NffgService Interface document

Aim of this document is to explain and show the structure of the service and to allow the user to correctly interface with the server.

1. Resources

In the following we will use the tag <base_path> to reference the deploy location of the service, made of an IP address and a local port. In our case, we deployed the service at “localhost:9008”

The NffgService includes the following resources:

Resource	Description
<base_path>/NffgService/rest	Base URL of NffgService
<base_path>/NffgService/rest/application.wadl	WADL descriptor
<base_path>/NffgService/rest/nffgVerifier	The nffgVerifier resource, i.e. the main resource
<base_path>/NffgService/rest/nffgVerifier/nffgs	The set of nffgs inside the service
<base_path>/NffgService/rest/nffgVerifier/nffgs/{id}	A single nffg identified by its name
<base_path>/NffgService/rest/nffgVerifier/policies	The set of policies inside the service
<base_path>/NffgService/rest/nffgVerifier/policies/{id}	A single policy identified by its name

The resulting structure is directly a mapping of the xml structure resulting from the schema “nffgVerifier.xsd”. The aim at this point has been to keep a clean structure, closely related to the schema, following the REST specifications.

2. Operations

In the following we will discuss what the operations available in the service are. When possible, operations will make use of the same message as input or output, in order to avoid replication of objects. For example, when a list of names is needed to delete a set of policies or nffgs, the same object is used as input for both methods.

The input and output of each operations is an xml, in order to avoid “Unsupported media type” error, it is needed to specify “Content-Type: application/xml” in the header of each HTTP POST (and, if it’s needed, specify “Accept: application/xml” in the header of each HTTP GET).

Examples of input and output are in the folder “Examples”.

2.1. Get the nffgVerifier

This method let the user get the actual nffgVerifier in the service, with all the nffgs and all the policies.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier	GET	-	200	NffgVerifierType	OK
			500	-	SERVER ERROR

2.2. Get the set of nffgs

This method allows the user to get the set of nffgs

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/nffgs	GET	-	200	NffgsType	OK
			500	-	SERVER ERROR

2.3. Upload a set of nffgs

This method allows the user to upload the set of nffgs

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/nffgs	POST	NffgsType	200	ReferenceByIDType	OK
			403	-	FORBIDDEN
			500	-	SERVER ERROR

2.4. Delete a set of nffgs

This method allows the user to delete a set of nffgs. Thanks to the associated query parameter “removeAssociatedPolicies” (of type “xsd:boolean”) – in case the parameter is missing, the default value is true- it’s also possible to delete (with the parameter set to true) all the policies that were referencing the deleted nffgs.

URL	Method	Query Param	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/nffgs	DELETE	{removeAssociatedPolicies}	ReferenceByIDType	200	ReferenceByIDType	OK
				404	-	NOT FOUND
				500	-	SERVER ERROR

2.5. Get a single nffg

This method allows the user to get the single nffg, referenced by the path parameter.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/nffgs/{id}	GET	-	200	NffgType	OK
			404	-	NOT FOUND
			500	-	SERVER ERROR

2.6. Get the set of policies

This method allows the user to get the set of policies in the server. If the query parameter “nffgName” (of type “xsd:string”) is valorized with the name of one the nffg presents in the service, than this method will reply with only the policies associated to that nffg. At the default this value is null, and in this case the method will reply with the policies associated to every nffg.

URL	Method	Query Param	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService /rest/nffgVerifier/policies	GET	{nffgName}	-	200	PoliciesType	OK
				500	-	SERVER ERROR

2.7. Upload a set of policies

This method allows the user to upload a set of policies. Only policies referencing existing nffg can be uploaded, in case the nffg does not exist, the method will reply with a forbidden error.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/policies	POST	PoliciesType	200	ReferenceByIDType	OK
			403	-	FORBIDDEN
			500	-	SERVER ERROR

2.8. Delete a set of policies

This method allows the user to delete a set of policies.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/policies	DELETE	ReferenceByIDType	200	ReferenceByIDType	OK
			404	-	NOT FOUND

			500	-	SERVER ERROR

2.9. Get a single policy

This method allow the user to get a single policy.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/policies/{id}	GET	-	200	PolicyType	OK
			404	-	NOT FOUND
			500	-	SERVER ERROR

2.10. Get the set of policies and the referenced nffgs

This method allows the user to get the set of policies and the nffgs referenced by those policies. Moreover thanks to two query parameters, the set of needed policies can be further filtered:

- If nffgName (of type "xsd:string") is valorized (and not left at the default null), the method will filter the policies based on the referenced nffg.
- If verificationTime (of type "xsd:dateTime") is valorized (and not left at the default null), the method will filter the policies based on the verification time, considering as positive policies only those that have verification time after the time specified by the parameter.

URL	Method	Query Param	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/policiesAndReferencedNffgs	GET	{nffgName},{verificationTime}	-	200	PoliciesType	OK
				404	-	NOT FOUND
				500	-	SERVER ERROR

2.11. Verify a set of stored policies

This method allows the user to verify a set of stored policies. The verification results of these policies will also be saved in the service.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/verifyStoredPolicies	POST	ReferenceByIDType	200	VerificationResultsType	OK
			404	-	NOT FOUND
			500	-	SERVER ERROR

2.12. Verify a set of new policies

This method allows the user to verify a set of polices not existing in the service. The results will not be stored in the server.

URL	Method	Request Body	Response Status	Response Body	Meaning
<base_path>/NffgService/rest/nffgVerifier/verifyNewPolicies	POST	PoliciesType	200	VerificationResultsType	OK
			404	-	NOT FOUND
			500	-	SERVER ERROR