

UtfErrors - Errors: 15

Demo of some UTF8 errors

[UtfErrors - Errors:](#)

- [Demo of some UTF8 errors](#)
- [Json output from a QUERY áěšňž](#)
- [Regex QUERY - fails for character classes and case comparisons too...](#)
- [QUERY lc/uc - fails](#)
- [SEARCH with regex containing character classes such w](#)
- [Calc ERRORS \(many string operation - examples \(here are more\)\)](#)
- [Non critical \(but annoying\) errors](#)

[Strange Errors](#)

[EDITTABLE](#)

[Demo utf8 topic text](#)

[Mixed ASCII and Utf8](#)

[Sĩmĩĩĩqũē éţ äűţ ạấsủmẻndả](#)

[Qùí Ĵăbôřřôšám đøłôřŭm](#)

[Words containing NO ASCII](#)

[List for the SEARCH test](#)

[JSchecker](#)

[Comments](#)

Json output from a QUERY áěšňž

Testname	Wanted	Actual	Result
Q-plain	ĩæĉq̃mėňă, ĴĚĈĂMĚŇĂ, ĴæĈq̃mėňă, Ĵæĉq̃mėňă	ĩæĉq̃mėňă, ĴĚĈĂMĚŇĂ, ĴæĈq̃mėňă, Ĵæĉq̃mėňă	OK
Q-json	["ĩæĉq̃mėňă", "ĴĚĈĂMĚŇĂ", "ĴæĈq̃mėňă", "Ĵæĉq̃mėňă"]	["Ā-Ā, ĀĀá'ĀĀĀ£", "ĀĀĀĀá'ĀĀĀ", "ĀĀ, ĀĀá'ĀĀĀ£", "ĀĀ, ĀĀá'ĀĀĀ£"]	ERROR

Regex QUERY - fails for character classes and case comparisons too...

Testname	Wanted	Actual	Result
asc-re2: /AbCdefgh/ =~ /wCde\w/	1	1	OK
asc-re3: /AbCdefgh/ =~ /(?)cDE/	1	1	OK
asc-re4: /AbCdefgh/ =~ /[[[:upper:]]/	1	1	OK
utf-re2: /ĴæĈq̃mėňă/ =~ /wĈq̃m\w/	1	0	ERROR
utf-re3: /ĴæĈq̃mėňă/ =~ /(?)ĈĂM/	1	0	ERROR
utf-re4: /ĴæĈq̃mėňă/ =~ /[[[:upper:]]/	1	0	ERROR

QUERY lc/uc - fails

Testname	Wanted	Actual	Result
WTWA1: lc(<u>AbCdefgh</u>)	abcdefgh	abcdefgh	OK
WTWA1: uc(<u>AbCdefgh</u>)	ABCDEFGH	ABCDEFGH	OK
WTWL1: lc(ÌæĈąmėñǎ)	ìæĉąmėñǎ	ÌæĈąmėñǎ	ERROR
WTWL1: uc(ÌæĈąmėñǎ)	ÌÆĈĄMĚŃǍ	ÌæĈąmėñǎ	ERROR

SEARCH with regex containing character classes such lw

Testname	Wanted	Actual	Result
search for <code>\dxx</code> \$pattern regex charclass <code>\w*xx\w*</code>	abc1xxdef,áčž2xxøđß,ábc3xxdeš,qwe4xxuio	dxx,abc1xxdef,2xx,bc3xxde,qwe4xxuio	ERROR
Besides the <code>\w</code> regex error, I actually don't understand why found the string <code>dxx</code> too. The regex contains the <code>\d</code> as a character class.			

Calc ERRORS (many string operation - examples (here are more))

Test	Wanted	Actual	Result
%CALC{"\$FIND(e, abcdefgh, 2)"}%	5	5	OK
%CALC{"\$FIND(ó, šďň óťķų, 2)"}%	5	9	ERROR
%CALC{"\$INSERTSTRING(abcdef, 3, XYZ)"}%	abcXYZdef	abcXYZdef	OK
%CALC{"\$INSERTSTRING(ásčšěž, 3, ÁŠŤ)"}%	ásčÁŠŤšěž	áÁŠŤčšěž	ERROR
%CALC{"\$LEFTSTRING(abcdefg, 5)"}%	abcde	abcde	OK
%CALC{"\$LEFTSTRING(ásčšěžů, 5)"}%	ásčšě	ás	ERROR
%CALC{"\$LOWER(TEXT)"}%	text	text	OK
%CALC{"\$LOWER(ÁĚŠŤ)"}%	áěšť	ÁĚŠŤ	ERROR
%CALC{"\$PROPER(a small STEP)"}%	A Small Step	A Small Step	OK
%CALC{"\$PROPER(á šňál' ŠŤĚŘ)"}%	Á Šňál' Šťěř	á šňál' ŠŤĚŘ	ERROR
%CALC{"\$REPLACE(acdeilnorsz, 6, 5, \$comma)"}%	acdei,z	acdei,z	OK
%CALC{"\$REPLACE(áčďéíľňóřšž, 6, 5, \$comma)"}%	áčďéí,ž	áč,ľňóřšž	ERROR
%CALC{"\$TRANSLATE(boom,bm,cl)"}%	cool	cool	OK
%CALC{"\$TRANSLATE(cumi,ci,čí)"}%	čumí	čum	ERROR
%CALC{"\$TRANSLATE(čumí,čí,ci)"}%	cumi	ciumii	ERROR

Non critical (but annoying) errors

The table sorting doesn't sorts by the accented characters. For this will be need to use **utf8 + locale** based sorting, what is correctly available only from perl v5.20 (using an special pragma usable from v.5.16).

Try: sort by the **Dolořěm** column.

Próviděňť ôffičiã ěiňiľ perspějčiatís quiderěm đučimús

Nisi	Aľiqúid	Dolořěm	Fáčěřę	Omqjs	Ipsum
1	C	Ňuiã ěat'ús coňšěqúatúř	John	1-Dec-02	
2	X	Řelease	John	1-Apr-02	
3		Get feedback	Anne	1-Feb-02	
12	C	Špec error handling	Jack	1-Dec-02	
5		Ábc	John		
Nisi	Aľiqúid	Dolořěm	Fáčěřę	Omqjs	Ipsum

The %SPACEOUT{"SomeUtfText"}% won't add spaces even if the Uppercases are ASCII.

%SPACEOUT{"UšđámAššŭmėndáCøňšěčťět'úrDøľóř" separator=", "%}UšđámAššŭmėndáCøňšěčťět'úrDøľóř

Strange Errors

EDITTABLE

The following **NOT utf8** table using the %EDITTABLE% (copy from the [EditRowPlugin](#)), screws the normal Utf8 topic text. (also somewhat changes the width of the previous table). Try **<nop>** the EDITTABLE macro and the topic will be OK.

Nr	Project	State	Progress	Timestamp
	Sliced yoghourt	completed	☺	26 Jun 2002

Demo utf8 topic text

Mixed ASCII and Utf8

Quđđđem a d'eleçtus áliás. Aliđđđš ľiběřó áčcúsamús in éxerćit'atiõnem dist'ñçtio. Quòš susćipit hic fugit áut țempořa vòľup'atěš. Nihđđđl eligendí řěm facilis fùgã. Rãřioně bjãnditiis laborum quís řeřum. Fãciliš ũllãm sũnt a. Eř dólóřę ea ítaque a nõň cõřrupti prõviděňť đucimús. Sinđđđ qúø accusãntium árchitěčto mağňi quĩ ěisí. Fugiat quae offičiis đicťa vòľuptãtem.

Simľĩqyē ēť äűť ąśśuměňďã

Sęqui dólóřem fűgít moll'ĩtã òmnis. Sũňť quãś řepudjandae õđđđcãęcatĩ mollitia jpšam. Hic fuga iurē earum quò architěčtò. Aũtem nãtús bęãtãe ęř eaquę. Ręřum ęđđđ řepřehęderĩť áľiquãm ęšť bęãtãe eã debitis omnĩs. Aňimi vjřaę vòľup'atěm offičiis. Adipiscđ dólóřęmqűe ét vòľuptatũm. Prđđđęšęntium ęiუს řãcere et eš cõrrũpti atquę. Nõstrum est ĩmpedĩť ařchitęçřđ đřgnissimós pęřfereňdis. Súscđđđpit omnis ęxcepturi äđ ěiňĩľ offičia omnis ěãm accúsãntĩum.

Siđđđ vòľup'tãřum šũćcipit řerũm eňim ét quia vòľuptãtem. Istđđđ eòš veřo țempóre quia nihil' ařchitęçřđ đřľořěmqűe. Quđđđ asperňãřũr dist'ñçtio nõn. Tõtãm eũm řapięňťě nęquę mólęštiaę ĩđ ét eos pęřfereňdis. Suđđđt qui undę sũnt vòľuptãš áspęrnãřũr nequę ęľigęndj quĩ. Et đřgníššimos cũľpa pãřjãřũř ũť numquam. Harđđđm coňšěqúatũř vòľup'tãš

accusamus ratione beatæ. Animi ex consecratus aliquis in ratione. Et sed iusto consequatur et et recusandæ.

Qui laboriosam dolum

Dolor enim delictus illo animi ipsam aut. **Sapienter** enim possimus ratione laboriosam hic pariatum et. Rerum itaque excepturi nempe voluptas voluptas autem ratione nascuntur. Mordaci quod velint saepe repellendus et. Culpam praesentium vero ab odio. Vero corporis dolor ducimus laborum odio pariatum quia. Reiciendus et pariatum omnis. **Consequatur** quatur impedit consequatur quasi. Consequatur distinctio nemo iste.

Voluptas illo recusandæ et dicitur non quae qui parvo. Ipsum non iste voluptatum. Unde quo ut cumque perferendis dignissimos et. Voluptatem nihil dolor dolor est veritatis. Unde hic eum vitae ut omnis dolo. Dolorem perferendis minus ratione explicabo. Tempora nobis tempore pariatum et. **Voluptatem** tenetur omnis est aut eligendi. **Quod** velit ea mollitia quisquam molestiae debitis velit nam.

Words containing NO ASCII

iaecamenā IÆĈAMĒŃĂ IæĈamenā Iæcamenā nblgawæ NBLĠĀŴĂĔ Nblgawæ Nblgawæ sōñlōtķu SŌŃĹŌŦĶU Sōñlōtķu Sōñlōtķu rūuñzú RŪŪŦŪŦŪŦŪ RŭŪñzú RŭŪñzú Rŭuñzú ħšëððĥġ GŞĔÐĤĠĠŪ ĢšĔððĥġ Ģšëððĥġ áćŋúais ÁĆŊŪĀŚĂ ÁćŋúaisÁ ćŋúaisŕ flşøøŕŔ FLŞŌÈŦŔ flşøøŕŔ flşøøŕć étiştióć ÉŦŦŞĬŌĆ eŦŦştióć étiştiŋ eęzşigçŃ ĘĚŚİĠÇŊ ĘęzşigçŃ éyāpācœĹ ÉŸĀPĀĈŒĹ éYāpācœĹ éyāpācœz ũāriyđjŹ ŪŪŸDĴ ŪāriyđjŹ ũāriyđjŹ pđððŖŦPĐ ĐĐŦĐĐĐ Đđððđ Đđððđ

List for the SEARCH test

- ┆┆┆Lístóň
- ┆┆┆abc1xxdef
- ┆┆┆áčž2xxøđŕ
- ┆┆┆ábc3xxdeš
- ┆┆┆qwe4xxuio

JSchecker

The topic uses some JQUERY script to compare the wanted and actual values. Here is the source:

```
// for the each table row try to find a text :FWTEST:
// if found, takes two previous cells and compares them
// and replace the :FWTEST: with the result of the compare.
$(document).ready(function() {
var toterr = 0;
$('tbody tr').each(function () {
    var $cells = $(this).children(),
        $cmp = $cells.filter(':contains(":FWTEST:")'),
        cmpIndex = $cells.index($cmp);
    if( cmpIndex >= 2 ) {
        // array of values of previous cells
        var values = $.map($cells.slice(cmpIndex - 2, cmpIndex), function (el) {
            //return $.trim($(el).text()); //trimmed text
            //return $(el).html(); //not trimmed html
            return $.trim($(el).html()); //in most cases want compare with html too, but
with trimmed spaces
        });
        // make sure we have 2 cells with values and compare
        if( values.length === 2 ) {
            if( values[0] === values[1] ) {
                $cmp.text('OK');
            }
            else {
                $cmp.text('ERROR').css('color','red');
                toterr++;
            }
        }
    }
});
$('#totalerror').text(toterr);
});
```

Comments

Topic revision: r1 - 04 May 2015, UnknownUser