ops4j

Ops4j is a framework which brings Java coding to the native OS without all of the hassle.

With ops4j, developers write to a specific contract defined in `Op`. Code written to this contract are known as operations and can be used as native OS tooling. This is great for automation, daily problem solving, etc...

Operations can be combined with other operations to solve greater problems. These solutions can be saved to `operation repositories` for common or personal use.

Ops4j puts Java to work on everyday problems without all of the ceremony required by more specialized environments such as J2EE or Spring based microservices.

Removed from it's ivory tower pedestal, this code is now used continuously as one would use a hammer. As the OSS adage goes, `Many eyes make all bugs shallow`. Defects of such common use utility code should be more quickly fixed.

Ops4j removes environmental barriers by transforming the native os shell (bash) into an agile environment for solving problems quickly. With ops4j the code is no longer buried under layers of process and environmental variables. With ops4j, we deal with our code (operations).

Ops4J is a Java framework which integrates Java code seamlessly into the native OS CLI. Ops4J code behaves identically to other shell utilities. These shell commands can be combined with other shell commands to solve even larger problems. This makes ops4j a nice tool for automation.

Ops4j solutions take the form of shell commands which can be stored and run anywhere.

concepts

CONCEPT	DESCRIPTION
operation	A single unit of code which operates upon each JSON document within the stream. For example, the operation `mongo:insert` will insert each JSON document in the stream into the designated mongo collection.
node operation	A single unit of code which operates at a single JSON node level. For example, the `encrypt` node operation will emit a text node containing the encrypted and base-64 encoded value of the input node.
operation Repository	A place to save configured operations by name. Example Repositories: filesystem, mongo, jdbc

operation lifecycle	A metadata object which describes the lifecycle of a given operation.
	INITIALIZE = intialize the operation.
	OPEN = open the operation
	RUN = process each json record in the stream.
	CLOSE = close the operation
	CLEANUP = perform any cleanup for the operation.

operation data	Also known as OpData, a single record of data upon which an operation performs it's task. Ultimately, OpData is a thin wrapper on top of Jackson JSON data.

Operations