

---

# **collective.restapi.linguaplone** **Documentation**

*Release 1.0a*

**Mikel Larreategi**

**Apr 27, 2018**



---

## Contents

---

<b>1</b>	<b>collective.restapi.linguaplone</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Installation . . . . .	3
1.3	Contribute . . . . .	3
1.4	License . . . . .	4
1.5	Contents . . . . .	4



build passing



---

## collective.restapi.linguaplone

---

An add-on providing plone.restapi endpoint for translations handled using [LinguaPlone](#)

### 1.1 Documentation

Full documentation for end users can be found in the “docs” folder, and is also available online at <https://collectiverestapilinguaplone.readthedocs.io>

### 1.2 Installation

Install collective.restapi.linguaplone by adding it to your buildout:

```
[buildout]

...

eggs =
    collective.restapi.linguaplone
```

and then running `bin/buildout`

### 1.3 Contribute

- Issue Tracker: <https://github.com/collective/collective.restapi.linguaplone/issues>
- Source Code: <https://github.com/collective/collective.restapi.linguaplone>
- Documentation: <https://collectiverestapilinguaplone.readthedocs.io>

## 1.4 License

The project is licensed under the GPLv2.

## 1.5 Contents

### 1.5.1 LinguaPlone Translations

---

**Note:** This addon will only work with `Products.LinguaPlone` in Plone 4. You can install `collective.restapi.pam` if you want to get similar features in Plone 4 using `plone.app.multilingual` versions 1.x and 2.x and use plain `plone.restapi` in Plone 5.

---

Using this addon you can get information about the translations of a content object handled using `Products.LinguaPlone`. To achieve that it provides a `@translations` endpoint to handle the translation information of the content objects.

Once we have installed `Products.LinguaPlone` and enabled more than one language we can link two content-items of different languages to be the translation of each other issuing a `POST` query to the `@translations` endpoint including the `id` of the content which should be linked to. The `id` of the content must be a full URL of the content object: `http`

```
POST /plone/en/test-document/@translations HTTP/1.1
Accept: application/json
Authorization: Basic YWRtaW46c2VjcmV0
Content-Type: application/json

{
  "id": "http://localhost:55001/plone/es/test-document"
}
```

curl

```
curl -i -X POST http://nohost/plone/en/test-document/@translations -H "Accept: application/json" -H "Content-Type: application/json" --data-raw '{"id": "http://localhost:55001/plone/es/test-document"}' --user admin:secret
```

httpie

```
http -j POST http://nohost/plone/en/test-document/@translations id=http://localhost:55001/plone/es/test-document -a admin:secret
```

python-requests

```
requests.post('http://nohost/plone/en/test-document/@translations', headers={'Accept': 'application/json'}, json={'id': 'http://localhost:55001/plone/es/test-document'}, auth=('admin', 'secret'))
```

---

**Note:** “id” is a required field and needs to point to an existing content on the site.

---

The API will return a `201 Created` response if the linking was successful.



```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://localhost:55001/plone/en/test-document

{}
```

After linking the contents we can get the list of the translations of that content item by issuing a GET request on the *@translations* endpoint of that content item.: http

```
GET /plone/en/test-document/@translations HTTP/1.1
Accept: application/json
Authorization: Basic YWRtaW46c2VjcmV0
```

curl

```
curl -i http://nohost/plone/en/test-document/@translations -H "Accept: application/
↪json" --user admin:secret
```

httpie

```
http -j http://nohost/plone/en/test-document/@translations -a admin:secret
```

python-requests

```
requests.get('http://nohost/plone/en/test-document/@translations', headers={'Accept':
↪'application/json'}, auth=('admin', 'secret'))
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "@id": "http://localhost:55001/plone/en/test-document",
  "language": "en",
  "translations": [
    {
      "@id": "http://localhost:55001/plone/es/test-document",
      "language": "es"
    }
  ]
}
```

To unlink the content, issue a DELETE request on the *@translations* endpoint of the content item and provide the language code you want to unlink.: http

```
DELETE /plone/en/test-document/@translations HTTP/1.1
Accept: application/json
Authorization: Basic YWRtaW46c2VjcmV0
Content-Type: application/json

{
  "language": "es"
}
```

curl

```
curl -i -X DELETE http://nohost/plone/en/test-document/@translations -H "Accept:
↪application/json" -H "Content-Type: application/json" --data-raw '{"language": "es"}'
↪' --user admin:secret
```

---

httpie

```
http -j DELETE http://nohost/plone/en/test-document/@translations language=es -a_↵
↵admin:secret
```

python-requests

```
requests.delete('http://nohost/plone/en/test-document/@translations', headers={'Accept
↵': 'application/json'}, json={'language': 'es'}, auth=('admin', 'secret'))
```

---

**Note:** “language” is a required field.

---

---

```
HTTP/1.1 204 No Content
```

---