


# Registering and Managing URS Links

- [URS API interface](#)
- [API Entry Point](#)
- [Methods](#)
  - [POST \(Create Redirect\)](#)
  - [PUT \(Update Redirect\)](#)
  - [GET \(View Redirect\)](#)
  - [DELETE \(Delete Redirect\)](#)
- [Security](#)

## URS API interface

- The purpose of the API is to be able to manage URS redirects. For this we need only a single endpoint.
- There will be a single resource exposed in the API -> Redirects:
  - GET: Used for retrieving the settings of a single redirect resource
  - POST: for creating new redirects
  - PUT: for updating an existing redirect
  - DELETE: for deleting a single redirect resource

## API Entry Point

 <http://urs.pbs.org/api/1.0/>

### Some generally applicable specifications

- All relative URLs below are relative to the API entry point.
- All DATETIME values are considered UTC in both inputs and outputs. DATETIME are in [W3-DTF](#) format which is a subset of ISO 8601

## Methods

### POST (Create Redirect)

**URL:** `/redirect/`

#### Required Inputs:

- ***original\_url***: URL
  - Not validated (i.e. for 404).

#### Optional inputs:

- ***geo\_blocking*** - can contain **one** of the following:
  - ***allow***: list of country codes OR
  - ***restrict***: list of country codesCountry codes can be found here: [http://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2)
- ***available\_start***: DATETIME
- ***available\_end***: DATETIME

#### Sample redirect creation payload:

```
{
  "original_url": "<URL>",
  "geo_blocking": {
    "allow": ["country code", "country code", ...],
  },
  "available_start": "2012-05-17T00:00:00",
  "available_end": "2012-08-17T00:00:00"
}
```

#### Output:

On success:

```
HTTP/1.0 201 CREATED
Date: <date created>
Server: <server information>
Content-Type: application/json
Location: <URS_BASE_URL>/api/1.0/redirect/<the_new_redirect_key>/

{
  "$self": "<URS_BASE_URL>/api/1.0/redirect/<the_new_redirect_key>/"
  "original_url": "<URL>",
  "protected_url": "<URL>",
  "geo_blocking": {
    "restrict": ["country code", "country code", ...],
  },
  "available_start": "2012-05-17T00:00:00",
  "available_end": null
}
```

On invalid payload:

```
HTTP/1.0 400 BAD REQUEST
Date: <date>
Server: <server information>
Content-Type: text/html; charset=utf-8
<message>
```

## PUT (Update Redirect)

**URL:** /redirect/<redirect\_key>/

**Optional inputs:**

At least one of the inputs should be present. An empty payload is invalid.

- **original\_url**: URL
  - We probably **should not** validate this (i.e. for 404).
- **geo\_blocking** - can contain **one** of the following:
  - **allow**: list of country codes
  - **restrict**: list of country codes
- **available\_start**: DATETIME
- **available\_end**: DATETIME

**Output:**

**On success:** HTTP 204

**On invalid key:** HTTP 404

**On invalid payload:** HTTP 400

## GET (View Redirect)

**URL:** /redirect/<redirect\_key>/

**Output:**

**On bad key:** HTTP 404

**On success:** HTTP 200

- **original\_url**
- **protected\_url**
- **geo\_blocking**
- **available\_start**
- **available\_end**
- **redirect\_key**

### Sample redirect GET response:

```
{
  "$self": "<URS_BASE_URL>/api/1.0/redirect/<redirect_key>/"
  "original_url": "<URL>",
  "protected_url": "<URL>",
  "geo_blocking": {
    "restrict": ["country code", "country code", ...],
  },
  "available_start": "2012-05-17T00:00:00",
  "available_end": null
}
```

## DELETE (Delete Redirect)

**URL:** /redirect/<redirect key>/

### Output:

#### On success:

```
HTTP/1.0 204 NO CONTENT
Date: <date>
Server: <server information>
Content-Length: 0
Content-Type: text/html; charset=utf-8
```

#### Bad redirect key:

HTTP 404

#### Already deleted:

Ideally HTTP 410 but almost surely have to use 404.

## Security

We will use a standard authentication method, 2-Legged OAuth for the URS API.

Resources:

- <http://pypi.python.org/pypi/oauth2/1.5.211>

Sample request:

```
GET *<URS_BASE_URL>/api/1.0/* HTTP/1.1
Host: <URS_BASE_URL>
Authorization: OAuth realm="<URS_BASE_URL>/api/1.0/",
  oauth_consumer_key="dpf43f3p2l4k3l03",
  oauth_nonce="kllo9940pd9333jh",
  oauth_timestamp="1191242096",
  oauth_signature_method="HMAC-SHA1",
  oauth_version="1.0",
  oauth_signature="tR3%2BTy81lMeYAr%2FFid0kMTYa%2FWM%3D"
```