API

RESTHeart API serves 4 types of resources:

Root	at URL /
DB	at URLs /<dbName>
Collection	at URLs /<dbName/>/<collectionName>
Document	at URLs /<dbName>/<collectionName/>/<documentId>

By using mongo-mounts RESTHeart can expose different resources at different URLs.

In this section we assume the following (default) configuration:

```
mongo-mounts:
 - what: "*"
   where: /
```

i.e. that all resources are being served from the root URL.

In this section, the examples make use of the brilliant **httpie** tool. If you don't have it already go httpie.org (http://httpie.org) and grab it NOW!

Resources are represented in the HAL+json (http://stateless.co/hal_specification.html) hypermedia type.

In HAL, a resource is represented as:

- a set of properties
- a set of embedded resources representations
- a set of links

- Resource ids starting with _ are reserved.
- Property keys starting with _ are reserved with the exception of "_id".

You cannot create a resource with a reserved id (if you try you'll get back `BAD_REQUEST` response code).

If you try to send a resource with a property having a reserved key, this will be filtered out. This includes `_embedded` and `_links`.

The Root resource

The root resource is the entry point of the data API.

The root resource is a ready only resource (does not have properties on its own).

`GET /`

returns the list of databases as embedded resources.

Paging supported.

(https://github.com/softinstigate/restheart)

Special Properties

Intro (./overvie
Configuration (./configuratio
API
root
db
collection
indexes
document

the following properties are automatically managed by RESTHeart. Their keys start with _ thus are reserved (cannot be modified)

key	description	when returned
_size	how many dbs exist	always
_returned	how many embedded dbs were returned	always
_total_pages	how many pages are available (refer to page query parameter)	always

Query parameters

parameter	description	default value
pagesize	used for paging: the maximum number of _embedded resources to return per page	100
page	used for paging: the page to return	1

Return codes

code	when	details
200 OK	the request completed successfully	
400 BAD REQUEST	a query parameter has an invalid value (e.g. pagesize is negative or not a number)	a response document is sent with detailed information about the error
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to request this resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.

Examples

> Notice, this root resource:
>
> - has 3 properties: `_size` , `_returned` , `_total_pages`
> - has 1 embedded resource representation of the DB mydb
> - has some links, including self, next, rh:paging and curies

```
$ http GET 127.0.0.1:8080/
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Length: 275
Content-Type: application/hal+json
Date: Thu, 23 Oct 2014 10:14:21 GMT
```

```
{
"_embedded": {
    "rh:db": [
        {
            "_id": "mydb",
            "_links": {
                "self": {
                    "href": "/mydb"
                }
            }
        }
    ]
},
"_links": {
    "curies": [
        {
            "href": "http://www.restheart.org/docs/v0.9/#api-root-{rel}",
            "name": "rh",
            "templated": true
        }
    ],
    "next": {
        "href": "/?page=2&pagesize=100"
    },
    "rh:paging": {
        "href": "/{?page}{&pagesize}",
        "templated": true
    },
    "self": {
        "href": "/"
    }
},
"_returned": 1,
"_size": 1,
"_total_pages": 1
}
```

The DB resources

The DB resources allow you to manage your MongoDB databases.
The operations described in this section are part of the data model API, i.e. they are used at development time to define the data structure.

> Each db has its own properties. This means that you can attach any JSON document to your dbs. For instance, you can use this feature to implement a custom Access Manager that uses some DB ACL properties to enforce the security policy.

`GET /<dbname>`

returns the HAL representation of the db <dbname> with its collections as embedded resources

> Paging supported.

Special Properties

the following properties are automatically managed by RESTHeart. Their keys start with _ thus are reserved (cannot be modified)

key	description	when returned
_id	the db name	always
_created_on	creation timestamp in ISO-8601 format	always
_lastupdated_on	last update timestamp in ISO-8601 format	always
_etag	the entity tag	always
_db-props-cached	true if the returned properties come from the local cache	always

_size	how many collections exist	always
_returned	how many embedded collections were returned	always
_total_pages	how many pages are available (refer to page query parameter)	always

Query parameters

parameter	description	default value
pagesize	used for paging: the maximum number of _embedded resources to return per page	100
page	used for paging: the page to return	1

Return codes

code	when	details
200 OK	the request completed successfully	
400 BAD REQUEST	a query parameter has an invalid value (e.g. pagesize is negative or not a number)	a response document is sent with detailed information about the error
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to request this resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.

Examples

```
example here
```

`PUT /<dbname>`

creates the db <dbname> or updates its properties

> PUT has upsert semantic: if the DB <dbname&get exists it will be updated, otherwise it will be created.

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been updated	
201 CREATED	the resource has been created	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _) or optional _id property does not match the resource id in URL	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`
If-Match	no (yes in case of updating)	request	the ETag must be provided via this header.

Examples

```
example here
```

`PATCH /<dbname>`

partially updates the properties of the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been partially updated	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _)	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`
If-Match	yes	request	the ETag must be provided via this header.

Examples

```
example here
```

`DELETE /<dbname>`

deletes the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details

204 NO CONTENT	the resource has been deleted	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	description		
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
If-Match	yes	request	in case of update, the ETag must be provided via this header.

Examples

```
example here
```

The Collection resources

The Collection resources allow you to manage your MongoDB collections.
The PUT, PATCH and DELETE operations are part of the data model API, i.e. they are used at development time to define the data structure.
THE GET and POST are part of the data API, i.e. they are used for retrieving and creating/updating data.

> Each collection has its own properties. This means that you can attach any JSON document to your collections.
> For instance, you can use this feature to implement a custom Access Manager that uses some collection ACL properties to enforce the security policy.
> Some collection properties are meta-data and have a special meaning. For instance the property `rels` allow to define the relationship between collections.

Collection meta-data

Meta-data are special properties used by RESTHeart to control the representation of the resources.

Meta-data properties

properties	description	type

rels	the `rels` properties allow to define the relationships between documents. When defined, the representation of the collection's documents auto-magically includes the links to the referenced documents between the HAL `_link` property	array of rel objects

rel object

a relationship is defined using the `rel` object. The format is:

{ "rel": <relid>, "type": <type>, "role:" <role>, "target-db": <dname>, "target-coll": <collname>, "ref-field": <reffieldname>}

property	description	mandatory
rel	the name of the relationship	yes
type	the type of relationship: ONE_TO_ONE, MANY_TO_ONE, ONE_TO_MANY or MANY_TO_MANY	yes
role	the role of the relationship: OWNING or INVERSE. This says where the reference field is. OWNING means it is in this document, INVERSE means that it is on the target document.	yes
target-db	the db name of the referenced document(s)	no. if omitted the target db is supposed to be the current one
target-coll	the collection name of the referenced document(s)	yes
ref-field	the name of the field storing the id of the referenced document(s). either the id string (in case the multiplicity of other side of the relationship is ONE) or an array of id strings (in case the multiplicity of other side of the relationship is MANY)	yes

`GET /<dbname>/<collname>`

returns the HAL representation of the collection <collname> of db <dbname> with its documents as embedded resources

> Paging supported.

> Sorting supported.

> Filtering supported.

> ETag based caching supported (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)).

> Eager db cursors preallocation supported. Dramatically improves read performances on big collections.

Special Properties

the following properties are automatically managed by RESTHeart. Their keys start with _ thus are reserved (cannot be modified)

key	description	when returned
_id	the collection name	always
_created_on	creation timestamp in ISO-8601 format	always
_lastupdated_on	last update timestamp in ISO-8601 format	always
_etag	the entity tag	always

_collection-props-cached	true if the returned properties come from the local cache	always
_size	how many documents exist in this collection	only if count query parameter was specified
_returned	how many embedded documents were returned	always
_total_pages	how many pages are available (refer to page query parameter)	only if count query parameter was specified

Query parameters

parameter	description	default value
count	if this parameter is in the query string, return the _size and _total_pages properties. Note: counting the elements of a collection slows down the request.	don't count
pagesize	used for paging: the maximum number of _embedded resources to return per page	100
page	used for paging: the page to return	1
sort_by	used for sorting: the field to sort (prepend with - for descending sorting). multiple sorty_by parameters can be specified.	-created_on
filter	used for filtering: a JSON MongoDB filtering expression expression to apply. e.g. filter={"name":{"$regex":".*k"}}. . multiple filter parameters can be specified.	<none>
eager	specify the eager db cursors preallocation policy: valid values are linear, random, none.	linear

Return codes

code	when	details
200 OK	the request completed successfully	
400 BAD REQUEST	a query parameter has an invalid value (e.g. page size is negative or not a number)	a response document is sent with detailed information about the error
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to request this resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.

Examples

```
example here
```

`PUT /<dbname>/<collname>`

creates the collection <collname> of the db <dbname> or updates its properties

> PUT has upsert semantic: if the collection exists it will be updated, otherwise it will be created.

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been updated	
201 CREATED	the resource has been created	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _) or optional _id property does not match the resource id in URL	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`
If-Match	no (yes in case of updating)	request	the ETag must be provided via this header.

Examples

```
example here
```

`POST /<dbname>/<collname>`

creates or updates a document in the collection <collname> of the db <dbname>

> POST has upsert semantic: if the document exists it will be updated, otherwise it will be created.
> The document id can be specified with the optional property _id; otherwise an _id will be created

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been updated	
201 CREATED	the resource has been created	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _) or optional _id property does not match the resource id in URL	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	in case of error a response document is sent together with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`

If-Match	no (yes if updating)	request	in case of update, the ETag must be provided via this header.
Location	n.a.	response	if a new document got created, the value of this header is the new resource URL.

Examples

```
example here
```

`PATCH /<collname>/<dbname>`

partially updates the properties of the collection <collname> of the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been partially updated	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _)	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`

If-Match	yes	request	the ETag must be provided via this header.

Examples

```
example here
```

`DELETE /<collname>/<dbname>`

deletes the collection <collname> of the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
204 NO CONTENT	the resource has been deleted	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	description		
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
If-Match	yes	request	in case of update, the ETag must be provided via this header.

Examples

```
example here
```

The Indexes and Index resources

The Indexes resources allow you to manage your MongoDB indexes.
The operations described in this section are part of the data model API, i.e. they are used at development time to define the data structure.

`GET /<dbname>/<collname>/_indexes`

returns the indexes of the collection /<collname> of the db <dbname>

Examples

```
example here
```

`PUT /<dbname>/<collname>/_indexes/<indexname>`

creates or updates an index on the collection /<collname> of the db <dbname>

> PUT index has insert semantic: if the index exists an error is returned, otherwise it will be created.

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
201 CREATED	the resource has been created	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
406 NOT_ACCEPTABLE	the request body is not a valid json or HAL+json document or the id is reserved (e.g. starts with _)	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter

Examples

```
example here
```

`DELETE /<dbname>/<collname>/_indexes/<indexname>`

deletes the index on the collection /<collname> of the db <dbname>

Return codes

code	when	details
204 NO CONTENT	the resource has been deleted	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.

Examples

```
example here
```

The Document resources

The Document resources allow you to manage your MongoDB documents.
The operations described in this section are part of the data API, i.e. they are used for CRUD operations on data.

`GET /<dbname>/<collname>/<docid>`

returns the HAL representation of the document with id <docid> in the collection <collname> of the db <dbname>

Return codes

code	when	details
200 OK	the request completed successfully	
304 NOT MODIFIED	the request completed successfully and the client side cached entity (identified by the ETag) is valid.	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to request this	

	resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
If-None-Match	no	request	to allow ETag based caching.

Examples

```
example here
```

`PUT /<dbname>/<collname>/<docid>`

creates or updates the document with id <docid> in the collection /<collname> of the db <dbname>

> PUT has upsert semantic: if the resource exists it will be updated, otherwise it will be created.

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been updated	
201 CREATED	the resource has been created	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _) or optional _id property does not match the resource id in URL	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to

code	when	details
		match the its current _etag property.
415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`
If-Match	no (yes in case of updating)	request	the ETag must be provided via this header.

Examples

```
example here
```

`PATCH /<dbname>/<collname>/<docid>`

partially updates the the document with id <docid> in the collection /<collname> of the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
200 OK	the resource has been partially updated	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
406 NOT_ACCEPTABLE	the request body is not a valid JSON or HAL+JSON document or the id is reserved (e.g. starts with _)	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.

415 UNSUPPORTED_MEDIA_TYPE	Content-Type request header must be either application/hal+json or application/json	
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	mandatory	type	description
Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
Content-Type	yes	request	either `application/json` or `application/hal+json`
If-Match	yes	request	the ETag must be provided via this header.

Examples

```
example here
```

`DELETE /<dbname>/<collname>/<docid>`

deletes the the document with id <docid> in the collection /<collname> of the db <dbname>

> Optimistic concurrency control via ETag (entity tag (http://en.wikipedia.org/wiki/HTTP_ETag)) is required.

Return codes

code	when	details
204 NO CONTENT	the resource has been deleted	
401 UNAUTHORIZED	the authentication failed or the authenticated user is not authorized to upsert the resource	
403 FORBIDDEN	no authentication credentials sent	this is only sent for requests with No-Auth-Challenge header or noauthchallenge query parameter
404 NOT FOUND	if the resource does not exist	
409 CONFLICT	in case of update, the If-Match header with the ETag value was not provided	in order to avoid ghost writes, updating a resource requires the ETag to be provided via the If-Match header.
412 PRECONDITION_FAILED	in case of update, the If-Match header with the ETag value does not match	in order to avoid ghost writes, updating a resource requires the ETag header value to match the its current _etag property.
500 INTERNAL SERVER ERROR	an error occurred (e.g. MongoDB is down)	a response document is sent with detailed information about the error, including the complete stack trace of the exception as an embedded resource

Headers

header	description

Authorization	no	request	basic authentication required according to the security policy.
Accept-Encoding	no (yes in case configuration `force-gzip-encoding` is true)	request	gzip encoding can be requested
No-Auth-Challenge	no	request	requests with this header will not be challenged for basic authentication (i.e. will not get the WWW-Authenticate response header). this can also be achieved passing the noauthchallenge query parameter.
If-Match	yes	request	in case of update, the ETag must be provided via this header.

Examples

```
example here
```

Follow us on Twitter (https://twitter.com/softinstigate)