**SLA USER GUIDE**

Table of Contents

1 Introduction 4

2 SLA Repository 5

3 SLA Service 6

3.1 APIs Provided 6

3.1.1 Templates 6

3.1.1.1 Get all the templates 6

3.1.1.2 Get a certain template by its id identifier 7

3.1.1.3 Get a certain template by its uuid identifier 8

3.1.1.4 Get the template(s) of a certain provider 9

3.1.1.5 Create a template 10

3.1.1.6 Update a template 11

3.1.1.7 Delete a template 12

3.1.2 Agreements 12

3.1.2.1 Get all the agreements 12

3.1.2.2 Get an specific agreements by the agreement-id 14

3.1.2.3 Get active agreements 15

3.1.2.4 Create an agreement 15

3.1.2.5 Update a specific agreement 16

3.1.2.6 Delete an agreement by the agreement-id 18

3.1.3 Enforcements 18

3.1.3.1 Get all the enforcements 18

3.1.3.2 Get a certain enforcement by agreement-id 19

3.1.3.3 Create an enforcement 20

3.1.3.4 Start an enforcement 21

3.1.3.5 Stop an enforcement 22

3.1.3.6 Delete the enforcements of a specific provider 22

3.1.4 Providers 24

3.1.4.1 Get all the providers 24

3.1.4.2 Get a specific provider 24

3.1.4.3 Create a provider 25

3.1.4.4 Update a provider 26

3.1.4.5 Delete a provider by the provider-id 27

3.1.4.6 Delete a provider by his name 28

3.1.5 Violations 28

3.1.5.1 Get all the violations 28

3.1.5.2 Get a specific violation 29

3.1.5.3 Get a violation by agreement-id 30

3.1.5.4 Get all the violations by service name 31

3.1.5.5 Get violations given a range of dates 32

3.1.5.6 Get the violations of a specific provider 33

3.1.5.7 Create a violation 34

3.1.5.8 Update a violation 35

3.1.5.9 Delete a violation 36

3.1.6 Breaches 37

3.1.6.1 Get all the breaches 37

3.1.6.2 Get a specific violation 38

3.1.6.3 Get a breach by agreement-id 38

3.1.6.4 Get all the breaches by metric name 39

3.1.6.5 Get violations given a range of dates 40

3.1.6.6 Get the breaches of a specific range of dates 41

3.1.6.7 Create a breach 42

3.1.6.8 Update a breach 43

4 SLA Management 47

4.1 Implementation details 47

5 Portal 48

5.1 Portal Plugin 48

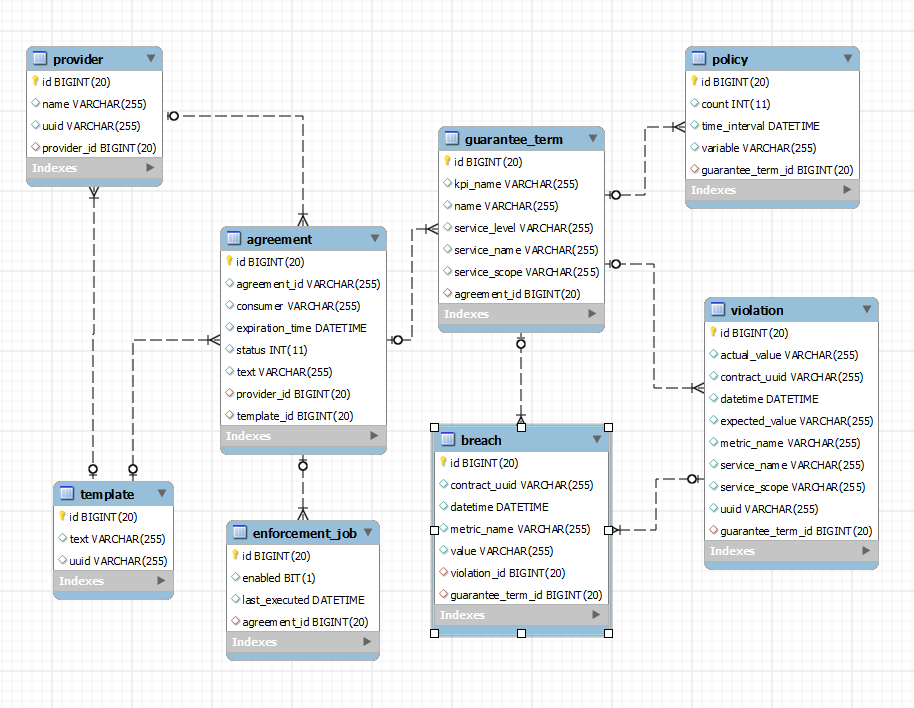
# Introduction

The SLA is made up of many software components. The documentation contains information about component architecture, responsability, APIs and relation with the other components.

The SLA architecture is composed of:

* SLA Repository
* SLA Management
* SLA Serializers
* SLA Service
* SLA Personalization

# SLA Repository



# SLA Service

This API is used by the SLA Management.

## APIs Provided

The SLA Service provides the following REST API:

### Templates

#### Get all the templates

Get the entire set of templates.

**Request**

Method: GET

Path: /templates

Curl command:

curl [–X GET] -u user:password -H ["Accept: application/json"| "Accept: application/xml"] localhost:8080/sla-service/templates

**Response**

Return code: 200 OK

Response format: application/xml or application/json depending on the accept header

Response: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<templates>

<wsag:Template xmlns:wsag="http://www.ggf.org/namespaces/ws-agreement" TemplateId="template01">

<wsag:Name>ExampleTemplate</wsag:Name>

<wsag:Context>

<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>

<wsag:ServiceProvider>provider01</wsag:ServiceProvider>

<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>

<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>

<wsag:TemplateId>template01</wsag:TemplateId>

<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>

</wsag:Context>

<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>

</wsag:Template >

<wsag:Template xmlns:wsag=<http://www.ggf.org/namespaces/ws-agreement> TemplateId="template04">

<wsag:Name>ExampleTemplate2</wsag:Name>

<wsag:Context>

<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>

<wsag:ServiceProvider>provider01</wsag:ServiceProvider>

<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>

<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>

<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>

</wsag:Context>

<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>

</wsag:Template >

</ templates>

Response: application/json

["name":"ExampleTemplate",

"templateId":"template01",

"context":{

"agreementInitiator":"provider02",

"agreementResponder":null,

"serviceProvider":"AgreementInitiator",

"templateId":"template01",

"service":"service3",

"expirationTime":"2014-03-07T12:00:00CET"

},

"name":"ExampleTemplate",

"terms":{

"allTerms":{…}

}

}, {

"templateId":"template02",

"context":{

"agreementInitiator":null,

"agreementResponder":"provider03",

"serviceProvider":"AgreementResponder",

"templateId":null,

"service":"service03",

"expirationTime":"2014-03-07T12:00:00CET"

},

"name":"ExampleTemplate2",

"terms":{…}

}

]

#### Get a certain template by its uuid identifier

For a given uuid identifier, get the specific template.

**Request**

Method: GET

Path: /templates/{template-uuid}

Curl command :

curl [–X GET] curl [–X GET] -u user:password -H ["Accept: application/json"| "Accept: application/xml"] localhost:8080/sla-service/templates/{template-uuid}

**Response**

Return code: 200 OK

Response format: application/xml or application/json depending on the accept header

Response: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<wsag:Template xmlns:wsag="http://www.ggf.org/namespaces/ws-agreement" TemplateId="template01">

<wsag:Name>ExampleTemplate</wsag:Name>

<wsag:Context>

<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>

<wsag:ServiceProvider>provider01</wsag:ServiceProvider>

<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>

<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>

<wsag:TemplateId>template01</wsag:TemplateId>

<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>

</wsag:Context>

<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>

</wsag:Template >

{

"templateId":"template02",

"context":{

"agreementInitiator":null,

"agreementResponder":"provider03",

"serviceProvider":"AgreementResponder",

"templateId":null,

"service":"service03",

"expirationTime":"2014-03-07T12:00:00CET"

},

"name":"ExampleTemplate2",

"terms":{ … }

}

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no template with id {uuid} in the SLA Repository Database</message>

</error>

Return code: 404 Not found

Response format: application/json

Response:

{"code":404,"message":"There is no template with id {uuid} in the SLA Repository Database"}

#### ~~Get the template(s) of a certain provider~~

~~For a given provider passing as parameter, get his templates associated.~~

**~~Request~~**

~~Method: GET~~

~~Path: /templates/~~

~~Query Parameters: provider-uuid~~

~~Curl command :~~

~~curl [–X GET] localhost:8080/sla-service/templates/?provider=”{provider-uuid}”~~

**~~Response~~**

~~Return code: 200~~

~~Response format: application/xml or application/json depending on the accept header~~

~~Response: application/xml~~

~~<?xml version="1.0" encoding="UTF-8"?>~~

~~<templates>~~

~~<wsag:Template xmlns:wsag="http://www.ggf.org/namespaces/ws-agreement" TemplateId="template01">~~

~~<wsag:Name>ExampleTemplate</wsag:Name>~~

~~<wsag:Context>~~

~~<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>~~

~~<wsag:ServiceProvider>provider01</wsag:ServiceProvider>~~

~~<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>~~

~~<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>~~

~~<wsag:TemplateId>template01</wsag:TemplateId>~~

~~<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>~~

~~</wsag:Context>~~

~~<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>~~

~~</wsag:Template >~~

~~<wsag:Template xmlns:wsag=~~[~~http://www.ggf.org/namespaces/ws-agreement~~](http://www.ggf.org/namespaces/ws-agreement) ~~TemplateId="template04">~~

~~<wsag:Name>ExampleTemplate2</wsag:Name>~~

~~<wsag:Context>~~

~~<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>~~

~~<wsag:ServiceProvider>provider01</wsag:ServiceProvider>~~

~~<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>~~

~~<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>~~

~~<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>~~

~~</wsag:Context>~~

~~<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>~~

~~</wsag:Template >~~

~~</ templates>~~

~~Response: application/json~~

~~["name":"ExampleTemplate",~~

~~"templateId":"template01",~~

~~"context":{~~

~~"agreementInitiator":"provider02",~~

~~"agreementResponder":null,~~

~~"serviceProvider":"AgreementInitiator",~~

~~"templateId":"template01",~~

~~"service":"service3",~~

~~"expirationTime":"2014-03-07T12:00:00CET"~~

~~},~~

~~"name":"ExampleTemplate",~~

~~"terms":{~~

~~"allTerms":{…}~~

~~}~~

~~}, {~~

~~"templateId":"template02",~~

~~"context":{~~

~~"agreementInitiator":null,~~

~~"agreementResponder":"provider02",~~

~~"serviceProvider":"AgreementResponder",~~

~~"templateId":null,~~

~~"service":"service03",~~

~~"expirationTime":"2014-03-07T12:00:00CET"~~

~~},~~

~~"name":"ExampleTemplate2",~~

~~"terms":{…}~~

~~}~~

~~]~~

~~Return code: 404 Not found~~

~~Response format: application/xml~~

~~Response:~~

~~<?xml version="1.0" encoding="UTF-8"?>~~

~~<error code=”404”>~~

~~<message> There is no template with provider: {provider} in the SLA Repository Database</message>~~

~~</error>~~

#### Create a template

Create a template

**Request**

Method: POST

Path: /templates/

Incoming format: application/xml or application/json depending on content-type header

Incoming message: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<wsag:Template xmlns:wsag="http://www.ggf.org/namespaces/ws-agreement" TemplateId="template01">

<wsag:Name>ExampleTemplate</wsag:Name>

<wsag:Context>

<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>

<wsag:ServiceProvider>provider01</wsag:ServiceProvider>

<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>

<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>

<wsag:TemplateId>template01</wsag:TemplateId>

<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>

</wsag:Context>

<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>

</wsag:Template >

Incoming message: application/json

{

"templateId":"template02",

"context":{

"agreementInitiator":null,

"agreementResponder":"provider03",

"serviceProvider":"AgreementResponder",

"templateId":null,

"service":"service03",

"expirationTime":"2014-03-07T12:00:00CET"

},

"name":"ExampleTemplate2",

"terms":{ … }

}

Curl command :

curl –X POST -u user:password -H ["Content-type: application/xml"|"Content-type: application/json"] -H ["Accept: application/xml"|"Accept: application/json"] -d @template.file localhost:8080/sla-service/templates/

**Response**

Return code: 201

Response format: application/xml or application/json depending on the accept header

Response: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<message code="201" message="The agreement has been stored successfully in the SLA Repository Database. It has location http://localhost:8080/sla-service/templates/template012"/>

Response: application/json

{

"code":201,

"message":"The agreement has been stored successfully in the SLA Repository Database. It has location http://localhost:8080/sla-service/templates/template06"

}

Return code: 409

Response format: application/xml or application/json depending on the accept header

Response: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<error code="409" message="Element with id:template012 already exists in the SLA Repository Database"/>

Response: application/json

{

"code":409,

"message":"Element with id:template06 already exists in the SLA Repository Database"

}

#### Update a template

Update one or several fields of a template

**Request**

Method: PUT

Path: /templates/

Incoming format: application/xml or application/json depending on content-type header

Incoming message: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<wsag:Template xmlns:wsag="http://www.ggf.org/namespaces/ws-agreement" TemplateId="template01">

<wsag:Name>ExampleTemplate</wsag:Name>

<wsag:Context>

<wsag:AgreementInitiator>provider02</wsag:AgreementInitiator>

<wsag:ServiceProvider>provider01</wsag:ServiceProvider>

<wsag:ExpirationTime>2014-03-07T12:00:00+0100</wsag:ExpirationTime>

<wsag:ServiceProvider>AgreementInitiator</wsag:ServiceProvider>

<wsag:TemplateId>template01</wsag:TemplateId>

<sla:Service xmlns:sla="http://sla.atos.eu">service3</sla:Service>

</wsag:Context>

<wsag:Terms><wsag:All>….</wsag:All></wsag:Terms>

</wsag:Template >

Incoming message: application/json

{

"templateId":"template02",

"context":{

"agreementInitiator":null,

"agreementResponder":"provider03",

"serviceProvider":"AgreementResponder",

"templateId":null,

"service":"service03",

"expirationTime":"2014-03-07T12:00:00CET"

},

"name":"ExampleTemplate2",

"terms":{ … }

}

Curl command :

curl -u user:password -H ["Content-type: application/xml"|"Content-type: application/json"] -H ["Accept: application/xml"|"Content-type: application/json"] -d@template.file http://localhost:8080/sla-service/templates -X PUT

**Response**

Return code: 202

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<template>

<id>4</id>

<uuid>24ccd79a-537e-47ac-8450-3276f45186cf</uuid>

<text> </text>

</template>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to update the template</message>

</error>

#### Delete a template

Delete a template by its template-id.

**Request**

Method: DELETE

Path: /templates/

Parameter: template-id

Curl command :

curl –X DELETE localhost:8080/sla-service/templates/{template-id}

**Response**

Return code: 204

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<message>The template with id: {template-id} was delete successfully</message>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the template</message>

</error>

### Agreements

#### Get all the agreements

Get the entire set of agreements.

**Request**

Method: GET

Path: /templates

Curl command:

curl [–X GET] localhost:8080/sla-service/agreements

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection href="/agreements">

<items offset="0" total="2">

<Agreement xmlns="http://www.ggf.org/namespaces/ws-agreement" AgreementId="2498db54-3658-4ffc-a37d-b60003eaea77">

<Name>ExampleAgreement</name>

<Context>

<AggreementInitiator>RandomClient</AgreementInitiator>

<AgreementResponder>Provider01</AgreementResponder>

<ServiceProvider>AgreementResponder</ServiceProvider>

<ExpirationTime>2014-03-07-1200</ExpirationTime>

<TemplateId>contract-template-2007-12-04</<TemplateId>>

</Context>

</Agreement>

<Agreement xmlns="http://www.ggf.org/namespaces/ws-agreement" AgreementId="deb45c8c-db2f-4a22-8ef2-48f36b58c96d">

<Name>ExampleAgreement</name>

<Context>

<AggreementInitiator>RandomClient</AgreementInitiator>

<AgreementResponder>Provider01</AgreementResponder>

<ServiceProvider>AgreementResponder</ServiceProvider>

<ExpirationTime>2014-03-07-1200</ExpirationTime>

<TemplateId>contract-template-2007-12-04</<TemplateId>>

</Context>

</Agreement>

</items>

</collection>

#### Get an specific agreements by the agreement-id

For a given agreement-id identifier, get the specific template.

**Request**

Method: GET

Path: /agreements/{agreement-id}

Curl command :

curl [–X GET] localhost:8080/sla-service/agreements/{agreement-id}

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<Agreement xmlns="http://www.ggf.org/namespaces/ws-agreement" AgreementId="deb45c8c-db2f-4a22-8ef2-48f36b58c96d">

<Name>ExampleAgreement</name>

<Context>

<AggreementInitiator>RandomClient</AgreementInitiator>

<AgreementResponder>Provider01</AgreementResponder>

<ServiceProvider>AgreementResponder</ServiceProvider>

<ExpirationTime>2014-03-07-1200</ExpirationTime>

<TemplateId>contract-template-2007-12-04</<TemplateId>>

</Context>

</Agreement>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<error code="404" message="There is no agreement with id: deb45c8c-db2f-4a22-8ef2-48f36b58c96"/>

#### Get active agreements

Get active agreements basing in the field expiration-time of the agreement.

**Request**

Method: GET

Path: /agreements/

Query Param: active

Curl command :

curl [–X GET] localhost:8080/sla-service/agreements/?active=”{true}”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no active agreements in the SLA Repository Database</message>

</error>

#### Create an agreement

Create an agreement

**Request**

Method: POST

Path: /agreements/

Incoming format: application/xml

Incoming message:

Curl command :

curl –X POST –d@filename localhost:8080/sla-service/agreements/

**Response**

Return code: 201

Response format: application/xml

Response:

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to create the agreement</message>

</error>

#### Delete an agreement by the agreement-id

Delete an agreement by its agreement-id.

**Request**

Method: DELETE

Path: /agreements/

Parameter: agreement-id

Curl command :

curl –X DELETE localhost:8080/sla-service/agreements/{agreement-id}

**Response**

Return code: 204

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<message>The agreement with id: {agreement-id} was delete successfully</message>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the agreement</message>

</error>

### Enforcements

#### Get all the enforcements

Get the entire set of enforcements.

**Request**

Method: GET

Path: /templates

Curl command:

curl [–X GET] localhost:8080/sla-service/enforcements

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Problem:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/enforcementJobs">

<items offset="0" total="2">

<enforcement\_job>

<id>1</id>

<enabled>true</enabled>

<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>

**<agreement-id>----<agreement-id>**

</enforcement\_job>

<enforcement\_job>

<id>3</id>

<enabled>false</enabled>

<last\_executed>2013-11-30T00:00:00+01:00</last\_executed>

**<agreement-id>----<agreement-id>**

</enforcement\_job></collection>

#### Get a certain enforcement by agreement-id

For a given identifier agreement-id, get the specific enforcement.

**Request**

Method: GET

Path: /enforcements/{agreement-id}

Curl command :

curl [–X GET] localhost:8080/sla-service/enforcements/{agreement-id}

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no enforcements with id: {id} in the SLA Repository Database</message>

</error>

#### Create an enforcement

Create an enforcement

**Request**

Method: POST

Path: /enforcements/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

<id>1</id>

<enabled>true</enabled>

<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Curl command :

curl –X POST localhost:8080/sla-service/enforcements/

**Response**

Return code: 201

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

<id>4</id>

<enabled>true</enabled>

<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to create the enforcement</message>

</error>

#### Start an enforcement

Start an enforcement

**Request**

Method: PUT

Path: /enforcements/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

**<enabled>true</enabled>**

**<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>**

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Curl command :

curl –X PUT -d@filename localhost:8080/sla-service/templates/

**Response**

Return code: 202

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

<id>4</id>

**<enabled>true</enabled>**

**<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>**

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to start the enforcement</message>

</error>

#### Stop an enforcement

Stop an enforcement

**Request**

Method: PUT

Path: /enforcements/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

**<enabled>false</enabled>**

**<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>**

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Curl command :

curl –X PUT -d@filename localhost:8080/sla-service/templates/

**Response**

Return code: 202

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<enforcement\_job>

<id>4</id>

**<enabled>true</enabled>**

**<last\_executed>2014-11-30T00:00:00+01:00</last\_executed>**

**<agreement-id>----<agreement-id>**

</enforcement\_job>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to stop the enforcement</message>

</error>

#### Delete the enforcements of a specific provider

Delete the enforcements of a specific provider.

**Request**

Method: DELETE

Path: /enforcements/

Query Parameter: provider-id

Curl command :

curl –X DELETE localhost:8080/sla-service/enforcements/?provider=”{provider-id}”

**Response**

Return code: 202

Response format: application/xml

Response:

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the enforcement(s)</message>

</error>

### Providers

#### Get all the providers

Get the entire set of providers.

**Request**

Method: GET

Path: /providers

Curl command:

curl [–X GET] localhost:8080/sla-service/providers

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Problem:

<?xml version="1.0" encoding="UTF-8"?>

<collection href="/providers">

<items offset="0" total="60">

<provider>

<id>4</id>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider2</name>

</provider>

<provider>

<id>8</id>

<uuid>ad899b11-25a7-4c6d-a4d8-22e582859008</uuid>

<name>Provider3</name>

</provider>

<provider>

<id>12</id>

<uuid>0ef31dd0-3137-469f-998d-b40dcabfaa16</uuid>

<name>Provider5</name>

</provider>

</items>

</collection>

#### Get a specific provider

For a given identifier, get the specific provider.

**Request**

Method: GET

Path: /providers/{provider-id}

Curl command :

curl [–X GET] localhost:8080/sla-service/provider/{provider}

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/providers">

<items offset="0" total="60">

<provider>

<id>4</id>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider2</name>

</provider>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no provider with id: {id} in the SLA Repository Database</message>

</error>

#### Create a provider

Create a template

**Request**

Method: POST

Path: /templates/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/providers">

<items offset="0" total="60">

<provider>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider 50</name>

</provider>

Curl command :

curl –X POST localhost:8080/sla-service/providers/

**Response**

Return code: 201

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/providers">

<items offset="0" total="60">

<provider>

<id>4</id>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider 50</name>

</provider>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to create the provider</message>

</error>

#### Update a provider

Update a provider

**Request**

Method: POST

Path: /providers/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/providers">

<items offset="0" total="60">

<provider>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider 30</name>

</provider>

Curl command :

curl –X POST localhost:8080/sla-service/providers/

**Response**

Return code: 201

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/providers">

<items offset="0" total="60">

<provider>

<id>4</id>

<uuid>1ad9acb9-8dbc-4fe6-9a0b-4244ab6455da</uuid>

<name>Provider 30</name>

</provider>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to update the provider</message>

</error>

#### Delete a provider by the provider-id

Delete a provider by the provider-id.

**Request**

Method: DELETE

Path: /providers/

Parameter: provider-id

Curl command :

curl –X DELETE localhost:8080/sla-service/providers/{provider-id}

**Response**

Return code: 204

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<message>The provider with id: {provider-id} was delete successfully</message>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the provider</message>

</error>

#### Delete a provider by his name

Delete a provider by his name.

**Request**

Method: DELETE

Path: /providers/

Query Parameter: name

Curl command :

curl –X DELETE localhost:8080/sla-service/providers/?name=”{name}”

Return code: 204

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<message>The provider with name: {name} was delete successfully</message>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the provider</message>

</error>

### Violations

#### Get all the violations

Get the entire set of violations.

**Request**

Method: GET

Path: /templates

Curl command:

curl [–X GET] localhost:8080/sla-service/templates

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope</service\_scope>

<metric\_name>metric name</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<violation>

<id>14</id>

<uuid>dc61cff7-4fa6-436e-b913-29444a7aac8a</uuid>

<contract\_uuid>fa1641d7-ad7a-4587-87bb-0b623557d9cf</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope</service\_scope>

<metric\_name>metric name</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

</items>

</collection>

#### Get a specific violation by

For a given violation-id identifier, get the specific violation.

**Request**

Method: GET

Path: /violations/{violation-id}

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/{violation-id}

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope</service\_scope>

<metric\_name>metric name</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) with violation-id: {violation-id} in the SLA Repository Database</message>

</error>

#### Get a violation by agreement-id

For a given violation-id identifier, get the specific violation.

**Request**

Method: GET

Path: /violations/

Query param: agreement

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?agreement=”{agreement-id}”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) with agreement-id: {agreement-id} in the SLA Repository Database</message>

</error>

#### Get all the violations by service name

For a given service-name identifier, get the specific violation.

**Request**

Method: GET

Path: /violations/

Query param: service-name

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?service-name=”service-name”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 1</service\_scope>

<metric\_name>metric name 2</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 2</service\_scope>

<metric\_name>metric name 3</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<items>

<collections>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) with violation-id: {violation-id} in the SLA Repository Database</message>

</error>

#### Get violations given a range of dates

For a given service-name identifier, get the specific violation.

**Request**

Method: GET

Path: /violations/

Query parameters : [from], [until]

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?from=”fromDate”&until=”untilDate”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 1</service\_scope>

<metric\_name>metric name 2</metric\_name>

<date\_time>2013-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 2</service\_scope>

<metric\_name>metric name 3</metric\_name>

<date\_time>2013-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<items>

<collections>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) [from : {fromDate} ] [until: {untilDate} in the SLA Repository Database</message>

</error>

#### Get the violations of a specific provider

For a given provider, get the specific violation.

**Request**

Method: GET

Path: /violations/

Query parameters : provider

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?provider=”{provider-id}”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) of the provider-id: {provider-id} in the SLA Repository Database</message>

</error>

### Breaches

#### Get all the breaches

Get the entire set of Breaches.

**Request**

Method: GET

Path: /templates

Curl command:

curl [–X GET] localhost:8080/sla-service/breaches

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/breaches">

<items offset="0" total="2">

<breach>

<id>2</id>

<contract-uuid>b997304b-ba27-4e46-bd5d-f2d08ffdd2b7</contract-uuid>

<violation/>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>response time</metric-name>

<value>3</value>

</breach>

<breach>

<id>3</id>

<violation>

<id>1</id>

<contract\_uuid>b53d48b4-86a4-4a5a-a260-dddad47f4233</contract\_uuid>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>55</expected\_value>

<actual\_value>44</actual\_value>

</violation>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>breach name</metric-name>

<value>8.0</value>

</breach>

</items>

</collection>

#### Get a specific breach

For a given breach-id identifier, get the specific breach.

**Request**

Method: GET

Path: /breaches/{breach-id}

Curl command :

curl [–X GET] localhost:8080/sla-service/breaches/{breach-id}

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<breach>

<id>2</id>

<contract-uuid>b997304b-ba27-4e46-bd5d-f2d08ffdd2b7</contract-uuid>

<violation/>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>response time</metric-name>

<value>3</value>

</breach>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no breach(es) with breach-id: {breach-id} in the SLA Repository Database</message>

</error>

#### Get a breach by agreement-id

For a given agreement-id identifier, get the specific violation.

**Request**

Method: GET

Path: /breaches/

Query param: agreement

Curl command :

curl [–X GET] localhost:8080/sla-service/breaches/?agreement=”{agreement-id}”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no breach(es) with agreement-id: {agreement-id} in the SLA Repository Database</message>

</error>

#### Get all the breaches by metric name

For a given metric-name identifier, get the specific breach.

**Request**

Method: GET

Path: /breaches/

Query param: metric-name

Curl command :

curl [–X GET] localhost:8080/sla-service/breaches/?metric-name=”metric-name”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 1</service\_scope>

<metric\_name>metric name 2</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 2</service\_scope>

<metric\_name>metric name 3</metric\_name>

<date\_time>1970-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<items>

<collections>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) with violation-id: {violation-id} in the SLA Repository Database</message>

</error>

#### Get violations given a range of dates

For a given service-name identifier, get the specific violation.

**Request**

Method: GET

Path: /violations/

Query parameters : [from], [until]

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?from=”fromDate”&until=”untilDate”

**Response**

Return code: 200 OK

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<collection xmlns="http://eu.atos.sla.eu/doc/schemas/xml" href="/violations">

<items offset="0" total="2">

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 1</service\_scope>

<metric\_name>metric name 2</metric\_name>

<date\_time>2013-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<violation>

<id>7</id>

<uuid>05dbbbab-9a6e-4faf-b355-31050aa03fee</uuid>

<contract\_uuid>a9561a82-aeb8-483e-9521-3ec21ae7b0a7</contract\_uuid>

<service\_name>service name</service\_name>

<service\_scope>service scope 2</service\_scope>

<metric\_name>metric name 3</metric\_name>

<date\_time>2013-01-01T01:00:02+01:00</date\_time>

<expected\_value>expected value</expected\_value>

<actual\_value>value</actual\_value>

</violation>

<items>

<collections>

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no violation(s) [from : {fromDate} ] [until: {untilDate} in the SLA Repository Database</message>

</error>

#### Get the breaches of a specific range of dates

For a given range of dates, get the specific breach.

**Request**

Method: GET

Path: /violations/

Query parameters : until, from

Curl command :

curl [–X GET] localhost:8080/sla-service/violations/?until=”{until},from=”{from}””

**Response**

Return code: 200 OK

Response format: application/xml

Response:

Return code: 404 Not found

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”404”>

<message>There is no breach(es) in that range of dates: {until}, {from} in the SLA Repository Database</message>

</error>

#### Create a breach

Create a violation

**Request**

Method: POST

Path: /violations/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<breach>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>breach name</metric-name>

<value>8.0</value>

</breach>

Curl command :

curl –X POST localhost:8080/sla-service/breaches/

**Response**

Return code: 201

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<breach>

<id>50</id>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>breach name</metric-name>

<value>8.0</value>

</breach>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to create the breach</message>

</error>

#### Update a breach

Update a breach

**Request**

Method: PUT

Path: /breaches/

Incoming format: application/xml

Incoming message:

<?xml version="1.0" encoding="UTF-8"?>

<breach>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>breach name</metric-name>

<value>9.0</value>

</breach>

Curl command :

curl –X POST localhost:8080/sla-service/breaches/

**Response**

Return code: 201

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<breach>

<id>50</id>

<datetime>1970-01-01T01:00:12+01:00</datetime>

<metric-name>breach name</metric-name>

<value>9.0</value>

</breach>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to create the breach</message>

</error>

##### Delete a breach

Delete a breach by its breach-id.

**Request**

Method: DELETE

Path: /templates/

Parameter: template-id

Curl command :

curl –X DELETE localhost:8080/sla-service/breaches/{breach-id}

**Response**

Return code: 204

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<message>The breach with id: {breach-id} was delete successfully</message>

Return code: 500

Response format: application/xml

Response:

<?xml version="1.0" encoding="UTF-8"?>

<error code=”500”>

<message> It was no possible to delete the breach</message>

</error>

# SLA Management

## Implementation details

# Portal

The portal merely accesses the functionalities exposed by the SLA APIs. Every function performed through the Portal could be performed by the experimenter without using the Portal by issuing the respective http requests directly to the API.

## Portal Plugin

All of the portal’s views are implemented as plugins for the Django CMS software. These plugins are implemented in the file cms\_plugins.py.

A CMS plugin consist of two parts:

- The actual plugin code implemented as a class

- An HTML template that is used for rendering the plugin’s output.

For further information on how to write a plugin, please refer to the documentation of Django CMS here: [http://docs.djangocms.org/en/2.1.3/extending\_cms/custom\_plugins.html](http://docs.django-cms.org/en/2.1.3/extending_cms/custom_plugins.html)