Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 14, 2014

J. Fox, Ed.
J. Fox
ciferproject.org
February 10, 2014

# CIFER API Framework 00

**ciferproject-org-framework-api-v1-draft-00**

## Abstract

The CIFER API Framework gives a consistant look, feel, syntax and nomenclature to all of the various CIFER resource representations.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 14, 2014.

## Copyright Notice

## Table of Contents



## 1. Introduction

The [CIFER] initiative includes work to define a set of shared APIs for core functions in identity and access management. One desired feature of such systems is a consistancy tructure and vocabulary of API URLs and representations.

## 2. Definitions

RESTful

A RESTful service is composed of resources. The state of a resource is defined by one or more representations and identified by one or more URIs.

Clients of the service may:

- Search for resources,
- Get a representation of a resource,
- Add or replace a resource, and
- Delete a resource.

## 3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 4. API Conventions

## 4.1. Representations

Representations in the CIFER APIs follow these conventions, except as noted in a specific API document.

1. Character set is UTF-8.
2. Default language is JSON. XML is optional.
3. Server major versions are indicated in the URI path, e.g. /v2/, and differentiate incompatible representations, parameters or headers.
4. Minor versioning is indicated in the responseMetadata---not in the URL.
5. Field names are camelCase (including acronyms), e.g. "selfUri"
6. Field names must start with a letter. They can contain upper/lower letters, numbers and underscores.
7. Field type, e.g. string, date,..., cannot change without a major version change.
8. Timestamps are ISO 8601, e.g.: 2012-10-04T03:10:14.123Z
9. Durations are ISO 8601, e.g.: P1.81S
10. Objects are identified by uri

    - e.g.: netId:mchyzer, or userId:12345678

11. The suggested hierarchical path separator is a colon ":".
    - An actual colon in a name part is represented by slash-colon ("\:").

## 4.2. Requests

Requests in the CIFER APIs follow these conventions, except as noted in a specific API document.

1. Full resources are identified by URI authority and path---not by query string or fragment.
2. The URL of a partial resource, e.g., a paged response, is further identified by query string parameters.
3. An XML representation may requested by header:
    - Accept: text/xml
4. If the request method is POST, the method may be overridden by a header:
    - X-HTTP-Method-Override: GET | PUT | DELETE
5. A client may request to "act as" another user via a header:
    - X-CIFER-Act-As: user_uri
6. A client may define or override request headers with query string parameters
    - e.g.: "&Accept=application%2Fjson&"
7. Boolean parameters are generously interpreted:
    - true|false, or t|f or yes|no, or y|n or 1|0, case insensitive
8. Use of ETag and If-Match:
    - A request to replace a resource must specify an If-Match header.
        - The value must match the resource's current ETag value, or
        - The value may be a 'match-any' wildcard, '*'
        - Acceptance of wildcard is service dependent.
        - A request to add a resource must not specify an If-Match header.
9. A PUT request need not specify all attributes.
    - Replacing a resource will not affect unspecified attributes.
    - A necessary attribute not specified will return an error 400.
10. Searches return resource URIs and default resource attributes.
    - Each service will document the fields returns on searches.
    - Searches may request "&extraFields=field3,field4,..&" to get those fields in addition to the default set.
    - Searches may request "&omitFields=field3,field4,..&" to omit those fields from the default set.
    - Support for these features is optional.
11. Searches may specify paging options by QS parameters
    - pageLimit: Number of entries per page
    - pageOffset: page number, first page is zero
12. Requests may specify field filters on returned resources.
    - Query string may specify a list desired fields: "&fields=field1,field2,...&"
    - A requests for an unknown field is not an error.
    - Support for this features is optional.
13. Requests for large resources, e.g. group membership, may specify paging parameters
    - sortField:
    - sortOrder: 'a*', 'd*', case insensitive
    - pageLimit: number of entries per page, zero means no paging
    - Mutually exclusive page start:
        - pageOffset: page number, starting with zero, OR
        - pageStartValue: first value returned will be the one AFTER this value

        - The service may provide default paging parameters, and may reduce the requested limits.

14. A client may request a pretty-printed response with the query string, "indent=true".
15. Unexpected Query String parameters will be ignored.

## 4.3. Responses

Responses in the CIFER APIs follow these conventions, except as noted in a specific API document.

1. Resources are wrapped in an object which contains:
    - resourceMetadata: metadata about the resource. fields as appropriate to the resource
        - lastModified: timestamp when the resource was modified
        - selfUri: URI to the current resource.
        - ETag, or equivalent, for resources that can be replaced, ...
    - responseMetadata: metadata about this particular response
        - responseTimestamp:
        - responseTime: time that the server took in processing this request
        - requestCommentary: freeform text about the request that the server processed (for debug)
        - httpStatusCode: response status as an HTTP status code.
        - serverVersion: major.minor server version number
        - paging parameters
            - pageLimit
            - pageOffest
            - sortField
            - pageOffsetValue
            - sortOrder
        - ...
    - resource: The actual resource for the request.
2. Response codes:
    - 200: the resource was found
    - 201: the resource was created
    - 400: the request was not valid
    - 401: the client was unauthenticated
    - 403: an authenticated client was not authorized for the request
    - 409: (conflict)
        - PUT to an existing resource without an If-Match
    - 412: (precondition failed)
        - If-Match header did not match resource's ETag, or
        - If-Match header supplied but resource not found (add with an if-match), or
        - wildcard if-match not allowed
    - 500: server error

# 5. Examples

## 5.1. Site directory

What might be returned by a top-level GET to a service.

```
{
  "resource":{
    "people": "/people",
    "places": "/places"
  },
  "resourceMetadata":{
    "lastModified":"2012-11-04T09:57:03.541Z",
    "baseUri":"https://example.u.edu/ppservice/v1/",
```

```
  },
  "responseMeta":{
    "responseTimestamp":"2013-07-04T09:57:03.541Z",
    "httpStatusCode":200,
    "responseTime":"P0.011S"
    "serverVersion":"1.1"
  }
}
```

## 5.2. Search response

What might be returned by a search GET to a service.

```
{
  "resource":[
   {"uuid": "3ac85497-1663-4ef4-9933-3df8bc7800bc",
    "lastName": "Doe",
    "givenName": "John A.",
    "address": { ... }
    "status": "something",
    "uri": "/v1/people/3ac85497-1663-4ef4-9933-3df8bc7800bc"
   },
   {"uuid": "3ac85497-1663-4ef4-9933-3df8bc7800cc",
    "lastName": "Doe",
    "givenName": "John B.",
    "address": { ... }
    "status": "something",
    "uri": "/v1/people/3ac85497-1663-4ef4-9933-3df8bc7800cc"
   },
  ],

  "resourceMetadata":{
    "baseUri":"https://example.u.edu/ppservice",
    "sortOrder":"a",
    "pageLimit":100,
    "pageOffset":0,
    "sortField":"name",
    "statusCode":"SUCCESS",
  },
  "responseMeta":{
    "responseTimestamp":"2013-07-04T09:57:03.541Z",
    "httpStatusCode":200,
    "responseTime":"P0.011S"
    "serverVersion":"1.1"
  }
}
```

# 6. IANA Considerations

This memo includes no request to IANA.

# 7. Security Considerations

The API framework does not specify how services identify their clients.

# 8. Acknowledgements

This document was produced by the CIFER Shared API work team. People who made particular contributions to the work include Jim Fox, Keith Hazelton, Chris Hyzer, Benn Oshrin, Jonathan Pass and Jimmy Vuccolo

# 9. References

## 9.1. Normative References

**[RFC2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 9.2. Informative References

**[CIFER]** CIFER, "CIFER Home Page", 2014.

## Authors' Addresses

**Jim Fox** (editor)
ciferproject.org
4333 Brooklyn Ave NE
Seattle, Washington 98195
US
Phone: +1 206 543 4320
EMail: fox@uw.edu

**Jim Fox**
ciferproject.org
EMail: fox@uw.edu