if the poll has finished. The poll expiration date is the entry expiration date therefore date variable formatting is available.

if yet_to_begin

```
{if yet_to_begin}
    This poll is yet to begin. Voting opens on {entry_date format="%D, %F %d, %Y - %g:%i:%s"}
{/if}
```

This conditional is used to show a custom message if the poll is yet to begin. The poll starting date is the entry published date therefore date variable formatting is available.

if show_results

```
{if show_results}
  {results_template}
{if:else}
  user notifications
{/if}
```

Results will be shown if **ANY** of the following conditions are met:

- The polls results mode is set to *'Show before vote'*
- The polls results mode is set to *'Show after vote'* and the user has voted
- The polls results mode is set to *'Show after poll'* and the poll has finished

and **ALL** of the following conditions are met:

- The poll has begun
- The user is not restricted
- The polls result mode is not *'Never show results'*

if never_show_results

```
{if never_show_results}
  The results of this poll will be made public at a later date. Stay tuned.
{/if}
```

This conditional allows you to render a custom message if the polls results mode is set to *'Never show results'*.

if show_results_after_poll

```
{if show_results_after_poll}
  The results of the poll will be made available on {expiration_date format="%D, %F %d, %Y - %g:%i:
{/if}
```

This conditional allows you to render a custom message if the polls results mode is set to *'Show after poll'*.

Variable Pairs

- {poll_form}
- {poll_answers}
- {results_answers}

{poll_form} ... {poll_form} [required]

```
{poll_form}
  <ul class='lg-polls-answers'>
    {poll_answers}
    <li class='a-{answer_count}'>
      <label for='lg-polls-answer-{answer_id}'>{answer_input} <span class='answer'>{answer}</span><
    </li>
  {/poll_answers}
  </ul>
  <input type='submit' value='submit' />
{/poll_form}
```

Wraps the contained content in an ExpressionEngine `<form>` tag. The form tag contains important information about the poll and adds XSS security if applicable.

{poll_answers} ... {poll_answers} [required]

```
{poll_answers}
<li class='a-{answer_count}'>
        <label for='lg-polls-answer-{answer_id}'>{answer_input} <span class='answer'>{answer}</span
</li>
{/poll_answers}
```

The content of this tag is looped over for every poll answer in the same order as set in the poll publish form. See Single Variables for a list of the nested single variables available.

{results_answers} ... {results_answers} [required]

```
{results_answers}
<li class='a-{answer_count}'>
        <span class='answer'>{answer}</span>
        <span class='answer-total-votes'>{answer_total_votes} votes</span>
        <div class='percentage' style='width:{answer_percentage}%'><small>{answer_percentage}%</sma
</li>
{/results_answers}
```

The content of this tag is looped over for every poll answer in the order set in the poll configuration. See Single Variables for a list of the nested single variables available.

Single Variables

As previously mentioned polls are just weblog entries with extra data. Therefore all the standard weblog entry variables and custom fields are available.

The following variables can be nested within the `{results_answers}` and `{poll_answers}` variable pairs.

- {answer_input}
- {answer}
- {answer_id}
- {answer_count}
- {answer_percentage}
- {answer_total_votes}

The following variables can be nested anywhere inside the `{exp:lg_polls:poll}` tag.

- {poll_total_votes}
- {pie_chart}

The following variables are only available in the AJAX template.

- {post_entry_id}

{answer_input} [required in `{poll_answers}`]

Replaced with a radio button.

{answer} [nested inside `{poll_answers}` & `{results_answers}`]

Replaced with the answer option. Used in both the polls and results template

{answer_id} [nested inside `{poll_answers}` & `{results_answers}`]

Replaced with the answers id.

{answer_count} [nested inside `{poll_answers}` & `{results_answers}`]

Replaced with the current answer count. Starts at 0.

{answer_percentage} [nested inside `{poll_answers}` & `{results_answers}`]

Replaced with the percentage of total votes the current option has received. Recommended to be used in the results poll.

{answer_total_votes} [nested inside `{poll_answers}` & `{results_answers}`]

Replaced with the number of total votes the current option has received. Recommended to be used in the results poll.

{poll_total_votes}

Replaced with the number of total votes for the current poll.

{post_entry_id}

This is a special single variable that stores the poll entry_id of a posted vote. It can only be used in an AJAX template.

LG Polls: Chart Tag `{exp:lg_polls:chart}`

The `{exp:lg_polls:chart}` renders a 2d or 3d pie chart of the results. The output of this tag can be highly customised using tag parameters, single variables and variable pairs. Your chart tag must be inside the {exp:lg_polls:poll} tag.

Parameters

A variety of parameters can be added to control the rendering of the chart. Parameters are added to the opening partition of the tag. Here is a full example of the chart tag with five parameters.

```
{exp:lg_polls:chart
  entry_id="{entry_id}"
  colors="BD4142|F7BA29|EFEB08|5A7DD6|73A25A|9CCB21"
  segment_label_format="{answer_percentage}%"
  size="300x200"
  type="p"}
    <img src="{chart_url}"  />
{/exp:lg_polls:chart}
```

The `{exp:lg_polls:chart}` tag also checks if the embedded value has been passed from any parent templates. If the value has not been passed it will unset the parameter so the `{embed:...}` tag will not be rendered.

The following parameters are available:

- entry_id="1"
- colors="BD4142|F7BA29|EFEB08|5A7DD6|73A25A|9CCB21"
- segment_label_format="{answer_percentage}%"
- size="300x200"
- type="p"

entry_id= [required]

```
entry_id="1"
```

The entry id of the poll. Generally the {exp:lg_polls:chart} tag is wrapped inside a {exp:weblog_entries} tag.

colors= [optional]

```
colors="BD4142|F7BA29|EFEB08|5A7DD6|73A25A|9CCB21"
```

A pipe delimited list of hex colors. Its advisable to have a long list of custom colors if you are using this paramter. As a safety if there are more answers than colors the colors will be repeated.

segment_label_format= [optional]

```
segment_title_format="{answer_percentage}%"
```

Each segment of the chart can have its own title generally the answer. However in some cases the answer string will be to long to display which is when the `segment_label_format` should be used.

This parameter will replace the following strings: `{answer_id}`, `{answer}`, `{answer_percentage}` and `{answer_total_votes}` with the corresponding answer value.

If this parameter is not included the chart will have no segment labels. No labels can be used in conjunction with a chart legend.

size= [optional]

```
size="300x200"
```

The size of the chart in pixels in the format of *width*x*height*. When displaying 3D charts the width will generally need to be wider than 2D charts.

type= [optional]

```
type="p"
```

Currently only two types of pie charts are supported; 2D and 3D. For 2D charts use `type="p"` or `type="p3"` for 3D charts.

Variable Pairs

- {legend_answers}

{legend_answers} ... {legend_answers} [optional]

```
<ul>
  {legend_answers}
    <li id='key-{answer_id}'>
      <span style='background:#{answer_color}'>{answer_color}</span> {answer} \ {answer_total_votes
    </li>
  {/legend_answers}
</ul>
```

A legend can be displayed below a chart using the `{legend_answers}` variable pair. This is especially useful when the chart has no segment labels.

The following standard answer loop single variables are available in the `{legend_answers}` tag:

- {answer}
- {answer_id}
- {answer_count}
- {answer_percentage}
- {answer_total_votes}

Additionally legend specific variables are available in this variable pair.

- {answer_color}

Single Variables

The following variables can be nested within the `{legend_answers}` variable pair.

- {answer}
- {answer_id}
- {answer_count}
- {answer_percentage}
- {answer_total_votes}
- {answer_color}

The following variables can be nested anywhere inside the `{exp:lg_polls:chart}` tag.

- {chart_url}

{answer_color} [nested inside `{legend_answers}`]

A hex color value corresponding to the chart segement.

{chart_url}

The src url for the generated chart image.

LG Polls: Head Tag `{exp:lg_polls:head}`

This tag is an required tag for those who wish to use AJAX. It is replaced with the jQuery core library file and an AJAX submission script. If you do not want to use this tag you can hardcode the paths into your template and move the required files. The scripts can be found in `modules/lg_polls/scripts/`.

Note: This tag will expose your system folder. If you are conscious about security and don't mind having two jQuery files lying around (one for the admin and one for the frontend) I recommend the following:

- Copy `system/modules/lg_polls/scripts/jquery-1.2.1.pack.js` into your theme folder
- Move `system/modules/lg_polls/scripts/lg_polls.js` into your theme folder
- Hardcode the paths to these two files into your templates

Change Log

Beta 1

- Initial release for selected Beta testers

1.0.1

- Commercial Release
- Documentation Update
- PHP4 Constructor bug fix

1.0.3

- Fixed issue with paths for ExpressionEngine installations in subdomains and masked administrations.

1.0.4

- Fixed entry preview issue.

1.0.6

- Fixed javascript path issue in the CP
- Added auto update notifications

License

LG Polls is a commercial product and therefore its usage is subject to the commercial license agreement.

Powered by: ExpressionEngine
Hosted with Dreamhost
Save $80 on hosting | Promo code: **saveheaps**.

Unless stated otherwise any theme, plugin, module, extension or documentation is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement of third party rights. In no event shall the copyright holder or holders included in this notice be liable for any claim, or any special indirect or consequential damages, or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.