REST API for ECK Validation Module

Revision History

Revision	Date	Author	Organisation	Description
V0.1	2013-02-25	Katalin Szűcs	MON	Initial version for discussion
V0.2	2013-04-24	Ágoston Berger	MON	Minor changes in the API
V0.3	2013-07-25	Ágoston Berger	MON	API redesign for WP3 It2 – proposal for shared API
V0.4	2013-08-06	Ágoston Berger, István Nagy, Sašo Zagoranski	MON, SEM	Shared API for WP3 It2

Validation profile management

Responsible for profile upload and management. Validation profiles are stored in an XML format that specifies name and an ordered list of validation rules. Clients are able to:

- Retrieve a list of currently available profiles
- Retreive a profile by name
- Upload a new profile or update an existing one
- Delete a profile

URL (pattern)	**Request body**	**HTTP Method**	**Function**	**Response body**
/Validation/{provider}/profiles	none	GET	List available profiles	List of available profiles
/Validation/{provider}/profiles/{name}	none	GET	Return a profile XML by name	Profile body in XML format
/Validation/{provider}/profiles/{name}	profile in XML format	POST/PUT	Upload a profile, existing profile will be overwritten	empty
/Validation/{provider}/profiles/{name}	none	DELETE	Delete a profile by name	empty

Record validation

Responsible for validating EDM records. Clients are able to:

- Choose a validation profile, and validate a record against it immediately.
  - A record can be in any supported format, default is LIDO.
- Validate a record set.
- Retrieve a validation result for a given set.
- List active sets.

URL (pattern)	Request body	HTTP Method	Function	Response body
/Validation/{provider}/single/validate/{name}	Record in LIDO (or other) format	POST	Validate a single record	Validation result in XML format, HTTP 200 if validation successful, 412 if there were errors.
/Validation/{provider}/sets	empty	GET	List available sets, with processing status	Sets as XML or JSON as determined by mime type
/Validation/{provider}/{setnumber}/validate/{name}	Records in LIDO format as a ZIP file (application/zip)	POST	Validate the record using the selected profile, batch mode	HTTP 202, empty body, HTTP 409 if set already exists for current provider
/Validation/{provider}/{setnumber}/result	empty	GET	Retrieve batch validation results	Results in XML format or as a zip (determined by header mime type), HTTP 200 if results are available, HTTP 102 if processing is still ongoing.

Validation profile format

The validation profile (plugin configuration) is an XML file resembling Spring bean definitions.

- Each plugin used has their own namespace containing plugin and configuration declarations that can be used in the profile. This ensures the module can be extended by additional plugins without modifying the core code.
- Each plugin used is declared in the profile. The module executes validator plugins in the order they are declared here.

Example (default LIDO validation configuration for Monguz implementation):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd-
validator="http://www.europeanainside.eu/schema/xsd-validator" xmlns:uri-
validator="http://www.europeanainside.eu/schema/uri-validator" xmlns:link-
validator="http://www.europeanainside.eu/schema/link-validator"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.europeanainside.eu/schema/xsd-validator
http://www.europeanainside.eu/schema/xsd-validator/xsd-validator.xsd
http://www.europeanainside.eu/schema/uri-validator
http://www.europeanainside.eu/schema/uri-validator/uri-validator.xsd
http://www.europeanainside.eu/schema/link-validator
http://www.europeanainside.eu/schema/link-validator/link-validator.xsd">
    <bean class="eu.europeanainside.validation.ValidatorPluginManager">
        <property name="plugins">
            <list>
                <ref bean="schemaPlugin" />
                <ref bean="uriPlugin" />
                <ref bean="linkPlugin" />
            </list>
        </property>
    </bean>
    <xsd-validator:define name="schemaPlugin" xsdPath="http://www.lido-
schema.org/schema/v1.0/lido-v1.0.xsd" />
    <uri-validator:define name="uriPlugin" />
    <link-validator:define name="linkPlugin" />
</beans>
```

Set lifecycle

- There is a call to delete results that are not needed anymore. This will be called by a maintenance process cleaning result after a given period.
- Optional: the result retrieval call might delete the result after successful retrieval.