Electronic Honest Broker Technical Manual

Aaron J Masino, PhD

6/16/11

Abstract

Contents

Chapter 1

API Requests

This chapter details the http request types that can be made to the eHB service. All requests are expected to be in HTTP format. Each section details the URL to which the request can be submitted, any necessary header information, the supported request body formats, and expected response formats. Note, **URLs must always end in a / character**. In all of the sections, it is assumed that the eHB is hosted at ehb-host, so that all requests will be of the form http://ehb-host/path/ where path is determined by the request type. Requests requiring a body will return a 422 UNKNOWN STATUS CODE error if the request body is malformed. For POST and PUT responses, the response body may include an "errors" element that includes error codes. The error codes and corresponding explanations are given in section 1.10

1.1 URL Summary

http://ehb-host/api/subject/

http://ehb-host/api/subject/id/#/

http://ehb-host/api/subject/organization/#/osid/#

http://ehb-host/api/externalsystem/

http://ehb-host/api/externalsystem/id/#/

http://ehb-host/api/externalsystem/query/

http://ehb-host/api/externalsystem/id/#/subjects/

http://ehb-host/api/externalsystem/id/#/organization/#/subjects/

http://ehb-host/api/externalsystem/id/#/records/

http://ehb-host/api/externalsystem/id/#/organization/#/records/

http://ehb-host/api/externalrecord/

http://ehb-host/api/externalrecord/id/#/

http://ehb-host/api/externalrecord/query/

http://ehb-host/api/organization/

http://ehb-host/api/organization/id/#/

http://ehb-host/api/organization/query/

http://ehb-host/api/group/

1.2 Subject Requests

The following sections detail the requests that can be made on eHB Subject records. The eHB Subject record is a patient identifier record with the following fields: first_name, last_name, dob, organization, organization_subject_id, created, modified. The eHB ensures that the combination of organization and organization_subject_id are unique and will return an appropriate error message for any action that would result in a duplicate entry.

1.2.1 Obtaining Subject Records Using GET

GET Requests

The URLs for GET requests are:
ehb-host/api/subject/id/#/ where # is the eHB Subject record id
ehb-host/api/subject/organization/#1/osid/#2 where #1 is the id for the eHB organization record, and #2 is the value of the field organization_subject_id for the subject of interest.

Individual Subject record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml. If the eHB is unable to find the Subject record, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
{
    "created": "yyyy-mm-dd hh:mm:ss",
    "dob": "yyyy-mm-dd",
    "first_name": "value",
    "id": "value",
    "last_name": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "organization_id": "ehb_id",
```

```
    "organization_subject_id": "value",
}
```

application/xml response:

```
<root>
    <first_name>value</first_name>
    <last_name>value</last_name>
    <created>yyyy-mm-dd hh:mm:ss</created>
    <dob>yyyy-mm-dd</dob>
    <modified>yyyy-mm-dd hh:mm:dd</modified>
    <organization_id>ehb_id</organization_id>
    <organization_subject_id>value</organization_subject_id>
    <id>value</id>
</root>
```

where id is the eHB Subject record id.

1.2.2 Creating Subject Records Using POST

The URL for this request is: **ehb-host/api/subject/**

Subject records are created using a POST request. A single request can be used to create one or more new Subject records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"first_name":"value","last_name":"value",
"organization":"ehb_id","organization_subject_id":"value","dob":"yyyy-mm-dd"},
{"first_name":"value","last_name":"value",
"organization":"ehb_id","organization_subject_id":"value","dob":"yyyy-mm-dd"}]
```

If it is not of this form, most likely a 500 Server Error response will be returned.
If the request form is correct, the response will be a JSON array of the form

```
[{"id":"value,"organization_subject_id": "value", "organization_id":"value",
"success": true, "created": "yyyy-mm-dd hh:mm:ss",
"modified": "yyyy-mm-dd hh:mm:ss"},
{"organization_subject_id": "value", "organization_id":"value", "success":
false, "errors": [{"error_label": errorcode}]}]
```

where id is the eHB Subject record id if successfully created.

Each entry in the response is associated with an organization_subject_id in the request and indicates whether the record was successfully create via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.2.3 Modifying Subject Records Using PUT

The URL for this request is: **ehb-host/api/subject/**

Subject records can be modified using a PUT request. A single request can be used to modify one or more Subject records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"id":"value",
"subject":{"first_name":"value","last_name":"value",
"organization":"ehb_id","organization_subject_id":"value","dob":"yyyy-mm-dd"}},
{"id":"value","subject":{"first_name":"value"}}]
```

where the "id" field indicates the eHB record id for the Subject record to be modified and the "subject" object contains the updated values for each Subject field. Note, as seen in the second entry, it is only necessary to specify the information that is going to be updated. The remaining field values will be unchanged. If the request is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value", "success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"id": "value", "success": false, "errors":
       [{"label": errorcode}]
},
{"id": "value", "success": false,"errors":
      [{"label": errorcode}]
}]
```

Each entry in the response is associated with an id in the request and indicates whether the record was successfully modified via the "success" field. If the record was not successfully modified the "errors" field will provide information as to why.

1.2.4 Deleting Subject Records Using DELETE

The URLs for Subject DELETE requests are:
ehb-host/api/subject/id/#/ where # is the Subject record id
ehb-host/api/subject/organization/#1/osid/#2 where #1 is the id for the eHB organization record, and #2 is the value of the field organization_subject_id for the subject of interest.

Individual Subject record data can be deleted using a DELETE request. The DELETE requests do not contain a body but must include the 'Accept' value in the header information. Currently, the supported accept types are: */*, application/json and application/xml. If the eHB is able to find the Subject record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.3 ExternalSystem Requests

The following sections detail the requests that can be made on eHB ExternalSystem records. The eHB ExternalSystem record is a system (e.g. database, webservice) identifier record with the following fields: name, description, created, modified. The eHB ensures that the name entries are unique and will return an appropriate error message for any action that would result in a duplicate external system name.

1.3.1 Obtaining External System Records Using GET or POST

GET Requests

The URL for a GET requests is:
ehb-host/api/externalsystem/id/#/
where # is the ExternalSystem record id

Individual ExternalSystem record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the ExternalSystem record, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
{
    "created": "yyyy-mm-dd hh:mm:ss",
    "description": "value",
    "id": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "name": "value",
```

```
    "url": "value"
}
```

application/xml response:

```
<root>
   <id>value</id>
   <description>value</description>
   <name>value</name>
   <modified>yyyy-mm-dd hh:mm:ss</modified>
   <created>yyyy-mm-dd hh:mm:ss</created>
   <url>value</url>
</root>
```

where id is the ExternalSystem record id.

Queries with Post

The URL for this request is: **ehb-host/api/externalsystem/query/**

If the ExternalSystem record id is not known, the ExternalSystem record information can be obtained using a POST request if the External System name or URL is known. A single request can be used to obtain one or more ExternalSystem records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON If the content type is specified as application/json, then the request body must be of the following form:

```
[{"name":"value"},
{"url":"value"},
{"name":"value"}]
```

If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"name": "value",
  "externalSystem": {
     "created": "yyyy-mm-dd hh:mm:ss",
     "description": "value",
     "id": "value",
     "modified": "yyyy-mm-dd hh:mm:ss",
     "name": "value",
     "url": "value
   }
},
{"url": "value",
```

```
  "externalSystem": {
     "created": "yyyy-mm-dd hh:mm:ss",
     "description": "value",
     "id": "value",
     "modified": "yyyy-mm-dd hh:mm:ss",
     "name": "value"
     "url": "value"
   }
}
{"name": "value",
 "errors": [{"label":errorcode}]
}]
```

1.3.2 Creating ExternalSystem Records Using POST

The URL for this request is: **ehb-host/api/externalsystem/**

External system records are created using a POST request. A single request can be used to create one or more new ExternalSystem records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"name":"value","description":"value", "url":"value"},
{"name":"value","description":"value", "url":"value"}
]
```

If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value","name": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"name": "value","success": false,
"errors": [{"label":errorcode}]
}]
```

where id is the ExternalSystem record id if successfully created.

Each entry in the response is associated with an name in the request and indicates whether the record was successfully create via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.3.3 Modifying ExternalSystem Records Using PUT

The URL for this request is: **ehb-host/api/externalsystem/**

ExternalSystem records can be modified using a PUT request. A single request can be used to modify one or more ExternalSystem records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"id":"value",
"external_system":{"name":"value","description":"value","url":"value"}},
{"id":"value", "external_system":{"name":"value"}}, {"id":"value","external_system":{"name"
```

where the "id" field indicates the record id for the ExternalSystem record to be modified and the "external_sytem" object contains the updated values for each ExternalSystem field. Note, as seen in the second entry, that it is only necessary to specify the information that is going to be updated, the remaining field values will be unchanged. If the request is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"id": "value","success": false,
"errors":[{"label":errorcode}]},
{"id": "value","success": false,
"errors":[{"label":errorcode}],
}]
```

Each entry in the response is associated with an id in the request and indicates whether the record was successfully modified via the "success" field. If the record was not successfully modified the "errors" field will provide information as to why.

1.3.4 Deleting ExternalSystem Records Using DELETE

The URL for a GET requests is:
ehb-host/api/externalsystem/id/#/
where # is the ExternalSystem record id

Individual ExternalSystem record data can be deleted using a DELETE request. The DELETE requests do not contain a body but must include the 'Accept' value in the header information. Currently, the supported accept types

are: */*, application/json and application/xml. If the eHB is able to find the ExternalSystem record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.4 ExternalRecord Requests

The following sections detail the requests that can be made on eHB ExternalRecord records. The eHB ExternalRecord record is links eHB Subject records to ExternalSystem records. ExternalRecords records have the following fields: subject, externalSystem, record_id, created, modified. Note that record_id is the id of the subject's record in the external system. The eHB ensures that the for a given ExternalSystem that record_ids are unique and will return an appropriate error message for any action that would result in a duplicate external system name.

1.4.1 Obtaining External Records Using GET or POST

GET Requests

The URL for a GET requests is:
ehb-host/api/externalrecord/id/#/
where # is the ExternalRecord record id in the eHB

Individual ExternalRecord record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the ExternalSystem record, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
{
    "created": "yyyy-mm-dd hh:mm:ss",
    "external_system_id": "value",
    "id": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "path": "value",
    "record_id": "value",
    "subject_id": "value"
}
```

application/xml response:

```
<root>
   <external_system_id>value</externalSystem_id>
   <created>yyyy-mm-dd hh:mm:ss</created>
   <subject_id>value</subject_id>
```

```
   <path>value</path>
   <modified>yyyy-mm-dd hh:mm:ss</modified>
   <record_id>value</record_id>
   <id>value</id>
</root>
```

where id is the ExternalSystem record id.

Queries with Post

The URL for this request is: **ehb-host/api/externalrecord/query/**

If the ExternalRecord eHB record id is not known, the ExternalRecord record information can be obtained using a POST request specifying one or more of the folowing: ExteranalSystem, Subject, path (i.e. the ExteranalRecord.path field value which is the path the record on the ExternalSystem). The subject identifier options are either the subject_id (i.e. the eHB subject record id) or subject_org (i.e. the eHB Organization id for the Organization associated with this Subject) and subjet_org_id (i.e. the record id for the Orginization for this subject). The ExternalSystem identifier options are the external_system_id, external_system_url or external_system_name value. The path option is simply path. Note, that unlike the get method above, it is not possible to gurantee that only one record will be found per query because a given subject, externalSystem, and path combination may have multiple externalRecords (indicating the subject has multiple records in the given external system for the given path if all 3 pieces of information are provided). As such, the response objects will include, if successful, an external_records element that is an array of external_records. The response object will also include the supplied query parameters. For options that are not supplied, the label will be suffixed with an _ and the value will be 'not_supplied'. A single http request can be used to submit one or more queries. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON If the content type is specified as application/json, then the request body must be of the following form:

```
[
{"subject_id":"id","external_system_id":"id", "path":"value"},
{"subject_org":"ehb_id","subject_org_id":"value"},
{"subject_id":"id","external_system_name":"value"},
{"subject_id":"value","external_system_url":"value"}
]
```

If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"external_system_id": "value",
  "subject_id": "2",
  "path":"value",
  "external_record":[
    {"created": "yyyy-mm-dd hh:mm:ss",
     "external_system_id": "value",
     "id": "value",
     "path": "value",
     "modified": "yyyy-mm-dd hh:mm:ss",
     "record_id": "value",
     "subject_id": "value"},
    {"created": "yyyy-mm-dd hh:mm:ss",
     "external_system_id": "value",
     "id": "value",
     "modified": "yyyy-mm-dd hh:mm:ss",
     "record_id": "value",
     "subject_id": "value"}
]},
{"external_system_": "not_provided",
  "path_":"not_provided",
  "subject_org": "value",
  "subject_org_id" : "value",
  "external_record":[
    {"created": "yyyy-mm-dd hh:mm:ss",
     "external_system_id": "value",
     "id": "value",
     "path": "value"
     "modified": "yyyy-mm-dd hh:mm:ss",
     "record_id": "value",
     "subject_id": "value"}]},
{"external_system_name": "value",
 "subject_id": "value",
 "path_":"not_provided"
 "errors": [{"label":errorcode}]
},
...]
```

1.4.2 Creating ExternalRecords Using POST

The URL for this request is: **ehb-host/api/externalrecord/**

ExternalRecord records are created using a POST request. A single request can be used to create one or more new ExternalRecord records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"subject":"id","external_system":"id","record_id":"value", "path":"value"},
{"subject":"id","external_system":"id","record_id":"value"},
{"subject":"id","external_system":"id","record_id":"value"}]
```

where the "id" for subject and externalSystem must correspond to the integer value ids of an eHB Subject and ExternalSystem record (see error examples below). Note that if the "path" value is not supplied, this entry will be stored as an empty string. If the request body is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"path": "value", "record_id":"value","success": true, "id": "value",
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"record_id": "value","success": false, "errors":[{"label":errorcode}]},
{"record_id": "value","success": false,
"errors":[{"label":errorcode},{"label":errorcode}]
}]
```

Each entry in the response is associated with an record_id in the request. The success field indicates if the new record was created. If it was, as in the first example above, the id field is the id of the newly created ExternalRecord. If not the error field, examples 2 and 3, will provide information as to why. The second example illustrates the case when an attempt is made to create an External-Record for where there is already an ExternalRecord with the given record_id for the given ExternalSystem id. The third example illustrates the case when the given Subject and ExternalSystem ids do not exist.

1.4.3 Modifying ExternalSystem Records Using PUT

The URL for this request is: **ehb-host/api/externalrecord/**

ExternalRecord records can be modified using a PUT request. A single request can be used to modify one or more ExternalRecord records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"id":"value",
  "external_record":{"subject":"id","external_system":"id","record_id":"value","path":"value
  {"id":"value","external_record":{"subject":"id"}},
  {"id":"value",
  "external_record":{"subject":"id","external_system":"id","record_id":"value"}},
  {"id":"value",
  "external_record":{"subject":"id","external_system":"id","record_id":"value"}},
  {"id":"value",
  "external_record":{"subject":"id","external_system":"id","record_id":"value"}}]
```

where the "id" field indicates the record id for the ExternalRecord record to be modified and the "external_record" object contains the updated values for each ExternalRecord field. Note, as seen in the second entry, that it is only necessary to specify the information that is going to be updated, the remaining field values will be unchanged. If the request is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"id": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"id": "value","success": false,
"errors":[
{"label":errorcode}
]},
{"id": "value","success": false
"errors":[
{"label":errorcode},
{"subject":errorcode},
{"id": "value","success": false,
"errors":[
{"label":errorcode}
]}]
```

Each entry in the response is associated with an id in the request and indicates whether the record was successfully modified via the "success" field. If the record was not successfully modified the "errors" field will provide information as to why. Note that as in the case of creating new ExternalRecord records using POST, the subject and externalSystem id values supplied in the request must match the integer value of a Subject and ExternalSystem record in the eHB.

1.4.4 Deleting ExternalRecords Using DELETE

The URL for a GET requests is:
ehb-host/api/externalrecord/id/#/

where # is the ExternalRecord record id in the eHB

Individual ExternalRecord record data can be deleted using a DELETE request. The DELETE requests do not contain a body but must include the 'Accept' value in the header information. Currently, the supported accept types are: */*, application/json and application/xml. If the eHB is able to find the ExternalRecord record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.5 Organization Requests

The following sections detail the requests that can be made on eHB Organization records. The eHB Organization record identifies an organization that may be associated with a Subject record. It has the following fields: name, subject_id_label, created, modified. The subject_id_label is the label that the organization uses for it's unique patient identifier, e.g. MRN. The eHB ensures that organization name is unique and will return an appropriate error message for any action that would result in a duplicate entry.

1.5.1 Obtaining Organization Records Using GET or POST

GET Requests

The URL for a GET requests is:
ehb-host/api/organization/id/#/
where # is the Organization eHB record id

Individual Organization record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the Organization record, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
{
    "created": "yyyy-mm-dd hh:mm:ss",
    "subject_id_label": "value",
    "id": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "name": "value",
}
```

application/xml response:

```
<root>
<id>value</id>
<subject_id_label>value</subject_id_label>
```

```
<name>value</name>
<modified>yyyy-mm-dd hh:mm:ss</modified>
<created>yyyy-mm-dd hh:mm:ss</created>
</root>
```

where id is the Organization record id.

Queries with Post

The URL for this request is: **ehb-host/api/organization/query/**

If the Organization record id is not known, the Organization record information can be obtained using a POST request if the Organization name is known. A single request can be used to obtain one or more Organization records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON If the content type is specified as application/json, then the request body must be of the following form:

```
[{"name":"value"},
{"name":"value"}]
```

If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"name": "value",
  "organization": {
     "created": "yyyy-mm-dd hh:mm:ss",
     "subject_id_label": "value",
     "id": "value",
     "modified": "yyyy-mm-dd hh:mm:ss",
     "name": "value",
   }
},
{"name": "value",
 "errors": [{"label":errorcode}]
}]
```

1.5.2 Creating Organization Records Using POST

The URL for this request is: **ehb-host/api/organization/**

Organization records are created using a POST request. A single request can be used to create one or more new Organization records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"name":"value","subject_id_label":"value"},
{"name":"value","subject_id_label":"value"}
]
```

If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value","name": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"name": "value","success": false,
"errors": [{"label":errorcode}]
}]
```

where id is the ExternalSystem record id if successfully created.

Each entry in the response is associated with an name in the request and indicates whether the record was successfully create via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.5.3 Modifying Organization Records Using PUT

The URL for this request is: **ehb-host/api/organization/**

Organization records can be modified using a PUT request. A single request can be used to modify one or more Organization records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formated in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"id":"value","organization":{"name":"value","subject_id_label":"value"}},
{"id":"value", "organization":{"subject_id_label":"value"}}]
```

where the "id" field indicates the eHB record id for the Organization record to be modified and the "organization" object contains the updated values for each Organization field. Note, as seen in the second entry, that it is only necessary to specify the information that is going to be updated, the remaining field values will be unchanged. If the request is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id": "value","success": true,
"created": "yyyy-mm-dd hh:mm:ss", "modified": "yyyy-mm-dd hh:mm:ss"},
{"id": "value","success": false,
"errors":[{"label":errorcode}]},
{"id": "value","success": false,
"errors":[{"label":errorcode}],
}]
```

Each entry in the response is associated with an id in the request and indicates whether the record was successfully modified via the "success" field. If the record was not successfully modified the "errors" field will provide information as to why.

1.5.4 Deleting Organization Records Using DELETE

The URL for a GET requests is:
ehb-host/api/organization/id/#/
where # is the Organization record id

Individual Organization record data can be deleted using a DELETE request. The DELETE requests do not contain a body but must include the 'Accept' value in the header information. Currently, the supported accept types are: */*, application/json and application/xml. If the eHB is able to find the ExternalSystem record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.6 Group Requests

The following sections detail the requests that can be made on eHB Group records. The eHB Group record is a used to organize eHB Subject and / or ExternalRecord records. A group has the following fields: ehb_key, client_key, isLocking, name, and description. The eHB ensures that name is unique and will return an error message for any action that would result in a duplicate entry. The ehb_key is a unique key generated by the eHB. It's length is governed by an eHB setting, the default is 15 characters selected from uppercase letters A-Z and digits 0-9. The eHB key is returned to the client when a new Group is created. The client_key is provided by the client. The client_key is not sent in ANY response provided by the eHB. It serves as a password for controlling groups.

1.6.1 Obtaining Group Records Using GET

GET Requests

The URLs for GET requests are:
ehb-host/api/group/?id=value
where value is the Group record primary key

ehb-host/api/group/?name=value
where value is the Group record name

Individual Group record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml. If the eHB is unable to find the Group record, a 416 REQUESTED RANGE NOT SATISFIABLE status response is returned. Otherwise the response will be of the form:
application/json response:

```
{
    "created": "yyyy-mm-dd hh:mm:ss",
    "description": "value",
    "id": "value",
    "is_locking": value,
    "modified": "yyyy-mm-dd hh:mm:ss",
    "name": "value",
    "ehb_key": "value",
}
```

application/xml response:

```
<root>
    <ehb_key>value</ehb_key>
    <description>value</description>
    <created>yyyy-mm-dd hh:mm:ss</created>
    <modified>yyyy-mm-dd hh:mm:ss</modified>
    <is_locking>value</is_locking>
    <id>value</id>
    <name>value</name>
</root>
```

where id is the eHB Group record id.

1.6.2 Creating Group Records Using POST

The URL for this request is:
ehb-host/api/group/

Group records are created using a POST request. A single request can be used to create one or more new Group records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formatted in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"name":"value","client_key":"value","is_locking":value,"description":"value"},
{"name":"value","client_key":"value","is_locking":value,"description":"value"}
]
```

where is_locking should be either true or false.
If it is not of this form, most likely a 500 Server Error response will be returned.
If the request form is correct, the response will be a JSON array of the form

```
[{"created": "yyyy-mm-dd hh:mm:ss", "ehb_key": "value", "id": "value", "modified": "yyyy-mm-
"value", "success": true },
{"errors": [{"label": error_code}],"name": "value","success": false}]
```

where id is the eHB Subject record id if successfully created.

Each entry in the response is associated with a name in the request and indicates whether the record was successfully create via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.6.3 Modifying Group Records Using PUT

The URL for this request is:
ehb-host/api/group/

Group records can be modified using a PUT request. A single request can be used to modify one or more Group records. The request must include the 'Content-Type' value in the header information. It is expected that the request body will be formatted in the manner specified by the content type. The response format will match the content type. Currently, the supported content types are: 'application/json'.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[{"id":"value",
  "group": {"current_client_key":"value",
            "name":"value","client_key":"value","is_locking":value,"description":"value"}
 },
{"id":"value",
  "group": {"current_client_key":"value",
            "name":"value","client_key":"value","is_locking":value,"description":"value"}
 }
]
```

where is_locking should be either true or false.

Note that it is necessary to supply the current value of the client_key in the field current_client_key regardless of wether or not the client_key value is being updated. If this field is not supplied a 400 Bad Request response will be returned. If the field is supplied, but is not the correct value, a 401 Unauthorized response will be returned.
If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"created":"yyyy-mm-dd hh:mm:ss","id": "value","modified":"yyyy-mm-dd hh:mm:ss","success":
{"errors": [{"label": error_code}],"id": "value","success": false}]
```

where id is the eHB Subject record id if successfully created.

Each entry in the response is associated with a name in the request and indicates whether the record was successfully create via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.6.4 Deleting Group Records N/A

The URLs for DELETE requests are:
ehb-host/api/group/?id=value
where value is the Group record primary key
ehb-host/api/group/?name=value
where value is the Group record name

Individual Group records can be removed using a DELETE request. The DELETE requests do not contain a body. It is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is not included or is incorrect a 403 Forbidden response will be returned. If the eHB is able to find the Group record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.7 Subject Groups

As discussed in section 1.6, it is possible to create a Group record in the eHB. Once a group is created, subject records can be added to it. A SubjectGroup is the relation of a Group to one or more Subject records. This section details the methods supported to for SubjectGroup records.

GET Requests

The URL for GET a request is:
ehb-host/api/group/id/#/subjects/
where # is the Group record primary key

Individual SubjectGroup record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. Additionally, if the Group associated with the SubjectGroup has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. The normal response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the Group or SubjectGroup record, a 404 NOT FOUND status response is returned. Otherwise the response will be an array of subject information of the form:
application/json response:

```
[{
    "created": "yyyy-mm-dd hh:mm:ss",
    "dob": "yyyy-mm-dd",
    "first_name": "value",
    "id": "value",
    "last_name": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "organization_id": "value",
    "organization_subject_id": "value"
}, ...
]
```

application/xml response:

```
<root>
    <subject>
        <first_name>value</first_name>
        <last_name>value</last_name>
        <created>yyyy-mm-dd hh:mm:ss</created>
        <dob>yyyy-mm-dd</dob>
        <modified>yyyy-mm-dd hh:mm:ss</modified>
        <organization_id>value</organization_id>
        <organization_subject_id>value</organization_subject_id>
        <id>46</id>
    </subject>
    <subject>
       ...
    </subject>
</root>
```

1.7.1 Adding Subjects to a SubjectGroup Using POST

The URL for this request is:

ehb-host/api/group/id/#/subjects/

Subjects are added to a Subject Group using a POST request. A single request can be used to add one or more Subjects to a SubjectGroup. The request must include the 'Content-Type' value in the header information. Additionally, if the Group associated with the SubjectGroup has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. The normal response format will match the accept type. Currently, the supported content types are: application/json.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[x,y,z, ...]
```

where each value in the JSON array is the primary key integer value of the subject to be added to the Subject Group.
If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id":value,"success": true},
{"errors":error_code,"id":value,"success": false}]
```

where id is the eHB Subject record id.

Each entry in the response is associated with a subject id in the request and indicates whether the subject was successfully added to the SubjectGroup via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.7.2 Removing Subjects From a SubjectGroup Using DELETE

The URLs for removing Subjects from a SubjectGroup are:
ehb-host/api/group/id/#1/subjects/id/#2/
where #1 is the Group record id and #2 is the Subject record id

Individual Subjects can be removed from a SubjectGroup using a DELETE request. The DELETE requests do not contain a body. If the Group associated with the SubjectGroup has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. If the eHB is able to find the Group and Subject record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.8 ExternalRecord Groups

As discussed in section 1.6, it is possible to create a Group record in the eHB. Once a group is created, ExternalRecord records can be added to it. An ExternalRecordGroup is the relation of a Group to one or more ExternalRecord records. This section details the methods supported to for ExternalRecord-Group records.

GET Requests

The URL for GET a request is:
ehb-host/api/group/id/#/records/
where # is the Group record primary key

Individual ExternalRecordGroup record data can be obtained using a GET request. The GET request does not contain a body but it must include the 'Accept' value in the header information. Additionally, if the Group associated with the ExternalRecord has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. The normal response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the Group or ExternalRecordGroup record, a 404 NOT FOUND status response is returned. Otherwise the response will be an array of subject information of the form:
application/json response:

```
[{
    "created": "yyyy-mm-dd hh:mm:ss",
    "external_system_id": "value",
    "id": "1",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "path": "value",
    "record_id": "value",
    "subject_id": "value"
}, ...
]
```

application/xml response:

```
<root>
    <external_record>
        <created>yyyy-mm-dd hh:mm:ss</created>
        <subject_id>value</subject_id>
        <external_system_id>value</external_system_id>
        <modified>yyyy-mm-dd hh:mm:ss</modified>
        <record_id>value</record_id>
```

```
        <path>value</path>
        <id>value</id>
    </external_record>
    <external_record>
       ...
    </external_record>
</root>
```

1.8.1 Adding ExternalRecords to a ExternalRecord Using POST

The URL for this request is:
ehb-host/api/group/id/#/records/

ExternalRecords are added to a ExternalRecordGroup using a POST request. A single request can be used to add one or more ExternalRecords to a ExternalRecordGroup. The request must include the 'Content-Type' value in the header information. Additionally, if the Group associated with the ExternalRecordGroup has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. The normal response format will match the accept type. Currently, the supported content types are: application/json.

JSON

If the content type is specified as application/json, then the request body must be of the following form:

```
[x,y,z, ...]
```

where each value in the JSON array is the primary key integer value of the ExternalRecord to be added to the ExternalRecordGroup.
If it is not of this form, most likely a 500 Server Error response will be returned. If the request form is correct, the response will be a JSON array of the form

```
[{"id":value,"success": true},
{"errors":error_code,"id":value,"success": false}]
```

where id is the eHB ExternalRecord record id.

Each entry in the response is associated with a ExternalRecord id in the request and indicates whether the ExternalRecord was successfully added to the ExternalRecordGroup via the "success" field. If the record was not successfully created the "errors" field will provide information as to why.

1.8.2 Removing ExternalRecord From a ExternalRecordGroup Using DELETE

The URLs for removing ExternalRecord from a ExternalRecordGroup are:

ehb-host/api/group/id/#1/records/id/#2/
where #1 is the Group record id and #2 is the ExternalRecord record id

Individual ExternalRecords can be removed from a ExternalRecordGroup using a DELETE request. The DELETE requests do not contain a body. If the Group associated with the ExternalRecordGroup has is_locking set to true, is also necessary to include a custom header, *GROUP_CLIENT_KEY* whose value should be set to the value of the Group's client_key. If this value is required but not included or is incorrect a 403 Forbidden response will be returned. If the eHB is able to find the Group and ExternalRecord record, the response to a DELETE request will be an 200 OK response, otherwise a 404 NOT FOUND response is sent.

1.9 Cross Reference Queries

The following sections details the requests that can be made across model types on the ehB.

1.9.1 Subjects by ExternalSystem Using GET

The URLs for this request is:
ehb-host/api/externalsystem/id/#1/subjects/
ehb-host/api/externalsystem/id/#1/organization/#2/subjects/
where # is the ExternalSystem record id and #2 is eHB Organization record id. This request will return all of the Subject data for all of the Subject records that have an ExternalRecord for the ExternalSystem indicated by id=#1. If the organization id is provided (#2 in second URL), then only Subjects belonging to that organization will be inlcuded. Requests are made using the GET verb. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the ExternalSystem, Organization, or any corresponding Subject records, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
[{
    "created": "yyyy-mm-dd hh:mm:ss",
    "dob": "yyyy-mm-dd",
    "first_name": "value",
    "id": "ehb_id",
    "last_name": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "organization_id": "ehb_id",
    "organization_subject_id": "value",
}, ...
```

```
]
```

application/xml response:

```
<root>
    <first_name>value</first_name>
    <last_name>value</last_name>
    <created>yyyy-mm-dd hh:mm:ss</created>
    <dob>yyyy-mm-dd</dob>
    <modified>yyyy-mm-dd hh:mm:dd</modified>
    <organization_id>ehb_id</organization_id>
    <organization_subject_id>value</organization_subject_id>
    <id>ehb_id</id>
    ...
</root>
```

where id is the Subject record id.

1.9.2 ExternalRecords by ExternalSystem Using GET

The URL for this request is:
ehb-host/api/externalsystem/id/#1/records/
ehb-host/api/externalsystem/id/#1/organization/#2/records/
where #1 is the ExternalSystem record id and #2 is eHB Organization record id. This request will return all of the ExternalRecord data for all of the ExternalRecord records that are associated with the ExternalSystem indicated by id=#1. If the organization id is provided (#2 in second URL), then only ExternalRecords whose Subject belongs to that organization will be inlcuded. Requests are made using the GET verb. The GET request does not contain a body but it must include the 'Accept' value in the header information. The response format will match the accept type. Currently, the supported accept types are: application/json and application/xml.
If the eHB is unable to find the ExternalSystem, Organization, or any corresponding ExternalRecords records, a 404 NOT FOUND response is returned. Otherwise the response will be of the form:
application/json response:

```
[{
    "created": "yyyy-mm-dd hh:mm:ss",
    "external_system_id": "value",
    "id": "value",
    "modified": "yyyy-mm-dd hh:mm:ss",
    "path": "value",
    "record_id": "value",
    "subject_id": "value",
}, ...
]
```

application/xml response:

```
<root>
    <created>yyyy-mm-dd hh:mm:ss</created>
    <subject_id>value</subject_id>
    <external_system_id>value</external_system_id>
    <modified>yyyy-mm-dd hh:mm:ss</modified>
    <record_id>value</record_id>
    <path>value</path>
    <id>value</id>
    ...
</root>
```

where id is the Subject record id.

1.10 Error Codes

Response bodies may include an error element. The error element will include error items of the form key:value, where key is a String label and value is an integer value from the error codes below. For example, if an attempt was made to update a Subject record with id=N provided as input, an error element of "id":1 would indicate that the no Subject record exists with the specified id. The possible response error codes are

ERROR_UNKNOWN = 0 : indicates an unknown error occurred

ERROR_RECORD_ID_NOT_FOUND = 1 : indicates record could not be found with given information

ERROR_FIELD_REQUIRED = 2 : indicates a value must be provide for "key"

ERROR_SUBJECT_ORG_ID_EXISTS = 3 : indicates an attempt was made to create a new Subject record or modify an existing one to have the same organization_subject_id value as another Subject record

ERROR_INVALID_DATE_FORMAT = 4 : indicates the provided date was not in YYYY-MM-DD format

ERROR_EXTERNAL_SYSTEM_NAME_EXISTS = 5 : indicates an attempt was made to create a new ExternalSystem record or modify an existing one to have the same name as another ExternalSystem record

ERROR_RECORD_ID_ALREADY_IN_EXTERNAL_SYSTEM = 6 : indicates an attempt was made to create an ExternalRecord or modify an exsiting one to have the same path and record_id value as another ExternalRecord record

ERROR_INVALID_CHOICE = 7 : indicates an invalid choice was provided, e.g. attempting to create an ExternalRecord for a Subject that does not exist

ERROR_INVALID_QUERY = 8 : indicates the information provided to a query was incomplete or inaccurate

ERROR_NO_RECORD_FOUND_FOR_QUERY = 9 : indicates no record could be found for the query using the given information

ERROR_EXTERNAL_SYSTEM_URL_EXISTS = 10 : indicates an attempt was made to create a new ExternalSystem record or modify an existing one to have the same URL as another ExternalSystem record

ERROR_ORGANIZATION_NAME_EXISTS = 11 : indicates an attempt was made to create a new Organization record or modify an existing one to have the same name as another Organization record

ERROR_GROUP_NAME_EXISTS = 12 : indicates an attempt was made to create a new Group record or modify an existing one to have the same name as another Group record

ERROR_ID_NOT_FOUND = 13 : indicates that supplied id for to locate a given resource does not exist (what else would it mean?)

Bibliography