Service Description

Main server resource: /NffgService/rest/resource

Resources	URLs	XML Repr	Meaning
1 database	/database	NffgInfoType	The whole database made of NFFGs and Policies
* nffgs	/nffgs	NffgSetType	The collection of NFFGs
1 {id}	/nffgs/{id}	NffgType	A single NFFG
* policies	/policies	PolicySetType	The collection of Policies
1 {id}	/policies/{id}	PolicyType	A single Policy
* verifications	/verifications	PolicySetType	The collection of Verified Policies
1 {id}	/verifications/{id}	PolicyType	A single Verified Policy

Resource	Method	Req. Body	Query Param.	Status	Resp. Body	Meaning / Description
database	GET			200	NffgInfoType	Ok Receive the whole database
				500	default error page	Server Error
nffgs	GET		all : Boolean id : List<IDType>	200	NffgSetType	OK Receive all set of NFFGs all=true -> get all nffgs List != null -> get specified NFFGs
				400	custom error page	Bad Request, invalid query parameters
				404	custom error page	Not Found, one or more id requested have not been found
				500	default error page	Server Error
	POST	NffgType		201	NffgType	Res. Created, Create a new Nffg
				400	custom error page	Bad Request invalid data
				403	custom error page	Forbidden, data violates DB Constraints
				503	custom error page	Neo4J Service Unavailable
				500	default error page	Server Error
nffgs/{id}	GET			200	NffgType	Ok Get specified NFFG
				404	Custom error page	Not Found
				500	Default error page	Server error
	DELETE		force : boolean	200	NffgType	Ok Delete specified NFFG and return it in the response, if referenced policies are present “force” parameter must be set to true to delete also related policies
				403	Custom error page	Forbidden, NFFG has related policies
				404	Custom error page	Not Found
				503	Custom error page	Neo4J Service Unavailable
				500	default error page	Server Error

policies	GET		all : Boolean id : List<IDType>	200	PolicySetType	Ok Get the whole set of policies both traversal and reachable all=true -> get all policies List<IDType> != null -> get specified Policies
				400	Custom error page	Bad Request, invalid query parameters
				404	Custom error page	Not Found, one or more id requested have not been found
				500	default error page	Server Error
	POST	PolicyType		201	PolicyType	Created, create a new policy
				400	Custom error page	Bad Request invalid data
				403	Custom error page	Forbidden, data violates DB Constraints
				500	default error page	Server Error
	PUT	PolicyType		200	PolicyType	Ok, update an existing policy
				400	Custom error page	Bad Request invalid data
				403	Custom error page	Forbidden, data violates DB Constraints Eg. Referenced NFFG not present in DB
				404	Custom error page	Not Found, cannot update
				500	default error page	Server Error
policies/{id}	GET			200	PolicyType	Ok, Get specified Policy
				404	Custom error page	Not Found
				500	Custom error page	Server error
	DELETE			200	PolicyType	Ok Delete a Policy and return it in the response
				404	Custom error page	Not Found
				500	default error page	Server Error
verifications	GET		all : Boolean id : List<IDType>	200	PolicyType	Ok, Execute the verification and return the list of Verified Policies all=true -> get all policies List != null -> get specified Policies
				400	Custom error page	Bad Request, invalid query parameters

				404	Custom error page	Not Found, one or more id requested have not been found
				503	custom error page	Neo4J Service Unavailable
				500	default error page	Server Error
	PUT	PolicySetType		200	PolicySetType	Ok Put a Set of Policies to be verified but not stored and return the verified set
				400	Custom error page	Bad Request invalid data
				403	Custom error page	Forbidden, policies do not reference NFFGs
				503	Custom error page	Neo4J Service Unavailable
				500	default error page	Server Error
verifications/{id}	GET			200	PolicyType	Ok execute the Verification and return the policy
				404	Custom error page	Not Found
				503	Custom error page	Neo4J Service Unavailable
				500	default error page	Server Error

Details

The service implemented has been designed to be REST Level 3 and perform HATEOAS. In GET responses are provided links for all the resources of the service, furthermore in case of client errors in building requests (invalid query parameters, bad request body etc..) custom html pages are built to provide suggestions. Types for request bodies of POST and PUT operations are all present in the WADL description to help clients build requests. To scale well with hundreds of policies and NFFGs, all resources have been stored in Maps, so time to retrieve an NFFG or a Policy will be constant regardless of the database size. Also the neo4j data used to build verification requests is kept locally in a Map, so also the time to build a verification request for a policy will be constant regardless of database size and even of the number of nodes in the involved NFFG.

Clients will be able to send or update (in case of a policy) a single NFFG or Policy per request. This choice has been taken on purpose because checking database constraints for long lists of elements is a time-consuming operation that will not be good in a multithread environment (a client posting a long list will block the service for several time). For the GET operations there is not this limitation and client can decide how much data

to retrieve: whole set of policies or NFFGs, single policies, single NFFGs, a specified list of policies or NFFGs, execute the verification of a single policy a list of policies or all the policies.

The possibility to retrieve the whole database with a single GET operation has been also implemented, because clients may need a consistent snapshot of data at a certain time that could not be achieved by retrieving NFFGs and Policies with separate calls. (other clients may delete data between calls).

Submitted data is always validated initially by a MessageBodyReader component that uses a schema. Further validation processes to ensure the internal database consistency are performed in java.