

Server Process /rsg/index#p

/rsg/index#p, where output is the location of the newly created resource with id. In other words, some transitions have return values beyond a HTTP status code.

The process /rsg/index#p is a singleton process, that is, has a state (as a single occurrence). id++ for each newly created occurrence. Infinite states (!) if we do not restrict the range of id.

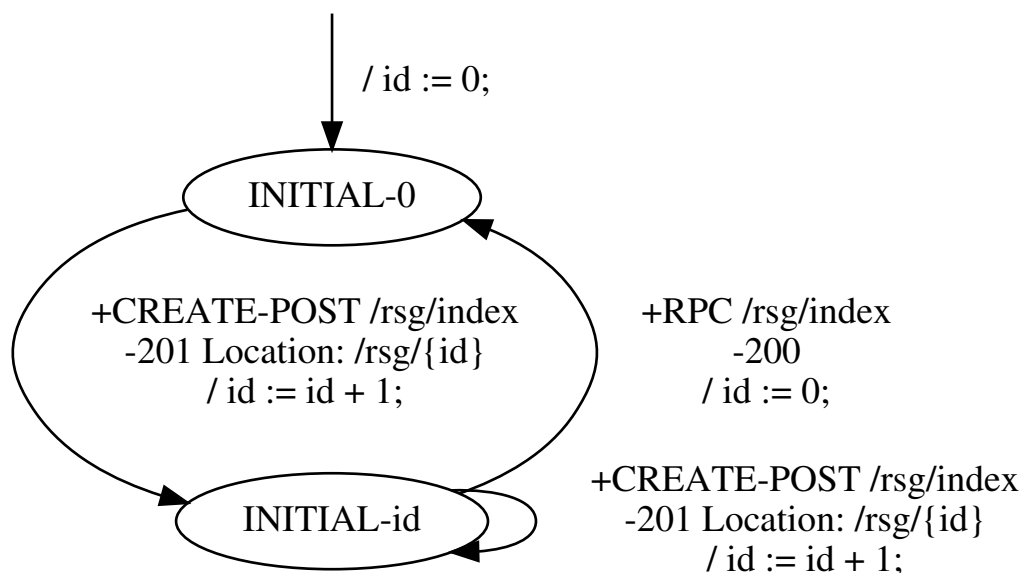
Process /rsg/index#p is related to /ready/index#p, /set/index#p and /go/index#p. Or rather: /ready/{id}#p, /set/{id}#p, /go/{id}#p?

CREATE-POST-rsg-index creates new occurrences of /rsg/{id}#p, including subs.

Is the server process /rsg/index#p analogous to CREATE-PUT server fragments (both are describing resource lifecycle)? But then transition from NOT-EXISTS-RSG-id to EXISTS-RSG-id.

The request RPC-rsg-index wipes out the existing occurrences /rsg/{id}#o (and subordinate resources /ready/{id}#o, /set/{id}#o, /go/{id}#o), even those with a 410 Gone tombstone.

- Input: -
- Output: id via URI template in location header



The process representation contains the name of the interaction, e.g., CREATE-POST-rsg-index-201.

Occurrence /rsg/index#o

The occurrence representation include the affordance (with name). The affordance contains concrete requests/response, with concrete target URI and location header URI.

t = 0

Form POST /rsg/index, 201 (named CREATE-POST-rsg-index-201)

t = 1

Form POST /rsg/index, 201 (named CREATE-POST-rsg-index-201)

Form POST /rsg/index, 200 (named RPC-rsg-index-200) - probably needs a tweak because of clash with POST on URI /rsg/index