

Architectural Styles of Extensible REST-based Applications

Justin R. Erenkrantz
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425
jerenkra@ics.uci.edu

ISR Technical Report # UCI-ISR-06-12

August 2006

Abstract:

At the beginning of the World Wide Web (WWW or Web), there was no clear set of principles to guide the decisions being made by developers and architects. In these early days, a cacophony emerged without a clear direction to guide the evolution of the Web. If there was any direction during the inception of the Web, it was a weak focus on how communication might occur between machines on the Web and the content that was to be transferred. Within a matter of a few years, scalability and other design concerns threatened the future of the early Web - this led to the introduction of REpresentation State Transfer architectural style (REST). The REST style imposed constraints on the exchange of communication over the Web and provided guidance for further modifications to the underlying protocols. The introduction of REST, through the HTTP/1.1 protocol, restored order to the Web by articulating the necessary constraints required for participation.

In this survey, we will characterize any environment that is governed by REST constraints to be in a RESTful world. Obviously, the largest example of the RESTful world is that of the Web with almost 75 million websites existing today and many more daily users. Yet, to this day, people are still struggling with how to write applications and architectures that adhere to the constraints of the REST architectural style. Consequently, it is all too common to see programs falling into a trap of ignoring and compromising the REST principles. These traps can jeopardize the beneficial induced properties dictated by the REST style - which could ultimately reintroduce the problems that REST was specifically imposed to address.

The existing Web infrastructure, and especially important components of that infrastructure like Apache, Mozilla, and others, can inform us about how to implement other RESTful components; indeed, examining the architectures of these tools and the infrastructure as a whole is key. With the rich history of the Web, we now have over ten years of real-world architectural evolution from which to base our examinations. Our aim in this survey is to classify the evolution, supported by real software architectures and frameworks, and to indicate insights and techniques useful for developing applications as a whole - that is, complete configurations of RESTful nodes that together form RESTful software applications without compromising the beneficial properties of REST.