
Subject: Skylark modules?

Posted by [mirek](#) on Fri, 07 Sep 2012 10:45:18 GMT

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

Our nice little web framework looks now fine and good, but I am starting to feel that there is something missing: the ability to create "modules" as a group of "parameterizable" Skylark handlers to e.g. handle common CRUD scenarios.

I am tinkering with this issue for some time now, so far the best what I could have come with is something along (not actual Skylark code, just demonstration how the code could look, if we decide to go this way):

```
struct Crud {
    Sqlld table;
    Sqlld id;
    String home;
    String dialog;

    void SubmitNew(Http& http);
    void New(Http& http);
    void SubmitEdit(Http& http);
    void Edit(Http& http);
    void Delete(Http& http);

    typedef Crud CLASSNAME;

    void Skylark();
};

void Crud::SubmitNew(Http& http)
{
    SQL * http.Insert(table);
    http.Redirect(home);
}

void Crud::New(Http& http)
{
    http("ACTION", call(".SubmitNew"))
    .RenderResult(dialog);
}

void Crud::SubmitEdit(Http& http)
{
    SQL * http.Update(table).Where(key == http.Int(0));
    http.Redirect(homd);
}
```

```

void Crud::Edit(Http& http)
{
    int id = http.Int(0);
    http
        (Select(SqlAll()).From(table).Where(key == id))
        ("ID", id)
        ("ACTION", call(".SubmitEdit"), id)
        .RenderResult(dialog);
}

void Crud::Delete(Http& http)
{
    SQL * Delete(table).Where(key == atoi(http[0]));
    http.Redirect(home);
}

void Crud::Skylark()
{
    SKYLARK_METHOD(SubmitNew, "submit_create:POST");
    SKYLARK_METHOD(New, "create");
    SKYLARK_METHOD(SubmitEdit, "edit/*:POST");
    SKYLARK_METHOD(Edit, "edit/*");
    SKYLARK_METHOD>Delete, "person/delete/*")
}

// this is client code:
SKYLARK_USE(BasicCrud, Person, "person")
{
    Person.table = PERSON;
    Person.key = ID;
    Person.home = "HomePage";
    Person.dialog = "AddressBookWeb/Dialog";
}

```

Does this look good? Any better ideas?

Not 100% user I can really express this in C++, but it should be possible.

One apparent disadvantage is that we cannot use method names directly anymore, but have to use string literals. Found now way around so far...

Mirek