

# API for creating and updating resources from external sources - experimental

- The ei-x JSON request and response formats
  - [EIExchangeInstance](#) field descriptions:
  - [InstanceUpdateResponse](#) field descriptions
- REST service description
  - Obtain an ei-x representation of a resource
  - Create or update a resource in an eagle-i repository
  - Authentication and headers
- ei-x examples

These RESTful services use a JSON exchange format (ei-x) to allow external parties to create and update resources in an eagle-i repository. They are currently deployed in the development server:

<https://montana.dev.eagle-i.net>

IMPORTANT: use https and not http - there is a redirect in place from http to https, and many REST clients that follow redirects do not forward POSTed data.

## Known issues:

- error codes not properly handled
- currently no ontology-level validation of externally-produced resource descriptions
- currently does not handle links to referenced taxonomy terms (as opposed to links to eagle-i resources). Example of record not handled due to the related technique field: <http://montana.dev.eagle-i.net/i/0000012b-00be-dcdf-df3b-3fdc80000009>
- we currently do not enforce the association URI-foreign key (this will come)
- DELETE is not yet implemented
- **This is not thoroughly tested**

## The ei-x JSON request and response formats

ei-x is comprised of two classes:

`EIExchangeInstance` -- a simplified container for the RDF of an eagle-i resource \*and\* minimal information of directly linked resources -- also referred to as primary resource and secondary resources. Information for secondary resources includes \*only\* text properties (i.e. no tertiary resources!)

`InstanceUpdateResponse` -- encapsulates the status of requested updates

Browse the Java code from which these are serialized:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/datatools/catalyst/src/main/java/org/eaglei/datatools/services/>

## EIExchangeInstance field descriptions:

### instanceEntity -- EIEntity

A container for the primary resource's EIURI and Label; when requesting creation of a new resource, the `value` field of the EIURI should be empty, in which case the resource will be assigned a new eagle-i URI.

### specificTypeUri -- EIURI

The most specific ontology type assigned to the resource. The type EIURI can be obtained from the [ontology browser](#) (grab it from the URL bar), e.g. DNA Sequencer: [http://purl.obolibrary.org/obo/OBI\\_0400103](http://purl.obolibrary.org/obo/OBI_0400103)

### rootType -- EIURI

The root type of the resource (one of the eagle-i Primary Types), e.g. Instrument: [http://purl.obolibrary.org/obo/ERO\\_0000004](http://purl.obolibrary.org/obo/ERO_0000004)

### foreignKey -- String

A string that uniquely identifies the resource in its external location (format determined by the source)

### source -- String

A string that uniquely identifies the source of the resource (format TBD)

**linkedResources -- Map<String, EIExchangeInstance>**

A map that contains minimal information of secondary resources. It is keyed on the URI of the ontology property that associates the secondary resource to the primary resource (the URI is a String and not an EIURI, as complex map keys are not well supported by GSON)

**textProperties -- Map<String, String>**

A map that contains the value of text properties, keyed on the URI of the ontology property used to capture them.

## InstanceUpdateResponse field descriptions

All these maps are keyed on the resource's (primary or secondary) eagle-i URI.

**newlyCreatedUriToForeignKey -- Map<String, String>**

Newly created URIs and their foreign keys. Note that there can be multiple eagle-i URIs for a single foreign key (to cover the use case of an external resource being split into more granular eagle-i records)

**updatedUriToForeignKey -- Map<String, String>**

Updated resource URIs and their foreign keys. Same remark as above.

**failures -- Map<String, String>**

the eagle-i URI of the resource that failed, and an error message

## REST service description

### Obtain an ei-x representation of a resource

**URL:**

`/sweet/api/externalResource`

**Method:**

GET

**Args:**

`uri` -- the eagle-i identifier of the resource

**Example:**

<https://montana.dev.eagle-i.net/sweet/api/externalResource?uri=http://montana.dev.eagle-i.net/i/0000012f-989d-df6c-b822-788380000000>

**Result:**

EIExchangeInstance in JSON

### Create or update a resource in an eagle-i repository

**URL:**

`/sweet/api/externalResource`

**Method:**

POST

**Content-type:**

`application/json`

**Payload:**

The EExchangeInstance JSON of the primary resource and secondary resources to be created/updated (no instance URI indicates that the resource should be created). Note that for updates, the API user must have claim permission over the resources (current user is hardcoded in the backend)

#### Result:

InstanceUpdateResponse in JSON, indicating successes and failures

## Authentication and headers

These RESTful services use Http Basic Authentication, that is, they expect the eagle-i credentials to be passed in the `authorization` header of each request. You **must** use https in order for these credentials to be encrypted.

The POST service expects content to be passed as a JSON string, hence the Content-type HTTP header must be set to `application/json`

REST clients should be able handle this out-of-the box.

Example GET CURL command, substitute your own credentials:

```
curl -v --get -u eagle-i-user:password -d uri=http://montana.dev.eagle-i.net/i/0000012f-989d-df6c-b822-788380000000 https://montana.dev.eagle-i.net/sweet/api/externalResource
```

Example POST CURL command, substitute your own credentials (assumes a valid json instance in the file json.txt):

```
curl -v -X POST -u eagle-i-user:password -H "Content-Type: application/json" -d @json.txt https://montana.dev.eagle-i.net/sweet/api/externalResource
```

## ei-x examples

Sample New instance with new linked instances. New instances must contain a foreign key and no eagle-i uri.

```
{
  "source": "OfficialSourceString",
  "foreignKey": "OfficialForeignKey123",
  "linkedResources": {
    "http://purl.obolibrary.org/obo/ERO_0000022": [
      {
        "source": "OfficialSourceString",
        "foreignKey": "OfficialForeignKey456",
        "linkedResources": {

        },
        "instanceEntity": {
          "uri": {
            "id": ""
          },
          "label": "Frog, Kermit, Ph.D."
        },
        "specificTypeUri": {
          "id": "http://xmlns.com/foaf/0.1/Person"
        },
        "textProperties": {
          "http://purl.obolibrary.org/obo/ERO_0000052": [
            "Professor",
            "Specialist in CRCH"
          ],
          "http://purl.obolibrary.org/obo/ERO_0000042": [
            "Kermit"
          ],
          "http://purl.obolibrary.org/obo/ERO_0000041": [
            "kermit@hawaii.edu"
          ],
          "http://purl.obolibrary.org/obo/ERO_0000051": [
            "(123) 456-3008"
          ]
        }
      }
    ]
  }
}
```

```

    ],
    "http://purl.obolibrary.org/obo/ERO_0000049":[
        "Frog"
    ],
    "http://purl.obolibrary.org/obo/ERO_0000040":[
        "Cancer Research Center\n12345 Lauhala St.\nR504c\nHonolulu, HI 96813"
    ]
},
"rootType":{
    "id":"http://xmlns.com/foaf/0.1/Person"
}
}
],
"http://purl.obolibrary.org/obo/ERO_0000066":[
    {
        "source":"OfficialSourceString",
        "foreignKey":"OfficialForeignKey789",
        "linkedResources":{

        },
        "instanceEntity":{
            "uri":{
                "id":""
            },
            "label":"University of Hawaii at Manoa Organized Research Units"
        },
        "specificTypeUri":{
            "id":"http://vivoweb.org/ontology/core#Consortium"
        },
        "textProperties":{

        },
        "rootType":{
            "id":"http://xmlns.com/foaf/0.1/Organization"
        }
    },
    {
        "source":"OfficialSourceString",
        "foreignKey":"OfficialForeignKey101112",
        "linkedResources":{

        },
        "instanceEntity":{
            "uri":{
                "id":""
            },
            "label":"University of Hawai'i Cancer Center"
        },
        "specificTypeUri":{
            "id":"http://vivoweb.org/ontology/core#Center"
        },
        "textProperties":{
            "http://purl.obolibrary.org/obo/ERO_0000480":[
                "http://www.kermitcancercenter.org/"
            ],
            "http://purl.obolibrary.org/obo/ERO_0000051":[
                "678-910-1112"
            ],
            "http://eagle-i.org/ont/app/1.0/synonym":[
                "KCC"
            ],
            "http://purl.obolibrary.org/obo/ERO_0000040":[
                "123 Ala Moana Blvd.\nHonolulu, Hawai'i 96813"
            ]
        },
        "rootType":{
            "id":"http://xmlns.com/foaf/0.1/Organization"
        }
    }
}
],

```

```

"instanceEntity":{
  "uri":{
    "id":""
  },
  "label":"Kermit Analytical Laboratory"
},
"specificTypeUri":{
  "id":"http://vivoweb.org/ontology/core#CoreLaboratory"
},
"textProperties":{
  "http://purl.obolibrary.org/obo/ERO_0000055":[
    "Kermit Cancer Research Center, 1234 Lauhala St. R504c, Honolulu, HI 96813"
  ],
  "http://eagle-i.org/ont/app/1.0/resource_description":[
    "\Awesome facility.\"
  ],
  "http://purl.obolibrary.org/obo/ERO_0000054":[
    "Excel"
  ],
  "http://purl.obolibrary.org/obo/ERO_0000041":[
    "kermit@hawaii.edu"
  ],
  "http://purl.obolibrary.org/obo/ERO_0000480":[
    "http://www.kermit.org/"
  ],
  "http://eagle-i.org/ont/app/1.0/synonym":[
    "Small molecule analysis lab"
  ],
  "http://purl.obolibrary.org/obo/ERO_0000040":[
    "Kermit Cancer Research Center, 1236 Lauhala St. R504c, Honolulu, HI 96813"
  ]
},
"rootType":{
  "id":"http://xmlns.com/foaf/0.1/Organization"
}
}
}

```