

Sling in a few words

sling is a thin and RESTful web application framework on top of jcr

jcr is a tree structured store ("a hierarichal content repository")

this allows to map tree-structured URLs directly to the tree structure of JCR as if it were a file system with static files read by a simple fileserver

eg. download a file in jcr at /content/photos/holiday/01.jpg via <http://sling.localhost.com/photos/holiday/01.jpg> (notice that the root URL / is normally mapped to /content in the JCR, because you want to use the JCR for other stuff as well that is not directly accessible from outside)

normal GET, PUT (and other) HTTP methods are handled by sling (that's one point why it's restful, the other being the concept of the data in the JCR as the equivalent of a resource)

if you need to create dynamic webpages, you can write scripts (quick and cool) or plain old Java servlets (IDE support + managed code)

scripts can be in many popular languages (javascript, ruby, python, etc.) - sling as it is written in Java uses the Java Scripting framework (JSR-223)

these scripts (or servlets) are automatically selected depending on what is found in the jcr at the path given by the URL

this automatic selection has many possibilites to make the life of the developer as convenient as possible, aka there are lots of fallback mechanism - and no explicit config for the mapping of URL -> "handler" needed!

the most basic selection is done by setting a "sling:resourceType" as attribute on the node in jcr - eg. a "userprofile", which would be handled by the userprofile script

others... POST.js

handle.selector.extension ...
example for that ...

cool standard stuff (...0.json)

including other resources inside your script (components, like including jsps)

filters (again scripts or servlets) - both for request (before sling looks for the handler) and response (after the handler has run)