
Requirements for a URI matching function

Requirement 1 (introduced by Tim) “Match URL subtrees”

The requirement is to match nodes and leaves of a specified subtree in common URL schemes, such as “http”, “https” and “file”. Because the scheme part of a URL is case-insensitive and the authority part may contain a port range specification, reg-exp match is not a suitable solution. Instead, we need to leverage the IP-address- and DNS-name-match functions that have already been defined.

The initial solution was based on an algorithm defined by JSR115. It allows a single wild-card character either at the beginning or end of the path part.

For example, if the match string is:

`http://x.y/a/*` (where the “*” character represents the wild-card)

Then the following URLs must march:

`http://x.y/a`
`http://x.y/a/`
`http://x.y/a/c.txt`
`HTTP://www.x.y/a/b/c.html`

And the following URLs must not match

`http://x.y/a.txt`

If the match string is:

`http://x.y/*.html`

Then the following URLs must match

`http://x.y/c.html`
`HTTP://www.x.y/a/b/c.html`

The wild-card represents zero to many levels in the hierarchy.

Modification to Requirement 1 (suggested by Rich) “Allow a wild-card in the middle of the path part”

For example, if the match string is:

`http://x.y/a/*.html`

Then the following URLs must match

http://x.y/a/c.html
HTTP://www.x.y/a/b/c.html

Requirement 2 (introduced by Michiharu) “Match absolute path URIs”

The requirement is to match absolute path URIs (i.e. those that omit the scheme and authority parts, so they start with a single “/” character). The requirement is to match a) child nodes of the match string or b) descendant nodes of the match string.

For example, in the case of the child match function, if the match string is

/a/*

Then the following URIs must match

/a/b/
/a/c.x

And the following URIs must not match

/a
/a/
/a/b/c.x

In this case, the wild-card represents one level in the hierarchy.

In the case of the subtree match function, if the match string is

/a/*

Then the following URIs must match

/a/b/
/a/c.x
/a/b/c.x

And the following URIs must not match

/a

In this case, the wild-card represents one to many levels in the hierarchy.

Two approaches are possible: either separate functions should be defined for the child and descendant cases, using the same wild-card convention, or a single function should be defined for both cases, but separate wild-card conventions should be defined in each case.

It has been suggested that the reg-exp string match function is suitable for this purpose.