

GeoT API (draft)

1 Preamble

The API is purely REST based. POST requests are used for uploading data and a "unique token", henceforth called `uuid`, for session tracking. Output is `application/json`. All examples have been tested with `curl` or `XHR AJAX` http clients.

Initially a media file must be uploaded in order to obtain a `uuid` associated with the resource. This `uuid` can later be used to perform all other operations like: - Getting all probed metadata associated with the file - Changing existing metadata. - Adding new metadata.

1.1 URL:

All API calls are currently served under the test URL: `devel.edina.ac.uk:9110`

2 API Calls

2.1 uploadfile

A media file is POSTed using the Content-Type `multipart/form-data` according to RFC 2388. Upon success, a `uuid` is returned referring to the file for further API calls:

1. POST a file `/ws/uploadfile`
2. Return call is a JSON object:

```
{ uuid:<the uuid>, error:0, msg:<error message> } on success or  
{ uuid:0, error: ERROR_CODE } on failure
```

error codes are:

```
0: success (uuid is valid)  
1: unsupported media file  
2: internal error
```

Example with `curl`:

```
curl --form fileInput=@MyMedia.jpg <host:port>/ws/uploadFile
```

returns:

```
{"msg": "Success!", "uuid": "09874a957f6a49bba87c96e548f72d6b", "error": 0}
```

2.2 loadMetadata(uuid)

Get all probed properties of mediafile described by `<uuid>` and return them as a json file

Example with `curl`:

```
curl 127.0.0.1:8080/ws/loadMetadata/09874a957f6a49bba87c96e548f72d6b
```

Returns:

```
{"FileName": "MyMedia.jpg", "FileType": "JPEG", "MIMEType": "image/jpeg",  
"YResolution": 180, "ResolutionUnit": 2, "FocalPlaneYSize": 15.494, ..... }
```

2.3 saveMetadata(uuid,new_metadata):

A POST request of a json object containing the updated metadata fields. Only the changed values should be submitted.

Example with curl:

```
curl --data 'metadata={"Author":"Foo Bar Baz"}' \
127.0.0.1:8080/ws/saveMetadata/09874a957f6a49bba87c96e548f72d6b'
```

Returns:

```
{"msg": "Success!", "uuid": "09874a957f6a49bba87c96e548f72d6b", "error": 0}
```

3 Caveat Emptor

3.1 Supported Metadata

While a whole slew of different media files can be probed for their metadata, not all can be modified or written. GeoT uses the `exiftool` at its core for metadata processing. Therefore one should look at the [upstream documentation](#) for a list of supported tags and filetypes.
