µWeb - Feature #884

Arguments on template functions

2012-06-26 12:43 - Niek Bergman

Status:	Closed	**Start date:**	2012-06-26
Priority:	Normal	**Due date:**	
Assignee:	Niek Bergman	**% Done:**	100%
Category:	TemplateParser	**Estimated time:**	0.00 hour
Target version:	µWeb alpha release	**Spent time:**	8.50 hours

Description

Recently a possible use case for template arguments was established:

A template tag could be constructed which would limit the length to X characters including Y dots at the end.

For example:

data = "This is a long line that will be cut off so it will not exceed 70 characters."

[data|limitlength] would then return e.g.: "This is a long line that will be cut off so it will not exceed 70 c..."

Possible optional arguments for this template tag would be:

Specify how many characters (including dots at the end) the final data should be.
Specify how many dots there should be at the end.

Currently, several template tags would have to be constructed for an use case like this, one for each different setting desired.
Optional template tag arguments (with a possibility to have defaults) would be useful in such a use case.

Associated revisions

Revision 255:ba7abb242ae6 - 2012-07-05 18:21 - Elmer de Looff

Added capabilities and tests for function closures on tags. Restructured the regular expression to better match multiple indices and functions. This resolves #884.

History

#1 - 2012-06-26 13:00 - Elmer de Looff

- Subject changed from Template tag arguments. to Arguments on template functions

- Category set to TemplateParser

What this would require is a way of providing arguments to template tag functions. This is possible, but introduces some problems with optional *named* arguments, which need to be parsed, as well as all sorts of characters that might occur in the argument. Simple numerics are easy, but strings are significantly trickier to handle in the regex and processing code.

Basic template tag extension could be something like this: '[tag|limit(80)]' for a limit function with an argument of 80. Functions that accept arguments should return closures, though this means that using the above limit function with the default values will still need an invocation limit() because the function itself is not directly usable.

The basic mode of operation for this could be something like this (inside a PageMaker method):

```
def LengthLimitedReponse(self):
  def LengthLimiter(maxlen=80):
    def _Limited(string, maxlen=maxlen):
      return string[:maxlen]
    return _Limited

  self.parser.RegisterFunction('limit', LengthLimiter)
  return self.parser.ParseString('[string|limit(80)]', string='hello world, ' * 20)
```

The result for this function would be 'hello world, hello world, hello world, hello world, hello world, hello world, he'

#2 - 2012-06-26 13:02 - Jan Klopper

Seems ok to me, is it easily added? including unittests?

#3 - 2012-07-05 18:22 - Elmer de Looff

- Status changed from New to Resolved

- % Done changed from 0 to 70

Applied in changeset ba7abb242ae6.

#4 - 2012-07-05 18:27 - Elmer de Looff

- Status changed from Resolved to Closed

- Assignee changed from Elmer de Looff to Niek Bergman

- Target version set to µWeb alpha release

- % Done changed from 70 to 100

Done and done, with some limitations:

- Arguments are passed as strings, meaning the closure functions need to do some argument-mangling to get them to the expected type;
- Multiple arguments are possible, but every comma counts as an argument separator, even those inside quoted or parenthesized strings;
- Leading whitespace is restricted as the splitting is done on commas and any following whitespace.