

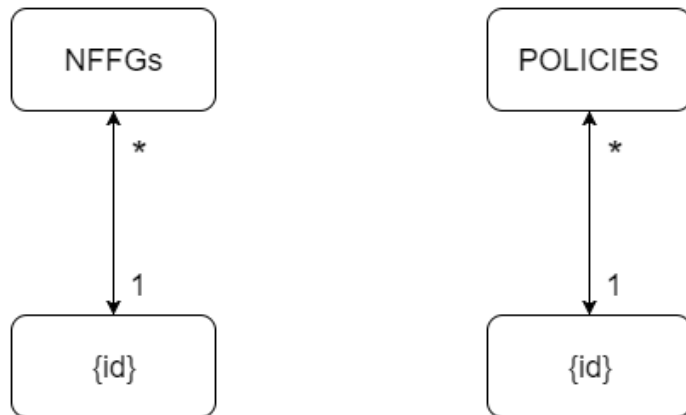
Distributed Programming II

A.Y. 2016/17

Assignment n. 3 part a – NffgService REST API Documentation

NffgService provides the possibility to load, delete and retrieve whole NFFGs. Moreover it is possible to load, delete, retrieve and verify reachability policies against the stored NFFGs.

• Resources Hierarchy



• Resources Mapping*

- | | |
|-----------------------------|---|
| ▪ baseURL/ | Documentantion page |
| ▪ baseURL/application.wadl | WADL description of NffgService |
| ▪ baseURL/nffgs | Collections of NFFGs |
| ▪ baseURL/nffgs/{id} | Single NFFG within the collection |
| ▪ baseURL/policies | Collections of Policies |
| ▪ baseURL/policies/{id} | Single policy within the collection |
| ▪ baseURL/policies/verifier | Verifier of the policy against the stored NFFGs |

*baseURL = base url of the service hosted by the server (e.g. <http://localhost:8080/NffgService/rest>)

- Operations on NFFGs

Resource URL*	Method	Query params	Request Body	Response Body	Status code	Meaning
/nffgs	GET	-	-	Content-type: application/xml Entity: entityPointers	200	Return the pointers to the nffgs loaded on the service.
/nffgs	GET	-	-	-	204	There are not nffgs loaded on the service.
/nffgs	POST	-	Content-type: application/xml Entity: nffgs	Content-type: application/xml Entity: entityPointers	201	Return the pointers to the nffg created.
/nffgs	POST	-	Content-type: application/xml Entity: nffgs	Content-type: text/plain Entity: error message	403	nffgs contained in the request body have not been created since there are already nffgs loaded on the service with the same names.
/nffgs	DELETE	-	-	-	204	All the nffgs and all the policies have been deleted.
/nffgs/{id}	GET	-	-	Content-type: application/xml Entity: nffgs	200	Return the nffg found.
/nffgs/{id}	GET	-	-	Content-type: text/plain Entity: error message	404	nffg not found.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	-	204	If <i>deletePolicies == true</i> the nffg and all eventual referring policies have been deleted; if <i>deletePolicies == false</i> the nffg has been deleted since there are no policies referring to it.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	Content-type: text/plain Entity: error message	403	nffg not deleted since <i>deletePolicies == false</i> and there is at least one policy referring to this nffg.
/nffgs/{id}	DELETE	deletePolicies: boolean	-	Content-type: text/plain Entity: error message	404	nffg to delete not found.

*Resource URL are relative to baseURL

- Operations on Policies

Resource URL*	Method	Query params	Request Body	Response Body	Status code	Meaning
<i>/policies</i>	GET	-	-	Content-type: application/xml Entity: entityPointers	200	Return the pointers to the policies loaded on the service.
<i>/policies</i>	GET	-	-	-	204	There are not policies loaded.
<i>/policies</i>	PUT	-	Content-type: application/xml Entity: policies	Content-type: application/xml Entity: entityPointers	201	Return the pointers to the policies created.
<i>/policies</i>	PUT	-	Content-type: application/xml Entity: policies	Content-type: text/plain Entity: error message	403	policies not created since the referenced nffgs or nodes are missing on the service.
<i>/policies</i>	DELETE	-	-	-	204	All the policies have been deleted.
<i>/policies/{id}</i>	GET	-	-	Content-type: application/xml Entity: policies	200	Return the policy found.
<i>/policies/{id}</i>	GET	-	-	Content-type: text/plain Entity: error message	404	policy not found.
<i>/policies/{id}</i>	DELETE	-	-	-	204	policy has been deleted.
<i>/policies/{id}</i>	DELETE	-	-	Content-type: text/plain Entity: error message	404	policy to delete not found.
<i>/policies/verifier</i>	PUT	-	Content-type: application/xml Entity: policies	Content-type: application/xml Entity: policies	200	The policies contained in the request body have been verified and sent back in the response along with the verification results.
<i>/policies/verifier</i>	PUT	-	Content-type: application/xml Entity: policies	Content-type: text/plain Entity: error message	403	The policies contained in the request body have not been verified since the referenced nffgs or nodes are missing on the service.
<i>/policies/verifier</i>	POST	-	Content-type: application/xml Entity: namedEntities	Content-type: application/xml Entity: policies	200	The policies stored on the service and referenced by the policy names contained in the request body have been verified and sent in the response along with the verification results.
<i>/policies/verifier</i>	POST	-	Content-type: application/xml Entity: namedEntities	Content-type: text/plain Entity: error message	403	The policies referenced by the policy names of the request body are missing on the service.

*Resource URL are relative to baseURL

Assignment n. 3 part b – NffgService REST Design and Implementation Documentation

- NffgService Design structure

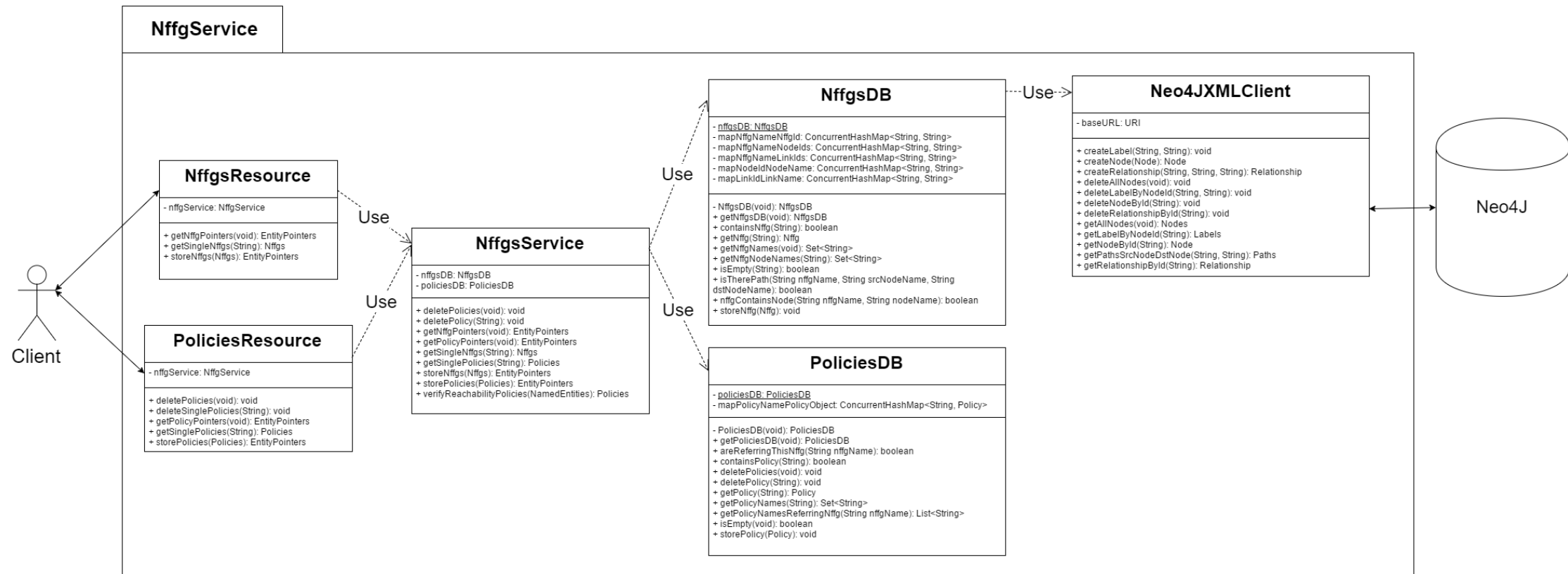


Figure 1: “simplified” class diagram of NffgService. It is intended only to give an idea of the developed solution.

- Design & Implementation choices

- To achieve scalability NffgService does not return the full collections of NFFGs. When a GET on “/nffgs” is performed, NffgService returns a list of the NFFGs names stored on the service and the relative “pointers” (URL where to perform the GET of a single nffg). The same approach has been used for the Policies
- All the information about NFFGs are stored on Neo4J (including networkFunctionalities and update time). NffgService is just a “cache” where to store simple and small data useful to retrieve nodes and relationships from Neo4J and map them with the data to send/requested to/from the client.
- When a client asks for the verification of the policy, NffgService returns the whole policies verified, each one of them containing the verificationResult element. It does not return a verificationResult alone in order to inform the client that the verificationResult has been computed exactly on the returned policy. This is MANDATORY in a concurrent NffgService to avoid a verification request on a certain version of policy and obtain a result based on a different policy. In this way the client can always check the real policy tested.