FrameSquared

Carbone, John
john@qlcorp.net

January 5, 2019

Abstract

FrameSquared performs database schema version control and scaffold generation for Postgres databases and the Treefrog C++ MVC framework using YAML to

1 Overview

The rise of iterative and incremental software development methodologies has been transformative. Continuous integration[1], deployment and delivery[2, 3] has met the challenge of regular changes in requirements. Unpredictable and changing requirements[4] have been factored into the development process to enable software architectures to evolve gracefully.

Iterative and incremental methodologies have been successfully applied to database schema revision control and migration[5]. Refactoring database schema effects the database and related queries but also has repercussions to external systems that access the database programmatically, these include models and controllers.

The nature of MVC frameworks lends itself to the use of boilerplate code. Some MVC frameworks create scaffolding using templates to generate boilerplate code. Scaffolding was made populate by Ruby-on-Rails. Most scaffold generators focus primarily on create, retrieve, update and delete (CRUD) operations using HTML[6, 7] and do not generally support single page applications[8] and associated messaging protocols such as SOAP or REST.

FrameSquared is tool that can generate code to be used as final source files and/or migrations based on serial numbers and up/down direction and can execute programs. FrameSquared uses the notion of "serial" numbers and "direction" to determine whether a migration "action" should be executed.

2 Getting Started

2.1 Required Libraries

FrameSquared utilizes the Inja[9] template engine. Inja and FrameSquared both rely on the nlohmann JSON library[10]. FrameSquared also uses the jbeder YAML[11] library.

```
Put your code here.
```

---

xxxxx

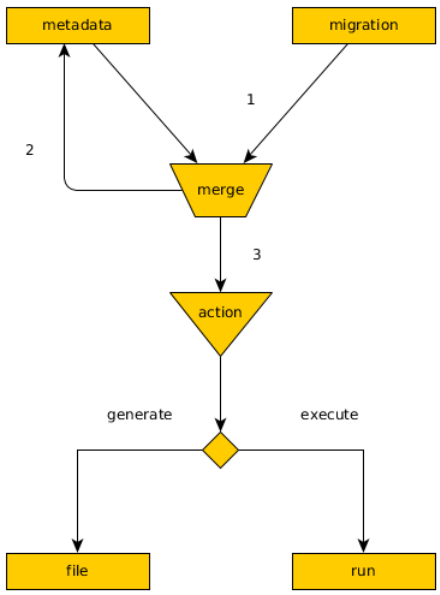

Figure 1: Merge Overview

References

[1] Paul M. Duvall, Steve Matyas, Paul Duvall, Andrew Glover *Continuous Integration: Improving Software Quality and Reducing Risk* Addison-Wesley, 2007

[2] Jez Humble and David Farley *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* Addison-Wesley, August 2010

[3] Sten Pitet *Continuous integration vs. continuous delivery vs. continuous deployment* `https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment` Atlassian, Retrieved January 1st, 2019

[4] Neal Ford, Rebecca Parsons, Patrick Kua *Building Evolutionary Architectures: Support Constant Change* O'Reilly Media Inc, October 8, 2017

[5] Scott J Ambler and Pramod J. Sadalage *Refactoring Databases: Evolutionary Database Design* Addison-Wesley Professional, March 13, 2006

[6] *Scaffoling (programming)* `https://en.wikipedia.org/wiki/Scaffold_(programming)` Wikipedia, Retrieved January 1st, 2019

[7] Nick Harrison *Using Scaffolding to Create MVC Applications with Visual Studio* `https://www.red-gate.com/simple-talk/dotnet/asp-net/using-scaffolding-to-create-mvc-applications-with-visual-studio/` www.red-gate.com, 14 December 2015

[8] Paul Sherman *How Single-Page Applications Work:Getting to know the browser behaviors and APIs responsible for powering more and more web pages* `https://medium.com/@pshrmn/demystifying-single-page-applications-3068d0555d46` mediun.com, 11 April 2018

[9] Inja *A Template Engine for Modern C++* `https://github.com/pantor/inja` GitHub

[10] JSON for Modern C++ *A Template Engine for Modern C++* `https://github.com/nlohmann/json/` GitHub

[11] YAML *A YAML parser and emitter in C++* `https://github.com/jbeder/yaml-cpp` GitHub