Implementing SOA Patterns: Input/Output Validator

Lori MacVittie, 2007-03-04

The *Input/Output Validator* pattern takes advantage of decoupling to provide a single resource through which input to and output from services can be validated. In a nutshell, this pattern implements data integrity controls through which data is forced to conform to specific constraints, such as length, data type, and character encoding.

The pattern is generally implemented as a service, centralizing data integrity controls. This pattern is interesting because there are several ways in which it can be implemented using an application delivery controller.

For patterns involving pure XML, an application firewall with XML specific intelligence - generally schema validation - can be used to implement this pattern. The "service" is deployed in-line, and enforces the XML schema on input and output, ensuring data integrity. For patterns that involve either non-XML data, or a mixture, an application firewall with XML capabilities can also provide the intelligence to implement this pattern inline, in the network. This implementation carries the same benefits as implementing a separate software-based service - centralization, consistency, compliance - as well as also providing additional security measures that are not required by the *Input/Output Validator* pattern.

Anther way in which this pattern can be implemented is by using an application delivery controller as a service coordinator to enforce the flow of messages through the service that implements the *Input/Output Validator* pattern. Using iRules, incoming messages can be first routed to the *Input/Output Validator* service, and then, after validating that the response was positive, the message can be routed to the appropriate service. If the validation service indicates an issue with the message, the BIG-IP can respond to the client appropriately. Similarly, upon receiving a response ***that*** data can be routed to the *Input/Output Validator* service to ensure compliance with data integrity controls before returning either the original response or a custom reply containing the appropriate information to the client.

The third way in which this pattern can be implemented is as a service through iRules, taking advantage of BIG-IP's ability to inspect and extract data from both requests and responses. This would require more work than the first two options, but is also a feasible method of implementing the *Input/Output Validator* that not only acts as a service but could potentially be called inline, thereby reducing the overhead of a separate call to another service (and improving performance in the process). This implementation also offers two methods of integration into the process, as an inline invocation or a remote call, which provides the flexibilty required of services for a successful SOA implementation.

Imbibing: Coffee and Mountain Dew (Don't look at me like that, it's only Tuesday after all)

Technorati tags: F5, iRules, SOA, XML, patterns

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.	F5 Networks	F5 Networks Ltd.	F5 Networks
Corporate Headquarters	Asia-Pacific	Europe/Middle-East/Africa	Japan K.K.
info@f5.com	apacinfo@f5.com	emeainfo@f5.com	f5j-info@f5.com