
Subject: Web framework - Skylark - first taste

Posted by [mirek](#) on Sun, 18 Dec 2011 19:26:00 GMT

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

I have got to the point where I can produce very simple application using (normal) forms, sqlite3 and witz templates (in the end, I am using "witz" both as the name of template language and as extension).

The point of application is to somewhat resemble examples/AddressBook. I have added pictures to test multipart posts...

Ugly boilerplate code excluded, application consists of following code and templates:

C++ code:

```
URL_VIEW(Picture, "person/*/picture")
{
    http.Content("image/jpeg", LoadFile(ConfigFile(http[0] + ".data")));
}

URL_VIEW(HomePage, "index.html")
{
    ValueArray person;
    Sql sql;
    sql * Select(ID, FIRSTNAME, LASTNAME, EMAIL, SEX)
        .From(PERSON)
        .OrderBy(LASTNAME);
    while(sql.Fetch()) {
        ValueMap vm;
        for(int i = 0; i < sql.GetColumns(); i++)
            vm.Add(ToUpper(sql.GetColumnInfo(i).name), sql[i]);
        vm.Add("HASPICTURE", FileExists(ConfigFile(AsString(sql[ID]) + ".data")));
        person.Add(vm);
    }
    http
        ("PERSON", person)
        .Render("AdrBook/Index.witz");
}

URL_VIEW(NotFound, "**")
{
    http.Redirect("/index.html");
}

URL_VIEW(SubmitNew, "submit")
{
    SQL * Insert(PERSON)
        (FIRSTNAME, http["firstname"])
```

```

    (LASTNAME, http["lastname"])
    (EMAIL, http["email"])
    (SEX, http["sex"])
;
SaveFile(ConfigFile(AsString(SQL.GetInsertedId()) + ".data"), http["file"]);
http.Redirect("/index.html");
}

```

```

URL_VIEW(New, "new")
{
    http("EDIT", 0).Render("AdrBook/Dialog.witz");
}

```

```

URL_VIEW(SubmitEdit, "submit/edit/*")
{
    int id = atoi(http[0]);
    SQL * Update(PERSON)
        (FIRSTNAME, http["firstname"])
        (LASTNAME, http["lastname"])
        (EMAIL, http["email"])
        (SEX, http["sex"])
        .Where(ID == atoi(http[0]));
;
    String file = http["file"];
    if(file.GetCount())
        SaveFile(ConfigFile(AsString(id) + ".data"), file);
    http.Redirect("/index.html");
}

```

```

URL_VIEW(Edit, "person/*/edit")
{
    int id = atoi(http[0]);
    Sql sql;
    sql * Select(FIRSTNAME, LASTNAME, EMAIL, SEX)
        .From(PERSON)
        .Where(ID == id);
    if(!sql.Fetch()) {
        http.Redirect("/index.html");
        return;
    }
    http
        ("ID", id)
        ("FIRSTNAME", sql[FIRSTNAME])
        ("LASTNAME", sql[LASTNAME])
        ("EMAIL", sql[EMAIL])
        ("SEX", sql[SEX])
        ("EDIT", 1)
        .Render("AdrBook/Dialog.witz");
}

```

```

}

URL_VIEW(Delete, "person/*/delete")
{
    SQL * Delete(PERSON).Where(ID == atoi(http[0]));
    DeleteFile(ConfigFile(atoi(http[0]) + ".data"));
    http.Redirect("/index.html");
}

```

And witz templates:

Style.witz

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>#TITLE</title>
</html>
<body>
    #BODY
</body>
</html>

```

```
#define TITLE This is an experimental Web AddressBook application
```

Dialog.witz

```
#include Style.witz
```

```

#define BODY
<FORM action=$(EDIT ? SubmitEdit(ID) : SubmitNew) method="post" accept-charset="utf-8"
enctype="multipart/form-data">
    <P>
        First name: <INPUT type="text" name="firstname" value="$FIRSTNAME"><BR>
        Last name: <INPUT type="text" name="lastname" value="$LASTNAME"><BR>
        email: <INPUT type="text" name="email" value="$EMAIL"><BR>
        <INPUT type="radio" name="sex" value="M" $(SEX != "F" ? "checked" : "")>Male<BR>
        <INPUT type="radio" name="sex" value="F" $(SEX == "F" ? "checked" : "")>Female<BR>
        <INPUT type="submit" value="Send" name="OK"><BR>
        Picture upload: <INPUT type="file" name="file" id="file" />
    </P>

```

</FORM>

#define TITLE Please enter a person!

Index.witz

#include Style.witz

#define BODY

<table border="1">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

<th>Sex</th>

<th>Action</th>

</tr>

\$for(i in PERSON)

<tr>

<td>\${i.FIRSTNAME}</td>

<td>\${i.LASTNAME}</td>

<td>\${i.EMAIL}</td>

<td>\${i.SEX == "M" ? "Male" : "Female"}</td>

<td>Edit Delete

 \${i.HASPICTURE} Show picture \$endif

</td>

</tr>

\$endfor

</table>

<p/>

Insert new person

C++ part contains lots of SQL<->http translations which can be easily automated. That excluded, I guess we are heading in the right direction...

Note that this is just hard bottom of framework. Nice things are yet to come

Mirek
