

Ushahidi 3.x REST API

Table of Contents

- Resources
- General Patterns
 - Methods
 - Links
 - Response Codes
- Testing

Resources

- Ushahidi 3.x API Config Resource
- Ushahidi 3.x API Form Attributes Resource
- Ushahidi 3.x API Forms Resource
 - Ushahidi 3.x API Form Groups Resource
- Ushahidi 3.x API Media Resource
- Ushahidi 3.x API Messages Resource
- Ushahidi 3.x API Posts Resource
 - Ushahidi 3.x API Posts GeoJSON
 - Ushahidi 3.x Revisions Resource
 - Ushahidi 3.x Translations Resource
 - Ushahidi 3.x Updates Resource
- Ushahidi 3.x API Sets Resource
 - Ushahidi 3.x API Sets Posts Resource
- Ushahidi 3.x API Tags Resource
- Ushahidi 3.x API Users Resource

General Patterns

Methods

- Create record: POST /collection
- Get all records: GET /collection
- Get record: GET /collection/[id]
- Update record: PUT /collection/[id]
- Delete record: DELETE /collection/[id]
- Search records: GET /collection?q=something&type=report

Links

- All records should return a link as part of the response. This will usually be an 'url' field
- They will also return links for related records.
 - Related records will be returned as an object with both ID and URL
 - Related records can be submitted with just the ID or ID within the object.

Response Example

```
{
  "id":1,
  "url":"http://ushv3.dev/api/v2/posts/1",
  "form":{
    "id":1,
    "url":"http://ushv3.dev/api/v2/forms/1",
  }
}
```

Response Codes

- 200 OK
- 404 Not found
- 400 Bad Request - includes invalid json, or invalid attributes supplied
- 405 Method not supported
- 500 Server error
- 401 Authentication Required

Testing

API tests are written in behat.

Each resources url / collection should test at least the following:

- Create a record
- Get all records
- Get a record
- Update a record
- Delete a record
- Search for a record
- Failure conditions
 - Get non-existent record
 - Create a record with invalid data
 - Delete non-existent record

Possible extra tests

- Test a record update which removing an attribute (if applicable)
- Different formats (ie. geojson for posts)