
Subject: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 10:29:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello !

I'm playing with witz and getting funny cryptic error messages that do not help much to find the problem:

Internal server error

Invalid value conversion: int -> N3Upp7WStringE

Subject: Re: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 11:03:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

We need somehow a stack trace something like this <https://github.com/Nanolat/c-callstack>

Subject: Re: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 11:19:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

The problem happen because I have a "\$post_identity()" instruction on a form but I'm accessing this page through GET somehow it dies here:

```
String GetIdentity(const Renderer *r)
{
    // This ugly hack expects that __identity__ is always present in r->var
    Http *http = const_cast<Http *>(dynamic_cast<const Http *>(r));
    if(!http)
        throw Exc("invalid POST identity call");
    String s = http->var[0]; ///<<<<<<<<< it dies here because I'm accessing through get / it's
intended to do this way mixed GET/POST
    if(s.GetCount())
        return s;
    s = AsString(Uuid::Create());
    http->SessionSet0("__identity__", s);
    http->var[0] = s;
    return s;
}
```

Subject: Re: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 11:24:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Changing the function to this make it works:

```
String GetIdentity(const Renderer *r)
{
    // This ugly hack expects that __identity__ is always present in r->var
    Http *http = const_cast<Http *>(dynamic_cast<const Http *>(r));
    if(!http)
        throw Exc("invalid POST identity call");
    Value val = http->var[0];
    String s;
    if(IsString(val))
    {
        s = http->var[0];
        if(s.GetCount())
            return s;
    }
    s = AsString(Uuid::Create());
    http->SessionSet0("__identity__", s);
    http->var[0] = s;
    return s;
}
```

Subject: Re: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 11:40:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Isn't worth to use something like Lua for witz, it already has all error handling well done and has more capabilities out of box.

I'm not saying to move business logic to templates, but having flexibility to try new things/variations without need to recompile is an interesting feature.

Cheers !

Subject: Re: Witz template cryptic error messages
Posted by [mingodad](#) on Tue, 15 Apr 2014 11:58:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

I did a microbenchmark with ESC, lua, luajit, SquiLu:

```
fact(x)
{
  if (x <= 1) {
    return 1;
  }
  return x * fact (x-1);
}

main() {
  i=0;
  x = 0;
  while (i < 100000) {
    x = fact (12);
    i+= 1;
  }
  Print(to_string(x) + "\n");
}
```

```
$ time ./EscApp
479001600
```

```
real 0m4.314s
user 0m4.304s
sys 0m0.000s
```

```
$ time squilu fact2.nut
479001600
```

```
real 0m0.425s
user 0m0.232s
sys 0m0.008s
```

```
$ time luajit fact2.lua
479001600
```

```
real 0m0.079s
user 0m0.016s
sys 0m0.000s
```

```
$ time lua-5.1.5 fact2.lua
479001600
```

```
real 0m0.243s
user 0m0.160s
sys 0m0.000s
```

Subject: Re: Witz template cryptic error messages
Posted by [mirek](#) on Wed, 16 Apr 2014 05:59:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

mingodad wrote on Tue, 15 April 2014 11:03We need somehow a stack trace something like this
<https://github.com/Nanolat/c-callstack>

Usually, I just run it in debugger.

Stack trace can be activated, the problem is that it requires everything being compiled to .so.

Subject: Re: Witz template cryptic error messages
Posted by [mirek](#) on Wed, 16 Apr 2014 06:05:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

mingodad wrote on Tue, 15 April 2014 11:40Isn't worth to use something like Lua for witz, it already has all error handling well done and has more capabilities out of box.

I'm not saying to move business logic to templates, but having flexibility to try new things/variations without need to recompile is an interesting feature.

Cheers !

I guess that is quite tough design decision in all template systems.

Before creating witz I was, of course, investigating other templating systems, then defined witz in a way that about permits as much as them.

I would say that if templates are common practice with PHP framework, then perhaps putting more flexibility into templates is a bad idea (because pure PHP can be considered a templating system with ultimate flexibility).

Subject: Re: Witz template cryptic error messages
Posted by [mirek](#) on Wed, 16 Apr 2014 06:29:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

mingodad wrote on Tue, 15 April 2014 11:19The problem happen because I have a "\$post_identity()" instruction on a form but I'm accessing this page through GET somehow it dies here:

While I agree there might be a problem in Skylark here, I believe it is more complicated than this.

In any case, just looking at the code, post_identity should work just fine with GET (of course, it is sort of weird to use it for GET, but whatever...).

Then I have tried adding \$post_identity() to Skylark05 example and it seems to work just fine.

One possibility is that you are using some function that breaks the witz's stack frame.

Perhaps you could post the testcase?

Mirek

Subject: Re: Witz template cryptic error messages
Posted by [mirek](#) on Wed, 16 Apr 2014 06:37:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

mingodad wrote on Tue, 15 April 2014 11:58I did a microbenchmark with ESC, lua, luajit, SquiLu:

There is no surprise there, Esc is just a very simple pure scripting language intended (originally) for .usc files, which are widget property/visuals definitions for Layout Designer.

Note however that Esc is completely unrelated to .witz. Witz is compiled to object representation and runs without interpreting anything. I have benchmarked WITZ and found it about 8 times slower than optimized C++. In real use, witz spends 80-90% of time just concatenating texts, therefore I doubt it would ever be performance bottleneck.

Mirek
