Top 5 open source frameworks every application developer should know

By John Esposito

December 17, 2015 | 0 Comments | 6 min read

Register or Login to like

Software has been eating the world for far longer than four years. But developers think of software a little differently. Insofar as we're solving real-world problems, we're not thinking mainly of the metal. And as the problems get bigger and the solutions more complex, pragmatic (and not-too-leaky) abstractions become more important than ever.

The upshot: From a productivity-oriented developer's point of view, *frameworks* are eating the world. But which ones are eating how much of which parts of the world?

Given the insane variety of superb open source frameworks available, I

1. Presentation: Bootstrap

Let's start at the top of the stack: the presentation layer, the stuff developers and us folks both touch. Here the clear winner remains Bootstrap. The forecast looks outstanding, leaving old alternatives, such as Foundation, and new kids, such as Material Design Lite, in the dust. Bootstrap dominates usage trends on BuiltWith, and on GitHub remains easily the most starred and most forked framework of all time.

And Bootstrap is still under active development. In August, Bootstrap celebrated its fourth birthday with

2. Web MVC: AngularJS

As the web platform continues to mature, developers enjoy increasingly well-crafted abstraction-distance from the still-markup-colored DOM. The work begun by XMLHttpRequest reaches its zenith in modern Single-Page Applications (SPAs), and the most popular SPA framework by far is AngularJS.

What's so special about AngularJS? In a word: directives. Just a little `ng-` can bring a (static, markup) tag to (dynamic, JavaScript-executing) life. (Dependency injection is pretty neat, too, and like many of Angular's features aims to simplify maintenance and abstract more fully from the DOM.) The basic principle is just the

3. Enterprise Java: Spring Boot

What's so great about Java? Fast, mature, comprehensive class library, gigantic ecosystem, write-once-run-everywhere, active community—but *not* painless bootstrapping. Even hard-core Java developers resort to Ruby or Python to write quick one-off programs (*admit it*). And yet Java continues to dominate the enterprise for those other reasons listed above.

Enter Spring Boot, the boilerplate evaporator—a framework that lets you fit a working Spring application in a single tweet:

4. Data processing: Apache Spark

Once upon a time (in 2004), Google developed a programming model (MapReduce) that generalized many distributed batch processing job structures, then wrote a famous paper about it; then some Yahoo folks wrote a Java framework (Hadoop) that implemented MapReduce and a distributed file system to simplify data access for MapReduce tasks.

For nearly a decade Hadoop dominated the "Big Data" framework ecosystem, despite the limited problem space addressed (optimally) batch processing—partly because business and scientific users were accustomed to batch analysis of large

5. Delivery: Docker

Okay, so Docker isn't a "framework" in the sense of "code library, generously defined, that imposes a specific set of conventions to solve large and recurrent problem sets". But if frameworks are just things that let you write code at a more suitable level of abstraction, then Docker is a framework extraordinaire. (Let's call it an *exoskeletal framework*, just to mix metaphors confusingly.) And it would feel funny to name "top 2015 anythings for developers" without including Docker on the list.