

# **pocket-knife**

A PHP, RESTful, web service framework

Andrew Brown  
October 22nd, 2013

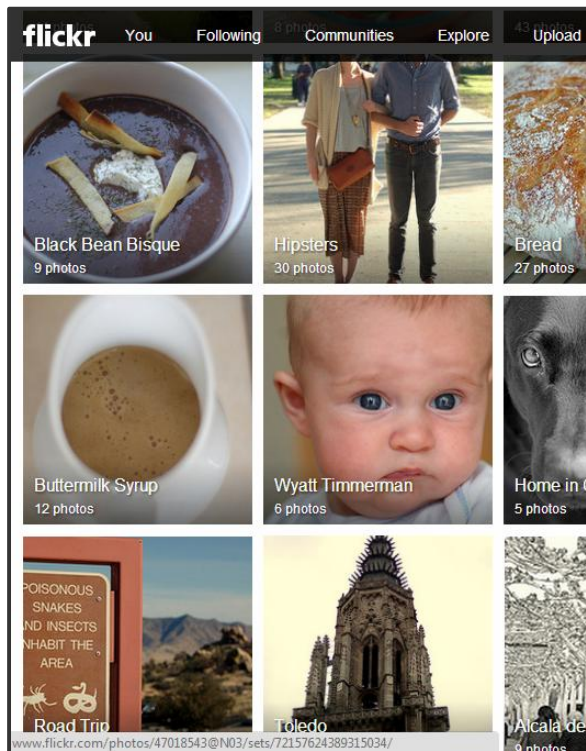
# Introduction

- In developing web-based applications, a common problem is how to save and later present data for users.
- Most users will have little to no experience with SQL, so raw database access is not recommended (or secure)
- Large-scale solutions like a Content Management System (CMS) are not flexible enough
- The problem is common enough that custom solutions are too work-intensive
- The solution must be web-based and use web technologies; ideally, a RESTful solution

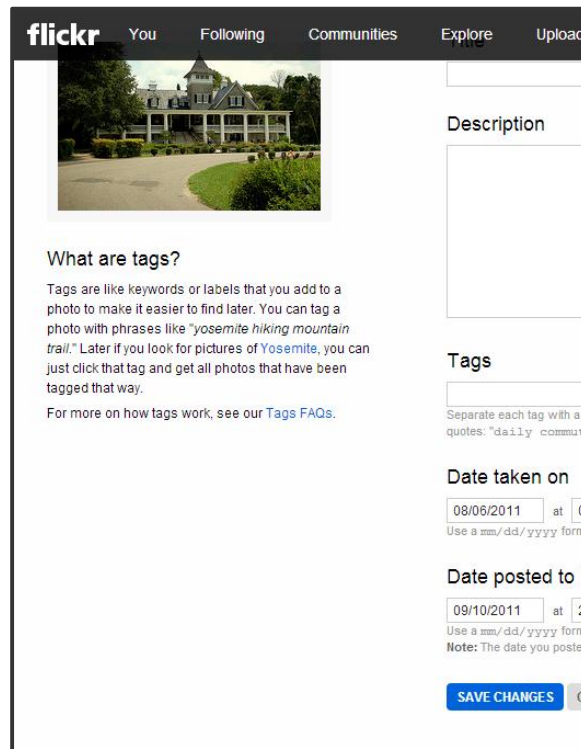
# A Common Problem

For each resource type, users need to:

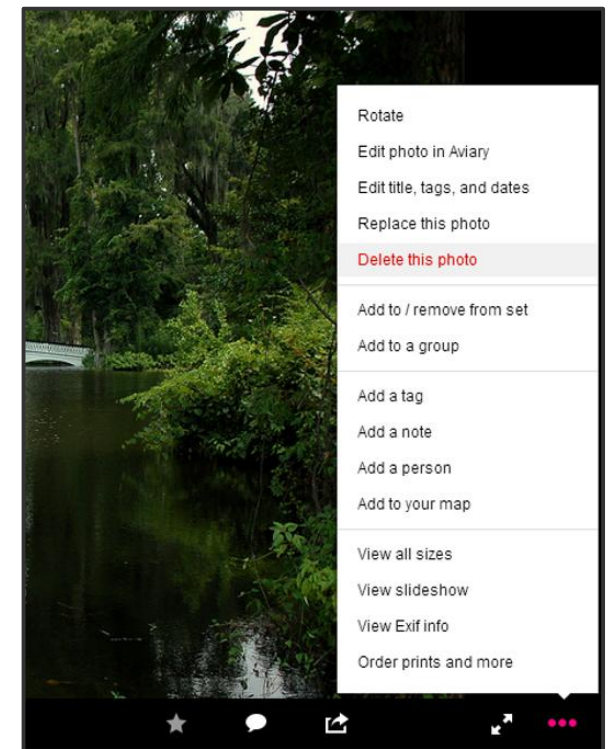
## List and View



## Create and Edit



## Delete



Users want the capabilities with a functional UI; developers do not want to build custom solutions for each domain

# pocket-knife

- Solution: **pocket-knife**, a PHP5 web service framework for building RESTful web services
- Features include:
  - **Extensible** implementation of Resource, Representation, Storage classes; this means it is simple for a developer to link to new storage types, for example
  - Automatically **matches the Content-Type** of the HTTP request (e.g. text/html, application/json, etc.) and returns the resource data in the desired format
  - Comprehensive security mechanism with multiple authentication types and configurable, **finely-granular ACL**
  - Storage modules covering **common data stores**, to include: JSON, CSV, PHP files, MySQL/SQLServer/Oracle/SQLite (using PDO), MongoDB, CouchDB, Amazon S3 (in progress)
  - **Built-in caching** system that automatically caches idempotent HTTP requests (e.g. GET); can be turned off/on easily in each Resource type

# pocket-knife

- Over 8,800 lines of code, with an additional 6,200 lines of documentation; over 90 PHP classes
- First GitHub commits in January 2011
- Used successfully in two commercial projects and one academic project
- Tested code; over 30 test files covering all core features of the framework
- Released under GNU/GPL license
- Code available at <http://github.com/andrewsbrown/pocket-knife>

# What Does RESTful mean?

- Representational State Transfer (REST) is a concept developed in Roy Fielding's 2000 doctoral dissertation
- In the 1990s Roy Fielding worked on the Apache HTTP server and the HTTP 1.0 and 1.1 standards
- Systems that are RESTful (e.g. web systems) are **client-server**, **stateless**, **layered**, may be **cached**, have **uniform interfaces** (e.g. URLs), and offer **code-on-demand** → **pattern for the web**
- Data is represented as a **resource**
- Data transfer uses a limited number of common verbs (compared here to database CRUD methods:

HTTP Verbs	CRUD Method	Data Commonly Sent...
GET	Read	Only in HTTP response
PUT	Update	In HTTP request and response
POST	Create	In HTTP request and response
DELETE	Delete	Only in HTTP response

# What Are Web Service Frameworks?

- A **web service** is a way to connect computers across networks, now usually exposed as Application Programming Interfaces (APIs); examples include Flickr API, Google Maps API, Twitter API, etc.
- A **framework** provides classes and methods to solve a computing problem; more than a library, however, it also defines the solution's architecture (e.g. MVC)

General-Purpose Libraries	Frameworks	Domain-Specific Solutions
PHP Modules	CakePHP	Wordpress
PEAR	Ruby-on-Rails (non-PHP)	OpenCart
Java Servlet (non-PHP)	<b>pocket-knife</b>	OwnCloud



- Other RESTful PHP web service frameworks include: Zend Framework, DAVE, Epiphany, FRAPI, Recess, etc.

# Other Frameworks Compared

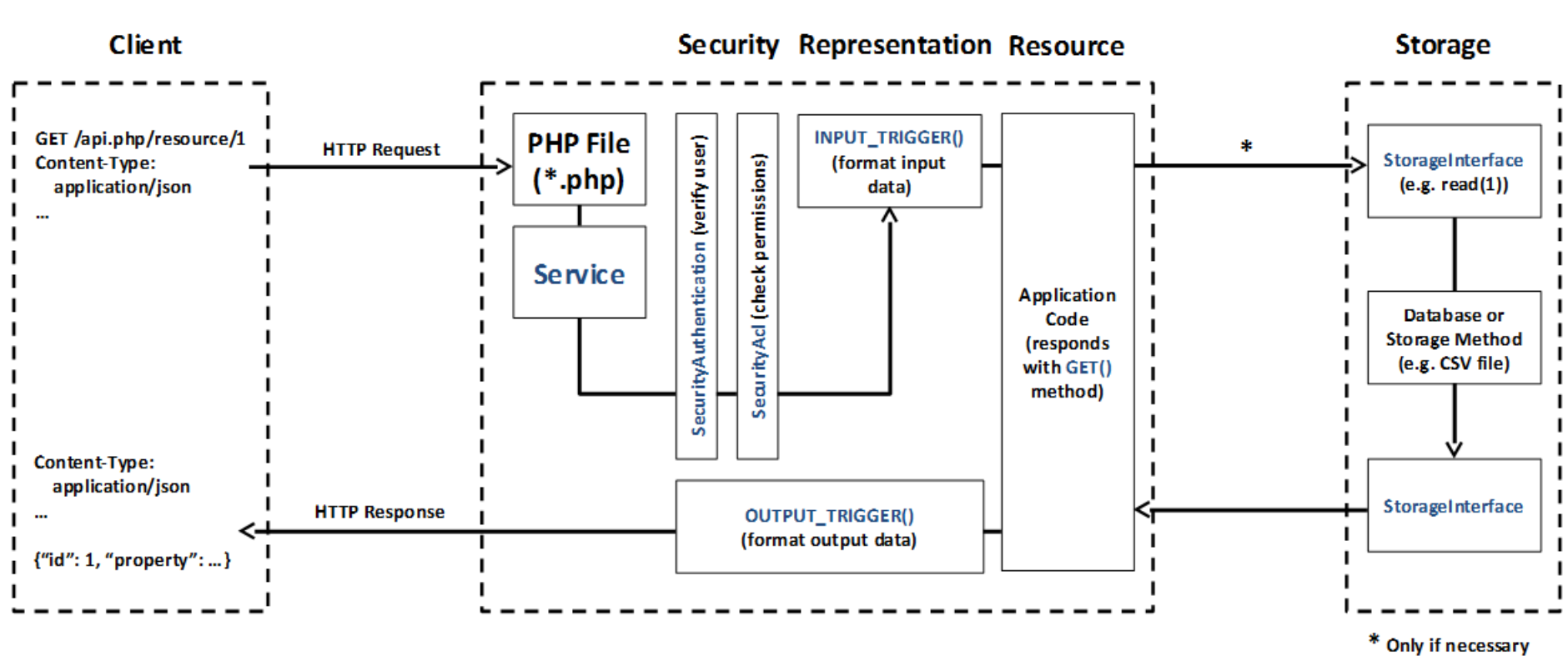
- The popularity of restful frameworks increased significantly in 2010, paralleling other Web 2.0 technologies; the following frameworks are compared against **pocket-knife** in several categories:

Framework	Type	REST	Auth	Caching	MIME Detect
pocket-knife	MVC	Yes	Yes	Yes	Yes
Zend	MVC	Yes	Yes	Yes	Yes
Yii	MVC	Yes?	Yes	Yes	No
DAVE	M + C	No?	No	Yes	Yes
Epiphany	C	Yes	No	Yes	No
FRAPI	V + C	Yes	No	Yes	Yes
Recess	MVC	Yes	No	No	No
Tonic	C	Yes	Yes	No	Yes



# The HTTP Request Cycle

- An HTTP request for `GET /api.php/resource/1` is modeled below:

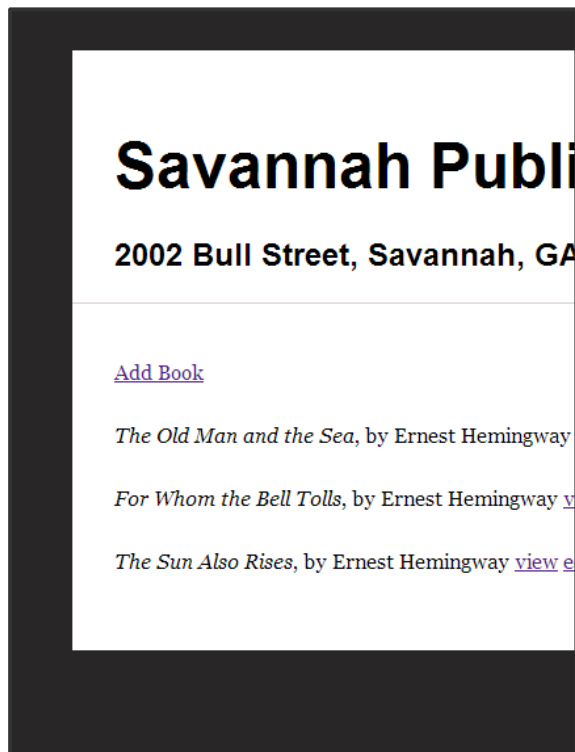


- Bottom Line: All aspects of the HTTP request cycle are covered by pocket-knife classes.

# Example Use Case: Library

- In 148 lines of PHP code (and 60 of CSS) we create a RESTful API representing a library catalog, accessible by browser and by JSON clients:

## List and View



**Savannah Public Library**  
2002 Bull Street, Savannah, GA 31401

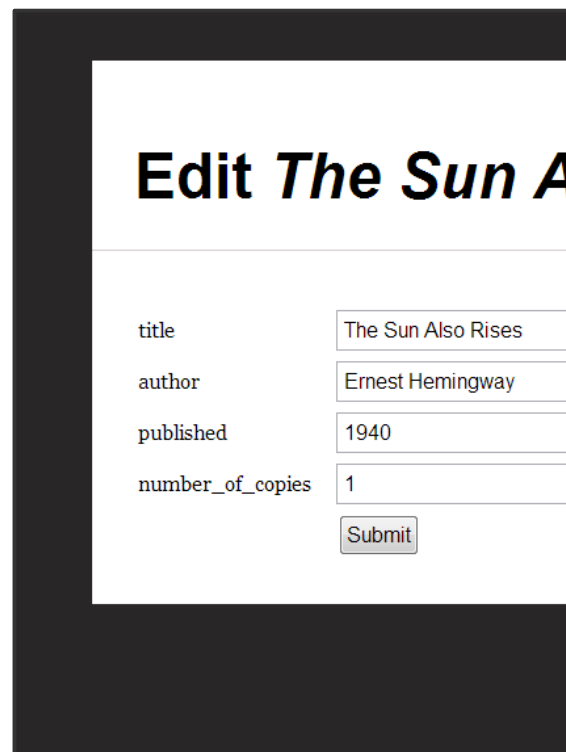
[Add Book](#)

*The Old Man and the Sea*, by Ernest Hemingway [view](#)

*For Whom the Bell Tolls*, by Ernest Hemingway [view](#)

*The Sun Also Rises*, by Ernest Hemingway [view](#)

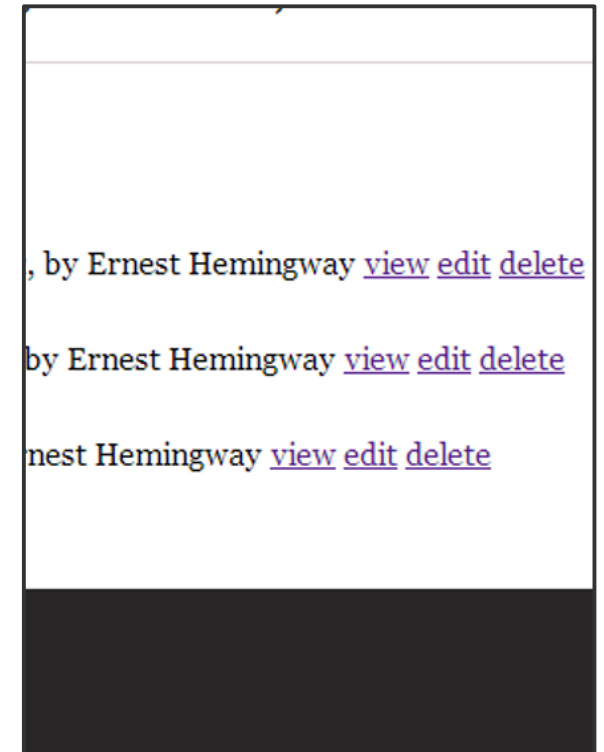
## Create and Edit



**Edit *The Sun Also Rises***

title	<input type="text" value="The Sun Also Rises"/>
author	<input type="text" value="Ernest Hemingway"/>
published	<input type="text" value="1940"/>
number_of_copies	<input type="text" value="1"/>

## Delete



*The Sun Also Rises*, by Ernest Hemingway [view](#) [edit](#) [delete](#)

*For Whom the Bell Tolls*, by Ernest Hemingway [view](#) [edit](#) [delete](#)

*The Old Man and the Sea*, by Ernest Hemingway [view](#) [edit](#) [delete](#)

- Test it out at: <http://casabrown.com/projects/pocket-knife/www/demo>
- From the browser, use GET arguments like 'method' and 'accept' to override HTTP headers

# Example Use Case: MentalSquares

- Used **pocket-knife** to implement the *MentalSquares* psychological evaluation model; the application stores mental evaluations by patient

## List and View (and Delete)

### Mental Squares

#### Patients

First Name	Last Name	Age	Address	
Joe	Schmoe	23	Atlanta, GA	<a href="#">Edit</a> <a href="#">Delete</a>
John	Doe	23	Savannah, GA	<a href="#">Edit</a> <a href="#">Delete</a>
Lisa	Simpson	13	Springfield, IL	<a href="#">Edit</a> <a href="#">Delete</a>

[New Patient](#)

#### Latest Tests

Patient	Modified On	
Lisa Simpson	7:53pm November 5, 2013	<a href="#">Edit</a> <a href="#">Delete</a>
John Doe	8:03pm November 5, 2013	<a href="#">Edit</a> <a href="#">Delete</a>

[New Test](#)

## Create and Edit

### Mental Squares

#### Edit This Test

Patient:

q1. Maintains personal hygiene and appearance	<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
q2. Is independent in transport and communication	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4
q3. Provides adequate nourishment for self	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4
q4. Attends to role performance tasks	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4
q5. Effective in role performance	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4
q6. Earns livelihood	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4
q7. Communicates with family	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4
q8. Participates in family activities	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4
q9. Maintains good family relations	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/>
q10. Has network of friends	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4
q11. Participates in social activities	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4

# Questions