

revIgniter
FRAMEWORK

Ignite Your Sites with revlgniter

by Ralf Bitter on November 11, 2016 1 comment

[f](#) Share on Facebook[t](#) Share on Twitter

[Editors note] This month we are celebrating the relaunch of our hosting service as [LiveCode Hosting](#). With the spotlight on LiveCode Server I asked Ralf Bitter if he would like to write a bit about his excellent LC Server Framework, revlgniter.

Why revlgniter?

It was 2009 when the On-Rev web hosting service was introduced, long before the LiveCode Server engine was available to be installed on arbitrary web servers. Back then early adopters like me were thrilled and eager to learn about the new possibilities enabled by “a new variant of the existing technology that allows you to mix Revolution code with HTML content on the same page much in the same way as PHP”¹. There was a vibrant community dedicated to server-side scripting at the forums which were rapidly flooded with lots of code samples all mixing LiveCode (formerly Revolution) code with HTML or even additionally with CSS and JavaScript.

And this leads to the first possible answer to the headline question – “Why revlgniter”?

Why did I create revlgniter?

At that time I had a lot of fun taking the first steps in writing LiveCode code for the web, but I had a strong feeling that mixing markup and code might not be the best approach to write reasonably sophisticated programs for a web server. I had a feeling that there was missing a best practice solution to avoid redundancy, to provide a logical structure to access reusable code and to separate logic and presentation, especially for comprehensive multi-page projects.

The introduction of the include and require commands added in June, 2009 which allowed script files other than the main script to be parsed and executed, in combination with the merge function, opened the door and was the key to separating presentation from LiveCode scripting.

I started to think about building something like a framework to achieve the goals I had in mind but I knew I would not need to reinvent the wheel regarding a viable concept. As LiveCode server-side scripting has been modeled very strongly after PHP I just took a look over the PHP fence where I found a lean yet very impressive and absolutely compelling framework called CodeIgniter by Ellislab, Inc. I chose this as the pattern for a LiveCode web application framework to make it even more fun to write LiveCode code for a web server.

Obstacles and challenges

At first glance one might think translating PHP to LiveCode should be pretty straight forward. This may be the case for isolated functions but in my case I had to have a fairly thorough understanding of the overall structure of the PHP framework before I could even think about writing the first line of LiveCode code, primarily because of the different programming principles of PHP being an object-oriented programming language adhering to classes, objects, inheritance etc. Furthermore I was faced with PHP runtime configuration functions like error_reporting, display_errors, error_log etc., for which there are no equivalent counterparts in LiveCode. The same applies to built-in functions in PHP like for example addslashes, preg_replace_callback, wordwrap, trim etc. This means that I had to be prepared to write lots of code from scratch without having any PHP prototype. That said, despite all the hurdles, at least for the first release I tried to adopt the whole concept as closely as possible to the one of CodeIgniter, so that those familiar with CodeIgniter would feel pretty fast at home using revlgniter.

Now to meaning number two of the question.

Why use revlgniter?

One could ask: why not use CodeIgniter or any other framework built using PHP? The answer should be clear: LiveCode is a multi-platform developing tool. It is that simple. As I am using LiveCode anyway to deploy programs to other platforms using code **which is reusable for the web too**, why should I dive into PHP or any other server-side scripting language if I don't have to? But what's the deal with revlgniter?

revlgniter helps to keep a project structured and focused. As mentioned above it helps to separate your scripts from presentation to avoid unnecessarily convoluted, cluttered code and hence to keep code maintainable. It speeds up development and minimizes the amount of code you are required to write by providing libraries for frequently needed tasks. To illustrate what all this means let's adopt a scenario where there is a database query to dynamically populate a HTML table:

The code in “myModel.lc”:

```
# THE DATABASE QUERY
function dataFromModel pTable
    return rigDBget(pTable)
end dataFromModel

# THE HTML TABLE TEMPLATE
function myTableTemplate
    put "<table border=" & rigQ("0") && " cellpadding=" &
    rigQ("4") && " cellspacing=" & rigQ("0") && " class=" &
    rigQ("mytable") & ">" into tMyTemplate["tableOpen"]
    put "<tr class=" & rigQ("myRow") & ">" into
    tMyTemplate["rowStart"]
    put "<tr class=" & rigQ("myAltRow") & ">" into
    tMyTemplate["rowAltStart"]

    return tMyTemplate
end myTableTemplate
```

The code in controller “myController.lc”:

```
command index
    # GET URI SEGMENT
    put rigFetchRsegment(2) into tDBtable

    # GET DATA FROM DATABASE
    put dataFromModel(tDBtable) into tQuery
    put tQuery["resultarray"] into tResult

    # # BUILD TABLE
    # HEADING
    rigSetTableHeading tQuery["fieldnames"], TRUE

    # FETCH TABLE TEMPLATE FROM MODEL
    rigSetTableTemplate myTableTemplate()

    # TABLE GENERATION
    put rigGenerateTable(tResult) into gData["myDynamicData"]
    #

    # LOAD MARKUP
    get rigLoadView("myView")
end index
```

Now to display the data simply add gData[“myDynamicData”] to markup in another file (the view “myView.lc”) like:

```
• • •
<div><p>[ gData["myDynamicData"] ]</p></div>
• • •
```

Who and why – case studies

Alex Tweedly builds web sites for friends and local community / charity organizations. He developed websites using a slowly evolving set of “template” scripts of varying complexity, before he found revlgniter. Since then he has used revlgniter for all the sites (except for a brief foray into WordPress) like for example for [kilmelford.com](#). The reasons he uses revlgniter are (citing Alex):

Mostly because there is just so much fully functioning, fully debugged library/framework/helper code to handle all the hard work.

Because it's so much better documented than my various home-grown schemes.

It comes with great example auth.lc on which I can build authentication / member login, etc. (which I would otherwise struggle with, or get it wrong and leave vulnerabilities).

Dave Kilroy has developed various sites using revlgniter including a pricing and invoicing webapp and a revlgniter ‘backend’ website [members.plymsocent.org.uk](#) associated with a wordpress website [plymsocent.org.uk](#). Citing Dave:

A wordpress site is great but it doesn't allow you to do so much – unless you are prepared to a) spend a long time learning their framework and b) possess php skills (or spend more time learning php).

He uses the revlgniter website to let members edit their profiles and also to give committee members various tools to manage the Network. The site includes a “code bridge” between WordPress and a .lc file that updates the database the revlgniter site connects to each time a transaction is made on the main WordPress site.

Martin Blackman uses revlgniter for some engineering calculations. See [xtrados.com/app/index.lc/PGML](#).

Martin Koob used revlgniter to create the backend API for a cloud based Mac OS app developed with LiveCode which streamlines the process of recording student performance on video and then marking that performance using annotations at time points on the video. The entire backend consists of the API, revlgniter, LiveCode Server and a WordPress part which serves as a means to handle user authentication, promotion, signing up, billing, managing the user's account and providing some documentation.

[Editors Note] We love revlgniter here at LiveCode. Why not go and [check it out for yourself](#), and maybe install it on your [LiveCode Hosting](#) website?

1. Excerpt from the On-Rev engine notes 2009-06-02 by Mark Waddingham. [↗](#)

🔗 hosting , [LiveCode Server](#) 🗨️ 1 comment

[< Previous](#)[Next >](#)

Related Posts

Take a look at these posts

Keeping an eye on your server

The Hacked Server Hangover

Livecode.org email addresses and talking to PHP from LiveCode Server

Profile

[t](#) Sign in with Twitter or [f](#) Sign in with Facebook

Comment

Name

Email

Not published

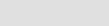
Website

Post It

1 Reply

[💬](#) 1 Comment[t](#) 0 Tweets[f](#) 0 Facebook[🔗](#) 0 Pingbacks

last reply was november 18, 2016

[Richard Gaskin](#)November 18, 2016 [💬](#)

revlgniter is among the very finest examples of excellent stewardship of open source projects in our LiveCode community. Ralf has been extremely responsive to feature requests and bug reports, and his documentation is very thorough and uncommonly well written.

[reply](#)

Recent Posts

My Extended LC Family

LiveCode 9.6 Released

LiveCode + Covid-19

Making the Tiger Roar!

Fence Painting or Coding Challenges?

Recent Comments

Elanor Buchanan on LiveCode Widgets: The Line Chart

Oldrich on LiveCode Widgets: The Line Chart

Heather Laine on My Extended LC Family

Michael Roberts on My Extended LC Family

Christian on My Extended LC Family

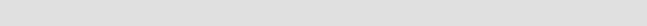
Tags

[Advanced](#) [android](#) [android app](#) [app](#) [Business](#) [Cloud](#) [Community](#) [Davos](#) [documentation](#) [Edinburgh](#) [Education](#) [Events](#) [featured](#) [Health](#) [heather](#) [High DPI](#) [hosting](#) [how to](#) [How To Create an App](#) [HTML5](#) [in-app purchasing](#) [infinite LiveCode](#) [iOS](#) [Java](#) [LiveCode 6.6](#) [LiveCode 6.7](#) [LiveCode 7](#) [LiveCode 8](#) [LiveCode 9](#) [LiveCode Builder](#) [LiveCode Server](#) [mergeEXT](#) [Mobile Apps](#) [Networking Layer](#) [On-Rev](#) [Quality](#) [supergeek](#) [Support](#) [tech](#) [Testing](#) [Training](#) [tsNet](#) [Unicode](#) [Widgets](#) [World Economic Forum](#)

Looking for the LiveCode open source project?

[🔗](#) Open Source

Looking for LiveCode FileMaker?

[LiveCode for FM](#)[Sitemap](#)[Terms](#) [Privacy Policy](#) [EULA](#)

Brought to you by LiveCode Ltd, Registered in Scotland, No. SC200728

[Offline \(Leave a message\)](#)