

Hi,

I started to use persistent sessions in my app, with default parameters (just changed cookie name and expiration time, so it should go to filesystem), and the app crashed inside `Http::Dispatch()` with invalid memory access:

```
if(SQL.IsOpen())
    SQL.Begin();
LoadSession();
session_dirty = false;
if(post && !bd.post_raw) {
    String id = Nvl((*this)[ "__post_identity__"], (*this)[ "__js_identity__"]);
    if(id != (*this)[ "__identity__"])
        throw AuthExc("identity error");
}
lang = Nvl(Int("__lang__"), LNG_ENGLISH);
Upp::SetLanguage(lang);
var.GetAdd("__lang__") = lang;
var.GetAdd(".language") = ToLower(LNGAsText(lang));
handlerid = bd.id;
LDUMP(handlerid);
bd.handler(*this);
if(session_dirty)
    SaveSession();
----CRASH HERE----> if(SQL.IsOpen())
    SQL.Commit();
```

Commenting out this (and the former one, to be sure...) the app runs well. It seems that it tries to access some uninitialized data inside SQL.

The weird stuff is that tracing it, on FIRST call (the one at start of this code fragment) it creates a static, empty Sql variable, NOT going to threading code; on second one (the crashing one) it goes through threading part.

I compiled in MT, but i'm running in debug mode, so it should be single threaded... I guess.