

Wt - Bug #4657

Returning 404 code in Wt app

01/11/2016 02:28 PM - Alan Finley

Status:	Feedback	Start date:	01/11/2016
Priority:	Normal	Due date:	
Assignee:	Koen Deforche	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	3.3.1		

Description

I would like to return 404 http code for pages that do not exist in my Wt app.

If I use `WApplication::setInternalPathValid`, I get 200 response code anyway:

```
Wt::WApplication* createApplication(const Wt::WEnvironment& env)
{
    Wt::WApplication *app = new Wt::WApplication(env);
    app->setInternalPathValid(false);
    return app;
}
```

I don't want to redirect a user to some static html page with some error text in it. The incorrect url should remain the same, like `http://www.google.com/dfsdfjk`

History

#1 - 01/11/2016 05:16 PM - Koen Deforche

- Status changed from New to Feedback
- Assignee set to Koen Deforche

You'll only see the 404 if you're using progressive bootstrap or having a browser without JavaScript (or a search-bot agent). Perhaps that explains?

#2 - 01/11/2016 05:51 PM - Alan Finley

It seems that in case of progressive bootstrap I can only use `setInternalPathValid` in the application constructor.

Is it possible to set 404 dynamically? When a user navigates to some internal path which is not 'supported' within an active Wt session, I would like to send 404 code:

```
MyApp::MyApp(const Wt::WEnvironment &env) :
    Wt::WApplication(env)
{
    internalPathChanged().connect(this, &MyApp::onInternalPathChanged);
}

void MyApp::onInternalPathChanged(std::string path)
{
    root()->clear();

    if (path == "/one")
    {
        root()->addWidget(new WText("page 1"));
    }
    else if (path == "/two")
    {
        root()->addWidget(new WText("page 2"));
    }
    else if (path == "/three")
    {
        root()->addWidget(new WText("page 3"));
    }
    else
    {
        root()->addWidget(new WText("ERROR: Page not found"));
        // here I would like to send code 404
    }
}
```

```
}  
}
```

#3 - 01/11/2016 06:06 PM - Koen Deforche

Hey Alan,

True. The system we use to change paths is to not actually load a new HTML page, but we handle this as an event within the page, not so different from a mouse event. We do have communication with the server, but in the form of an Ajax request (or WebSocket message). Ignoring websockets here, returning a 404 on this request would not be visible to the user, nor would it be correct since it's not a request for the missing page, it's a request for information from the server on how to update the current page.

Now what you could do (which would be correct) if you really want the 404 HTML page, is to redirect the user (`WApplication::redirect()`) in case the internal path is invalid, which would spawn a new session and generate the 404 server-side. That's even something Wt could do, as a configuration option. This also has as main draw-back that the user loses his session (going back would be a new session).

#4 - 01/11/2016 06:25 PM - Alan Finley

Koen Deforche wrote:

Now what you could do (which would be correct) if you really want the 404 HTML page, is to redirect the user (`WApplication::redirect()`) in case the internal path is invalid, which would spawn a new session and generate the 404 server-side. That's even something Wt could do, as a configuration option. This also has as main draw-back that the user loses his session (going back would be a new session).

It does not really matter for me if a user loses his session on redirect. All I want is to keep the initial url in browser. If a user types `http://myapp/some-bad-url`, his input should remain the same, but he should get the 404.

Is it possible to generate the 404 server-side using Wt after a redirect?

#5 - 01/11/2016 06:30 PM - Koen Deforche

Yes, that falls under the conditions listed above.