

M.A.L.E.N.A.H. API

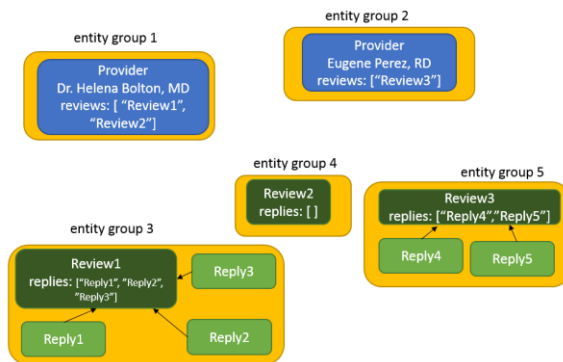
The MALENAH API was created using Google App Engine (Python) and hosted at:

<http://malenah-api.appspot.com>

Brief Recap of Background and Schema

MALENAH stands for **M**obile **A**lly for **L**GBT people, **E**nhancing through a **N**etwork, **A**ccess to **H**ealthcare. The MALENAH API stores information about healthcare providers as Provider entities. Each Provider has associated Specialization entities (e.g. chiropractic, emergency medicine, geriatrics, ophthalmology, endocrinology). Users can write reviews about a provider, which are stored as Review entities. Replies, which are responses to reviews are stored as Reply entities. A Review entity is always the root entity of an entity group containing Reply entities. In other words, Reply entities belong to a parent Review entity's entity group. A Provider entity is the sole (root) entity of its entity group, and a Specialization entity is the root entity of its entity group.

Relationships (Associations) between Entities



The diagram to the left is a schema of the database to help understand the URL structure of the API. Although a Review entity is always in its own entity group, a Review entity must be associated with a Provider entity because reviews are written about healthcare providers, and a Reply entity must be associated with a Review entity, so a Reply entity is always in an entity group with a Review entity as its parent. Reply keys are stored in a repeated property called replies within Review entities. This explains the provider/providerID/review/reviewID/reply/replyID structure. To minimize contention during queries and writes, Review entities are not stored in the same entity group as their associated Provider entity.

API URL Structure for Accessing Resources

The MALENAH API supports HTTP GET, POST, PUT and DELETE, and exposes the following URI's for public consumption (ending forward slashes are optional):

To access Provider entities (and its associated Review entities, and Review entities' associated Reply entities), use the following URL's:

<http://malenah-api.appspot.com/provider>
[http://malenah-api.appspot.com/provider/<providerID>/](http://malenah-api.appspot.com/provider/<providerID>)
<http://malenah-api.appspot.com/provider/<providerID>/review/>
<http://malenah-api.appspot.com/provider/<providerID>/review/<reviewID>/>
<http://malenah-api.appspot.com/provider/<providerID>/review/<reviewID>/reply/>
<http://malenah-api.appspot.com/provider/<providerID>/review/<reviewID>/reply/<replyID>/>

To access Specialization entities:

<http://malenah-api.appspot.com/specialization>
<http://malenah-api.appspot.com/specialization/<specializationID>/>

To access Review entities (and its associated Reply entities):

<http://malenah-api.appspot.com/review>
<http://malenah-api.appspot.com/review/<reviewID>/>
<http://malenah-api.appspot.com/review/<reviewID>/reply/>
<http://malenah-api.appspot.com/review/<reviewID>/reply/<replyID>/>

To access Reply entities:

<http://malenah-api.appspot.com/reply>
<http://malenah-api.appspot.com/reply/<replyID>/>

This screen capture of the code shows regular expressions handling the routes in Google App Engine (Python). These regular expressions determine which handler the request will be routed to. Note that the API evaluates numeric inputs for pid (provider ID), revid (review ID), and repid (reply ID).

```
routes = [
    webapp2.Route(r'<:specialization></sid:[0-9]+></?>', handler=S.SpecializationHandler, name='specialization'),
    webapp2.Route(r'<:specialization></?>', handler=S.SpecializationHandler, name='specialization-list'),
    webapp2.Route(r'<:provider></pid:[0-9]+></:review></revid:[0-9]+></:reply></repid:[0-9]+></?>', handler=r.ReplyHandler, name='provider-review-reply'),
    webapp2.Route(r'<:provider></pid:[0-9]+></:review></revid:[0-9]+></:reply></?>', handler=r.ReplyHandler, name='provider-review-reply'),
    webapp2.Route(r'<:provider></pid:[0-9]+></:review></revid:[0-9]+></?>', handler=R.ReviewHandler, name='provider-review'),
    webapp2.Route(r'<:provider></pid:[0-9]+></:review></?>', handler=R.ReviewHandler, name='provider-review-list'),
    webapp2.Route(r'<:provider></pid:[0-9]+></?>', handler=P.ProviderHandler, name='provider'),
    webapp2.Route(r'<:provider></?>', handler=P.ProviderHandler, name='provider-list'),
    webapp2.Route(r'<:review></revid:[0-9]+></:reply></repid:[0-9]+></?>', handler=r.ReplyHandler, name='reply'),
    webapp2.Route(r'<:review></revid:[0-9]+></:reply></?>', handler=r.ReplyHandler, name='reply-list'),
    webapp2.Route(r'<:review></revid:[0-9]+></?>', handler=R.ReviewHandler, name='review'),
    webapp2.Route(r'<:review></?>', handler=R.ReviewHandler, name='review-list'),
    webapp2.Route(r'<:reply></repid:[0-9]+></?>', handler=r.ReplyHandler, name='reply'),
    webapp2.Route(r'<:reply></?>', handler=r.ReplyHandler, name='reply-list'),
    webapp2.Route(r'/', handler=P.ProviderHandler, name='provider-list'),
]

application = webapp2.WSGIApplication(routes, debug=True, config=config)
```

RESTful Constraints

The MALENAH API meets some of the RESTful constraints, but does not meet others. Note that some aspects of a given constraint were met while some aspects of a constraint were not.

The following addresses some of the REST requirements met:

- client-server: client remains separate from the server. In the case of my API, the client (user or user application) can use my API without needing to store data as the server takes care of data storage. On the other hand, my API is not involved in issues regarding user interface. [2][3]
- stateless: The client (e.g. browser or chrome plugin Postman) sends HTTP requests to my API. These HTTP requests contain information in the body and header that communicates state to the server. The server also sends back information in the HTTP request status, header and body, communicating state to client. In this respect, the server does not maintain state about a client.
- layered system: In the case of the MALENAH API, the client connects directly to my API. There are no other opaque layers. However, the client application is unable to distinguish whether it is connected directly to my API or through some intermediary, so this RESTful constraint is met. [4]
- uniform interface:
 - resource identifiers: Each resource (in my web app, these are Provider, Review and Reply entities) are identified by URI's. [3]
 - representations: The resources are represented as JSON objects.
 - self-descriptive messages: The MIME type (JSON) is specified. Based on the JSON representation of the resources, users and user applications can manipulate the resource based on the information provided by these representations. For example, every Provider, Review and Reply resource includes its key in the JSON representation. The Provider entity's key can be used to get, post, put and delete Provider entities, and can be used in get, post, put and delete operations for Review and Reply resources as well, [3]
 - HATEOAS: The header and body of the HTTP messages exchanged between the API and the client (e.g. browser) communicate state. The state is handled by hyperlinks. In my application, these hyperlinks are:
 - /provider
 - /provider/providerID
 - /provider/providerID/review
 - /provider/providerID/review/reviewID
 - /provider/providerID/review/reviewID/reply
 - /provider/providerID/review/reviewID/reply/replyID
 - /review
 - /review/reviewID
 - /review/reviewID/reply
 - /review/reviewID/reply/replyID
 - /reply
 - /reply/replyID

Some REST requirements that were not met include the following:

- cacheable: The server does not send caching information in a Cache-Control header back to the client, so there is a possibility that the client may store stale data. [4]
- uniform interface:
 - self-descriptive messages: The cacheability of the resource is not specified in a Cache-Control header.
 - HATEOAS: While the resources themselves do not provide explicit links to other resources, it is not difficult to determine how to find other resources. [3]

Changes to the Schema from Last Week

In addition to the Provider, Review and Reply entity, I added a Specialization entity instead of storing specializations as string properties in Provider entities. Having Provider entities associated with Specialization entities improves the consistency of naming of specializations, so that a single specialization, like Endocrinology, will not be stored twice as endocrinology and Endocrinology, or any other correctly or incorrectly-spelled variants. No major changes to the schema were required. However, in order to complete the assignment in a timely fashion, I did not include user authentication or authorization, and thus did not include the User entity that was originally part of the schema submitted last week.

Test Suites

All tests were performed using **Postman**, a **Chrome browser plugin** that sends HTTP requests to my web application. I have included screen captures below, showing the input values for my tests. In addition to showing all of the JSON responses from Postman for HTTP GET, POST, PUT and DELETE requests, for HTTP delete requests, I have also included screen shots from the Google App Engine Administrator Console, showing that when an entity is deleted, all references to it are removed.

The following are two examples of one of my tests of the POST action for a Specialization entity in the Postman Chrome plugin. Note the URI (<http://malenah-api.appspot.com/specialization/>), the Headers and Body contents. The Header has the Accept: application/json to indicate that the web application accepts JSON responses, and the Content-type: application/x-www-form-urlencoded indicates that the content of the request being submitted to the web application is in the form of query string. [1]

The image displays two side-by-side screenshots of the Postman Chrome plugin interface. Both screenshots show a POST request to the URL `http://malenah-api.appspot.com/specialization/`. The left screenshot shows the 'Headers' tab with 'Accept' set to 'application/json' and 'Content-Type' set to 'application/x-www-form-urlencoded'. The right screenshot shows the 'Body' tab with 'x-www-form-urlencoded' selected and a single key-value pair: 'specializations[]' with the value 'Urgent care'. Both screenshots show a successful response with status '200 OK' and time '223 ms'. The response body is shown in 'Pretty' JSON format, containing a status message, duplicate array, added array (with one object for 'Urgent care' and key '5738600293466112'), and an invalid array.

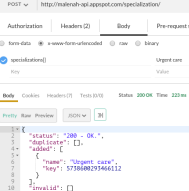
```
1 {
2   "status": "200 - OK.",
3   "duplicate": [],
4   "added": [
5     {
6       "name": "Urgent care",
7       "key": "5738600293466112"
8     }
9   ],
10  "invalid": []
11 }
```

I have divided my tests based on the following base URI's

1. /specialization: In addition to querying for Specialization entities (GET), these URI's are used for adding (POST), editing (PUT) and removing (DELETE) Specialization entities from the NDB datastore.
2. /provider: Used for querying for Provider entities in (GET), adding Provider entities to (POST), editing existing Provider entities in (PUT), and removing existing Provider entities from (DELETE) the NDB datastore.
3. /review: Used for querying for Review entities in (GET), adding Review entities to (POST), editing existing Review entities in (PUT), and removing existing Review entities from (DELETE) the NDB datastore.

4. /reply: Used for querying for Reply entities in (GET), adding Reply entities to (POST), editing existing Reply entities in (PUT), and removing existing Reply entities from (DELETE) the NDB datastore.

Base URI: <http://malenah-api.com/specialization>

Type	HTTP Request	URI	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome from Postman
empty datastore	GET	/specialization	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No specializations currently in database." }	{ "status": "200 - OK. No specializations currently in database." }
empty datastore	GET	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No specializations currently in database." }	{ "status": "200 - OK. No specializations currently in database." }
empty datastore	GET	/specialization/1	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No specializations matching the provided specialization id. " }	{ "status": "200 - OK. No specializations matching the provided specialization id. " }
basic, existing id	GET	/specialization/5657382461898752	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "name": "Motor Vehicle Accidents", "key": 5657382461898752 }	{ "name": "Motor Vehicle Accidents", "key": 5657382461898752 }
basic, invalid id	GET	/specialization/111	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No specializations matching the provided specialization id. " }	{ "status": "200 - OK. No specializations matching the provided specialization id. " }
basic, empty datastore (empty body)	POST	/specialization	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - Invalid input. Empty specializations[] field(s)." }	{ "status": "200 - Invalid input. Empty specializations[] field(s)." }
basic, missing required fields	POST	/specialization	Accept: application/json Content-Type: application/x-www-form-urlencoded	name: "	{ "status": "200 - OK.", "duplicate": [], "added": [], "invalid": "" }	{ "status": "200 - OK.", "duplicate": [], "added": [], "invalid": "" }
basic, empty datastore (first post)	POST	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "status": "200 - OK.", "duplicate": [], "added": [{ "name": "Urgent care", "key": 5676830073815040 }], "invalid": [] }	{ "status": "200 - OK.", "duplicate": [], "added": [{ "name": "Urgent care", "key": 5676830073815040 }], "invalid": [] }
duplicate entry	POST	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded	Key: specializations[] Value: Urgent care	{ "status": "200 - OK.", "duplicate": [{ "name": "Urgent care", "key": 5676830073815040 }], "added": [], "invalid": [] }	{ "status": "200 - OK.", "duplicate": [{ "name": "Urgent care", "key": 5676830073815040 }], "added": [], "invalid": [] }
basic, multiple entries including duplicates	POST	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded	Key: specializations[] Value: Urgent care Key: specializations[] Value: Sports Chiropractic Key: specializations[] Value: Emergency care	{ "status": "200 - OK.", "duplicate": [{ "name": "Urgent care", "key": 5676830073815040 }], "added": [{ "name": "Sports Chiropractic", "key": 5750085036015616 }, { "name": "Emergency care", "key": 5094432508477440 }], "invalid": [] }	{ "status": "200 - OK.", "duplicate": [{ "name": "Urgent care", "key": 5676830073815040 }], "added": [{ "name": "Sports Chiropractic", "key": 5750085036015616 }, { "name": "Emergency care", "key": 5094432508477440 }], "invalid": [] }
empty, no specialization id provided in URI	PUT	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid. No specialization id provided." }	{ "status": "400 - Invalid. No specialization id provided." }
empty, empty body	PUT	/specialization/5094432508477440	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input: missing name." }	{ "status": "400 - Invalid input: missing name." }

basic, modify existing record	PUT	/specialization/5094432508477440	Accept: application/json Content-Type: application/x-www-form-urlencoded	Key: name Value: Emergency Medicine (changed from Emergency care)	{ "status": "200 - OK", "name": "Emergency Medicine", "key": 5094432508477440 }	{ "status": "200 - OK", "name": "Emergency Medicine", "key": 5094432508477440 }
empty, no id provided in URI	DELETE	/specialization/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid. No specialization id provided." }	{ "status": "400 - Invalid. No specialization id provided." }
basic, delete existing record	DELETE	/specialization/5094432508477440	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - Delete specialization successful." }	{ "status": "200 - Delete specialization successful." }
duplicate, delete already deleted record	DELETE	/specialization/5094432508477440	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for specialization id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for specialization id." }
basic, invalid id	DELETE	/specialization/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for specialization id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for specialization id." }

1 All URI's in the table above are preceded by <http://malenah-api.com>

Deletion of Specialization entity removes its reference as well

Postman HTTP Delete Request	Google Cloud Administrator Console																								
<p>DELETE http://malenah-api.appspot.com/specialization/565316480401408</p> <p>Authorization Headers (2) Body Pre-request script</p> <p>form-data x-www-form-urlencoded raw binary</p> <p>Key Value</p> <p>Body Cookies Headers (7) Tests (0/0) Status 200 OK Time 220 ms</p> <p>Pretty Raw Preview JSON</p> <pre>1 { 2 "status": "200 - Delete specialization successful." 3 }</pre>	<p>Kind Specialization Filter entities</p> <table border="1"> <thead> <tr> <th>Name/ID</th> <th>name</th> </tr> </thead> <tbody> <tr><td><input type="checkbox"/> id=5203516087861248</td><td>LGBTQI Care</td></tr> <tr><td><input type="checkbox"/> id=5634387206995968</td><td>Transgender Medicine</td></tr> <tr><td><input type="checkbox"/> id=5653164804014080</td><td>Infectious diseases</td></tr> <tr><td><input type="checkbox"/> id=5657382461898752</td><td>Motor Vehicle Accidents</td></tr> <tr><td><input type="checkbox"/> id=5673385510043648</td><td>Endocrinology</td></tr> <tr><td><input type="checkbox"/> id=5693417237512192</td><td>Queer/Trans Health and Transition-related care</td></tr> <tr><td><input type="checkbox"/> id=5700735861784576</td><td>Ophthalmology</td></tr> <tr><td><input type="checkbox"/> id=5733935958982656</td><td>Work Injuries</td></tr> <tr><td><input type="checkbox"/> id=5738600293466112</td><td>Urgent care</td></tr> <tr><td><input type="checkbox"/> id=5750085036015616</td><td>Sports Chiropractic</td></tr> <tr><td><input type="checkbox"/> id=5766466041282560</td><td>Internal Medicine</td></tr> </tbody> </table> <p>Before deletion (11 entities)</p>	Name/ID	name	<input type="checkbox"/> id=5203516087861248	LGBTQI Care	<input type="checkbox"/> id=5634387206995968	Transgender Medicine	<input type="checkbox"/> id=5653164804014080	Infectious diseases	<input type="checkbox"/> id=5657382461898752	Motor Vehicle Accidents	<input type="checkbox"/> id=5673385510043648	Endocrinology	<input type="checkbox"/> id=5693417237512192	Queer/Trans Health and Transition-related care	<input type="checkbox"/> id=5700735861784576	Ophthalmology	<input type="checkbox"/> id=5733935958982656	Work Injuries	<input type="checkbox"/> id=5738600293466112	Urgent care	<input type="checkbox"/> id=5750085036015616	Sports Chiropractic	<input type="checkbox"/> id=5766466041282560	Internal Medicine
Name/ID	name																								
<input type="checkbox"/> id=5203516087861248	LGBTQI Care																								
<input type="checkbox"/> id=5634387206995968	Transgender Medicine																								
<input type="checkbox"/> id=5653164804014080	Infectious diseases																								
<input type="checkbox"/> id=5657382461898752	Motor Vehicle Accidents																								
<input type="checkbox"/> id=5673385510043648	Endocrinology																								
<input type="checkbox"/> id=5693417237512192	Queer/Trans Health and Transition-related care																								
<input type="checkbox"/> id=5700735861784576	Ophthalmology																								
<input type="checkbox"/> id=5733935958982656	Work Injuries																								
<input type="checkbox"/> id=5738600293466112	Urgent care																								
<input type="checkbox"/> id=5750085036015616	Sports Chiropractic																								
<input type="checkbox"/> id=5766466041282560	Internal Medicine																								
	<p>Kind Specialization Filter entities</p> <table border="1"> <thead> <tr> <th>Name/ID</th> <th>name</th> </tr> </thead> <tbody> <tr><td><input type="checkbox"/> id=5203516087861248</td><td>LGBTQI Care</td></tr> <tr><td><input type="checkbox"/> id=5634387206995968</td><td>Transgender Medicine</td></tr> <tr><td><input type="checkbox"/> id=5657382461898752</td><td>Motor Vehicle Accidents</td></tr> <tr><td><input type="checkbox"/> id=5673385510043648</td><td>Endocrinology</td></tr> <tr><td><input type="checkbox"/> id=5693417237512192</td><td>Queer/Trans Health and Transition-related care</td></tr> <tr><td><input type="checkbox"/> id=5700735861784576</td><td>Ophthalmology</td></tr> <tr><td><input type="checkbox"/> id=5733935958982656</td><td>Work Injuries</td></tr> <tr><td><input type="checkbox"/> id=5738600293466112</td><td>Urgent care</td></tr> <tr><td><input type="checkbox"/> id=5750085036015616</td><td>Sports Chiropractic</td></tr> <tr><td><input type="checkbox"/> id=5766466041282560</td><td>Internal Medicine</td></tr> </tbody> </table> <p>After deletion (10 entities): Notice that the Specialization entity with id 565316480401408 has been removed.</p>	Name/ID	name	<input type="checkbox"/> id=5203516087861248	LGBTQI Care	<input type="checkbox"/> id=5634387206995968	Transgender Medicine	<input type="checkbox"/> id=5657382461898752	Motor Vehicle Accidents	<input type="checkbox"/> id=5673385510043648	Endocrinology	<input type="checkbox"/> id=5693417237512192	Queer/Trans Health and Transition-related care	<input type="checkbox"/> id=5700735861784576	Ophthalmology	<input type="checkbox"/> id=5733935958982656	Work Injuries	<input type="checkbox"/> id=5738600293466112	Urgent care	<input type="checkbox"/> id=5750085036015616	Sports Chiropractic	<input type="checkbox"/> id=5766466041282560	Internal Medicine		
Name/ID	name																								
<input type="checkbox"/> id=5203516087861248	LGBTQI Care																								
<input type="checkbox"/> id=5634387206995968	Transgender Medicine																								
<input type="checkbox"/> id=5657382461898752	Motor Vehicle Accidents																								
<input type="checkbox"/> id=5673385510043648	Endocrinology																								
<input type="checkbox"/> id=5693417237512192	Queer/Trans Health and Transition-related care																								
<input type="checkbox"/> id=5700735861784576	Ophthalmology																								
<input type="checkbox"/> id=5733935958982656	Work Injuries																								
<input type="checkbox"/> id=5738600293466112	Urgent care																								
<input type="checkbox"/> id=5750085036015616	Sports Chiropractic																								
<input type="checkbox"/> id=5766466041282560	Internal Medicine																								

Base URI: <http://malenah-api.com/provider>

Type	HTTP Request	URI ¹	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome from Postman
empty datastore	GET	/provider	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No providers currently in database." }	{ "status": "200 - OK. No providers currently in database." }
empty datastore	GET	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No providers currently in database." }	{ "status": "200 - OK. No providers currently in database." }
empty datastore, invalid id	GET	/provider/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No providers matching the provided provider id." }	{ "status": "200 - OK. No providers matching the provided provider id." }
basic, existing id	GET	/provider/5629499534213120	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "website": "", "phone": "5034138988", "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120, "designation": "MD", "organization": "Legacy Medical Group - Northwest", "specializations": [5634387206995968, 5766466041282560], "accepting_new_patients": true, "email": "" }	{ "website": "", "phone": "5034138988", "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120, "designation": "MD", "organization": "Legacy Medical Group - Northwest", "specializations": [5634387206995968, 5766466041282560], "accepting_new_patients": true, "email": "" }

basic, invalid id	GET	/provider/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No providers matching the provided provider id." }	{ "status": "200 - OK. No providers matching the provided provider id." }
basic, empty datastore (empty body)	POST	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic, empty datastore (first post)	POST	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	first_name: Ben last_name: Ly designation: MD organization: Legacy Medical Group – Northwest specializations[]: 5634387206995968 specializations[]: 5766466041282560 phone: 5034138988 email: " website: " accepting_new_patients" True	{ "website": "", "status": "200 - OK", "first_name": "Ben", "last_name": "Ly", "designation": "MD", "email": "", "phone": "5034138988", "key": "5629499534213120", "organization": "Legacy Medical Group - Northwest", "specializations": [{ "name": "Transgender Medicine", "key": "5634387206995968" }, { "name": "Internal Medicine", "key": "5766466041282560" }], "accepting_new_patients": true }	{ "website": "", "status": "200 - OK", "first_name": "Ben", "last_name": "Ly", "designation": "MD", "email": "", "phone": "5034138988", "key": "5629499534213120", "organization": "Legacy Medical Group - Northwest", "specializations": [{ "name": "Transgender Medicine", "key": "5634387206995968" }, { "name": "Internal Medicine", "key": "5766466041282560" }], "accepting_new_patients": true }
basic, missing required fields	POST	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	first_name: " last_name: Ly designation: MD organization: Legacy Medical Group – Northwest specializations[]: 5634387206995968 specializations[]: 5766466041282560 phone: 5034138988 email: " website: " accepting_new_patients" True	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
duplicate entry	POST	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	first_name: Ben last_name: Ly designation: MD organization: Legacy Medical Group – Northwest specializations[]: 5634387206995968 specializations[]: 5766466041282560 phone: 5034138988 email: " website: " accepting_new_patients" True	{ "status": "400 - Invalid input: provider already exists in database." }	{ "status": "400 - Invalid input: provider already exists in database." }
empty, no id provided in URI	PUT	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid. No provider id provided." }	{ "status": "400 - Invalid. No provider id provided." }
empty, missing properties	PUT	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic, modify existing record	PUT	/provider/5707702298738688	Accept: application/json Content-Type: application/x-www-form-urlencoded	specializations[]: 5203516087861248 specializations[]: 5673385510043648 specializations[]: 5634387206995968	{ "website": "", "status": "200 - OK", "first_name": "Kelly", "last_name": "Washburn", "designation": "NP", "email": "", "phone": "5034138988", "key": "5707702298738688", "organization": "Merritt Health & Wellness", "specializations": [{ "name": "LGBTQI Care", "key": "5203516087861248" }, { "name": "Endocrinology", "key": "5673385510043648" }], }	{ "website": "", "status": "200 - OK", "first_name": "Kelly", "last_name": "Washburn", "designation": "NP", "email": "", "phone": "5034138988", "key": "5707702298738688", "organization": "Merritt Health & Wellness", "specializations": [{ "name": "LGBTQI Care", "key": "5203516087861248" }, { "name": "Endocrinology", "key": "5673385510043648" }], }

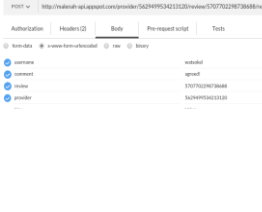
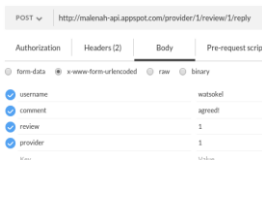
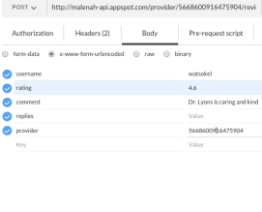
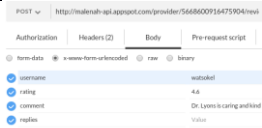
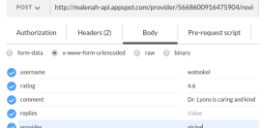
					<pre> "name": "Transgender Medicine", "key": 5634387206995968 }, "accepting_new_patients": true } </pre>	<pre> "name": "Transgender Medicine", "key": 5634387206995968 }, "accepting_new_patients": true } </pre>
empty, no id provided in URI	DELETE	/provider/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "400 - Invalid. No provider id provided." } </pre>	<pre> { "status": "400 - Invalid. No provider id provided." } </pre>
basic, delete existing record	DELETE	/provider/5707702298738688	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "200 - Delete provider successful." } </pre>	<pre> { "status": "200 - Delete provider successful." } </pre>
duplicate, delete already deleted record	DELETE	/provider/5707702298738688	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "400 - Invalid input. Unable to delete entity. No match for provider id." } </pre>	<pre> { "status": "400 - Invalid input. Unable to delete entity. No match for provider id." } </pre>
basic, invalid id	DELETE	/provider/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "400 - Invalid input. Unable to delete entity. No match for provider id." } </pre>	<pre> { "status": "400 - Invalid input. Unable to delete entity. No match for provider id." } </pre>
basic, wrong data type	DELETE	/provider/123a	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> 404 Not Found The resource could not be found. </pre>	<pre> 404 Not Found The resource could not be found. </pre>

¹ All URI's in the table above are preceded by <http://malenah-api.com>

Base URI: [http://malenah-api.com/provider/\[providerID\]/review/reviewID/reply/replyID](http://malenah-api.com/provider/[providerID]/review/reviewID/reply/replyID)

The following tests are for /provider/providerID and any URI's that extend on this base URI

Type	HTTP Request	URI ¹	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome from Postman
basic	GET	/provider/5629499534213120/review	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> [{ "username": "kendrick", "rating": 4.5, "replies": [], "comment": "busy office but caring staff", "key": 5707702298738688, "provider": 5629499534213120 }, { "username": "yosef", "rating": 4.5, "replies": [], "comment": "sensitive and cares about your issues", "key": 5715999101812736, "provider": 5629499534213120 }] </pre>	<pre> [{ "username": "kendrick", "rating": 4.5, "replies": [], "comment": "busy office but caring staff", "key": 5707702298738688, "provider": 5629499534213120 }, { "username": "yosef", "rating": 4.5, "replies": [], "comment": "sensitive and cares about your issues", "key": 5715999101812736, "provider": 5629499534213120 }] </pre>
basic invalid id	GET	/provider/123/review	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "400 - Invalid. No matches for the provided provider id." } </pre>	<pre> { "status": "400 - Invalid. No matches for the provided provider id." } </pre>
basic invalid URI wrong data type	GET	http://malenah-api.appspot.com/provider/review	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> 404 Not Found The resource could not be found. </pre>	<pre> 404 Not Found The resource could not be found. </pre>
basic valid URI and id		/provider/5629499534213120/review/5707702298738688	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "username": "kendrick", "rating": 4.5, "replies": [], "comment": "busy office but caring staff", "key": 5707702298738688, "provider": 5629499534213120 } </pre>	<pre> { "username": "kendrick", "rating": 4.5, "replies": [], "comment": "busy office but caring staff", "key": 5707702298738688, "provider": 5629499534213120 } </pre>
basic valid URI and invalid id	GET	/provider/5629499534213120/review/11	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "status": "400 - No match for review id." } </pre>	<pre> { "status": "400 - No match for review id." } </pre>
basic valid URI, valid ID's		/provider/5629499534213120/review/5707702298738688/reply/5685265389584384	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	<pre> { "username": "watsokel", "comment": "agreed!", "review": 5707702298738688, "key": 5685265389584384, "provider": 5629499534213120 } </pre>	<pre> { "username": "watsokel", "comment": "agreed!", "review": 5707702298738688, "key": 5685265389584384, "provider": 5629499534213120 } </pre>

basic valid URI and invalid id		/provider/5629499534213120/review/5707702298738688/reply/1	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - No match for reply id." }	{ "status": "400 - No match for reply id." }
basic valid URI, valid ID's	POST	/provider/5629499534213120/review/5707702298738688/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "comment": "agreed!", "username": "watsokel", "review": { "key": 5707702298738688 }, "status": "200 - OK", "key": 5066549580791808, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }	{ "comment": "agreed!", "username": "watsokel", "review": { "key": 5707702298738688 }, "status": "200 - OK", "key": 5066549580791808, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }
basic valid URI, invalid ID's	POST	/provider/5629499534213120/review/5707702298738688/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "comment": "agreed!", "username": "watsokel", "review": { "key": 5707702298738688 }, "status": "200 - OK", "key": 5066549580791808, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }	{ "comment": "agreed!", "username": "watsokel", "review": { "key": 5707702298738688 }, "status": "200 - OK", "key": 5066549580791808, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }
basic, POST	POST	/provider/5668600916475904/review/	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "comment": "Dr. Lyons is caring and kind", "rating": 4.6, "provider": { "first_name": "Jennifer", "last_name": "Lyons", "key": 5668600916475904 }, "username": "watsokel", "status": "200 - OK", "key": 5668600916475904, "replies": [] }	{ "comment": "Dr. Lyons is caring and kind", "rating": 4.6, "provider": { "first_name": "Jennifer", "last_name": "Lyons", "key": 5668600916475904 }, "username": "watsokel", "status": "200 - OK", "key": 5668600916475904, "replies": [] }
basic, missing required fields	POST	/provider/5668600916475904/review	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic, invalid required field	POST	/provider/5668600916475904/review	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "status": "400 - Invalid input: Provider id must be an integer." }	{ "status": "400 - Invalid input: Provider id must be an integer." }

Deletion of Provider entity removes its reference as well

Postman HTTP Delete Request

DELETE

http://malenah-api.appspot.com/provider/5724160613416960

Authorization

Headers (2)

Body

Pre-request script

form-data

x-www-form-urlencoded

raw

binary

Key

Value

Body

Cookies

Headers (7)

Tests (0/0)

Status 200 OK

Time 242 ms

Pretty

Raw

Preview

JSON

1 {

2 "status": "200 - Delete provider successful."

3 }

Google Cloud Administrator Console

Kind

Provider

Filter entities

Name/ID	accepting_new_patients	designation	email	first_name	last_name	organization	phone	specializations
id=5629499534213120	true	MD		Ben	Ly	Legacy Medical Group - Northwest	5034138988	[Key]Specialization: 5634387220994 [Key]Specialization: 5766466504123
id=5668600916475904	true	MD		Jennifer	Lyons	Oregon Eye Specialists	5034138988	[Key]Specialization: 570073586171
id=5724160613416960	true	NP		Kelly	Washburn	Merritt Health & Wellness	5034138988	[Key]Specialization: 56733851054 [Key]Specialization: 5634387220994
id=5741031244955648	true	DC	drochen@qchiropracticdx.com	Bekah	Cohen	Q Chiropractic, LLC	5035756020	[Key]Specialization: 579085036015616

Before deletion (4 entities):

Kind

Provider

Filter entities

Name/ID	accepting_new_patients	designation	email	first_name	last_name	organization	phone	specializations
id=5629499534213120	true	MD		Ben	Ly	Legacy Medical Group - Northwest	5034138988	[Key]Specialization: 5634387220994 [Key]Specialization: 5766466504123
id=5668600916475904	true	MD		Jennifer	Lyons	Oregon Eye Specialists	5034138988	[Key]Specialization: 570073586171
id=5741031244955648	true	DC	drochen@qchiropracticdx.com	Bekah	Cohen	Q Chiropractic, LLC	5035756020	[Key]Specialization: 5634387220994 [Key]Specialization: 579085036015616

After deletion (3 entities): Notice that the entity with id 5724160613416960 has been removed

Base URI: <http://malenah-api.com/review>

Type	HTTP Request	URI	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome from Postman
empty	GET	/review	Accept:	None	{	{

datastore			application/json Content-Type: application/x-www- form-urlencoded		"status": "200 - OK. No reviews currently in database. " }	"status": "200 - OK. No reviews currently in database. " }
empty datastore	GET	/review/	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "200 - OK. No reviews currently in database. " }	{ "status": "200 - OK. No reviews currently in database. " }
empty datastore, invalid id	GET	/review/123	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "200 - No match for review id." }	{ "status": "200 - No match for review id." }
basic, existing id	GET	/review/5639445604728832	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "username": "watsokel", "rating": 4.5, "replies": [], "comment": "caring and kind", "key": 5639445604728832, "provider": 5649050225344512 }	{ "username": "watsokel", "rating": 4.5, "replies": [], "comment": "caring and kind", "key": 5639445604728832, "provider": 5649050225344512 }
basic, invalid id	GET	/review/234	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 - No match for review id." }	{ "status": "400 - No match for review id." }
basic, empty datastore (empty body)	POST	/review	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic, empty datastore (first post)	POST	/review	Accept: application/json Content-Type: application/x-www- form-urlencoded	username: watsokel rating: 4.5 comment: caring and kind provider: 5649050225344512	{ "comment": "caring and kind", "rating": 4.5, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 }, "username": "watsokel", "status": "200 - OK", "key": 5629499534213120, "replies": [] }	{ "comment": "caring and kind", "rating": 4.5, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 }, "username": "watsokel", "status": "200 - OK", "key": 5629499534213120, "replies": [] }
basic, missing required fields	POST	/review/	Accept: application/json Content-Type: application/x-www- form-urlencoded		{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
empty, no id provided in URI	PUT	/review	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 - Invalid. No review id provided." }	{ "status": "400 - Invalid. No review id provided." }
empty, missing properties	PUT	/review/5649391675244544	Accept: application/json Content-Type: application/x-www- form-urlencoded	rating: " comment: 'meticulous'	{ "status": "400 - Invalid input: missing rating." }	{ "status": "400 - Invalid input: missing rating." }
basic, modify existing record	PUT	/review/5649391675244544	Accept: application/json Content-Type: application/x-www- form-urlencoded	rating: 4.52 comment: 'meticulous'	{ "comment": "meticulous", "rating": 4.52, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 }, "username": "watsokel", "status": "200 - OK", "key": 5649391675244544, "replies": [] }	{ "comment": "meticulous", "rating": 4.52, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 }, "username": "watsokel", "status": "200 - OK", "key": 5649391675244544, "replies": [] }
empty, no id provided in URI	DELETE	/review	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 -Invalid. No review id provided." }	{ "status": "400 -Invalid. No review id provided." }
basic, delete existing record	DELETE	/review/5649391675244544	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "200 - Delete review successful." }	{ "status": "200 - Delete review successful." }
duplicate, delete already deleted record	DELETE	/review/5649391675244544	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for review id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for review id." }
basic, invalid id	DELETE	/review	Accept: application/json Content-Type: application/x-www- form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for review id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for review id." }

¹ All URI's in the table above are preceded by <http://malenah-api.com>

Base URI: <http://malenah-api.com/reply>

Type	HTTP Request	URI ¹	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome
empty datastore	GET	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No replies currently in database." }	{ "status": "200 - OK. No replies currently in database." }
empty datastore	GET	/reply/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - OK. No replies currently in database." }	{ "status": "200 - OK. No replies currently in database." }
empty datastore, invalid id	GET	/reply/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - No match for reply id." }	{ "status": "400 - No match for reply id." }
basic, existing id	GET	/reply/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ { "username": "farah", "comment": "painless experience", "review": 5639445604728832, "key": 5144752345317376, "provider": 5649050225344512 }, { "username": "watsokel", "comment": "agreed!", "review": 5707702298738688, "key": 5066549580791808, "provider": 5629499534213120 } }	{ { "username": "farah", "comment": "painless experience", "review": 5639445604728832, "key": 5144752345317376, "provider": 5649050225344512 }, { "username": "watsokel", "comment": "agreed!", "review": 5707702298738688, "key": 5066549580791808, "provider": 5629499534213120 } }
basic, invalid id	GET	/reply/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - No match for reply id." }	{ "status": "400 - No match for reply id." }
basic, empty datastore (empty body)	POST	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic	POST	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	username: farah comment: some comment str! review: 5639445604728832 provider: 5649050225344512	{ "comment": "a reply", "username": "farah", "review": { "key": 5676073085829120 }, "status": "200 - OK", "key": 5668600916475904, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }	{ "comment": "a reply", "username": "farah", "review": { "key": 5676073085829120 }, "status": "200 - OK", "key": 5668600916475904, "provider": { "first_name": "Ben", "last_name": "Ly", "key": 5629499534213120 } }
Above operation should update the parent Review entity's replies attribute, as shown to the right Note that the replies attribute list now has a new reply ID appended to it:					{ "username": "watsokel", "rating": 4.5, "replies": [5629499534213120, 5668600916475904], "comment": "caring and kind", "key": 5676073085829120, "provider": 5629499534213120 }	{ "username": "watsokel", "rating": 4.5, "replies": [5629499534213120, 5668600916475904], "comment": "caring and kind", "key": 5676073085829120, "provider": 5629499534213120 }
basic, missing required fields	POST	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	username: farah comment: some comment str! review: provider: 5649050225344512	{ "status": "400 - Invalid input: missing properties." }	{ "status": "400 - Invalid input: missing properties." }
basic, invalid review id	POST	/reply	Accept: application/json	username: farah comment: some comment str! review: 1 provider: 5649050225344512	{ "status": "400 - Invalid input: no match for Review id. Please double check Provider id." }	{ "status": "400 - Invalid input: no match for Review id. Please double check Provider id." }
basic, invalid provider id	POST	/reply/	Content-Type: application/x-www-form-urlencoded	username: farah comment: some comment str! review: 5639445604728832 provider: 1	{ "status": "400 - Invalid input: no match for Provider id. Please double check Provider id." }	{ "status": "400 - Invalid input: no match for Provider id. Please double check Provider id." }
empty, no id provided in URI	PUT	/reply	Accept: application/json Content-Type:	username: maxine comment: my comment review	{ "status": "400 - Invalid. No reply id provided." }	{ "status": "400 - Invalid. No reply id provided." }

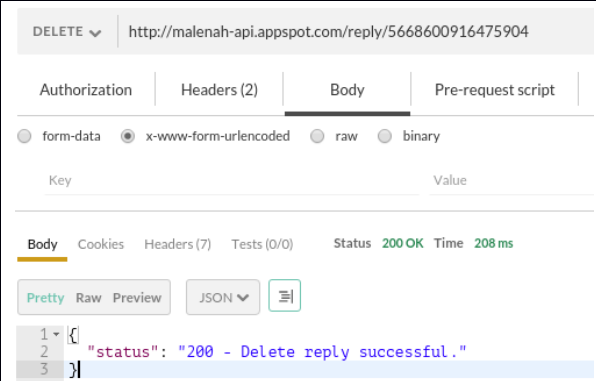
			application/x-www-form-urlencoded	4626744929681408 provider: 5998935441145856	}	}
empty, missing properties	PUT	/reply/5629499534213120	Accept: application/json Content-Type: application/x-www-form-urlencoded	username: maxine comment: my comment review provider: 5998935441145856	{"status": "400 - Invalid input: missing properties."}	{"status": "400 - Invalid input: missing properties."}
basic, modify existing record	PUT	/reply/5629499534213120/	Accept: application/json Content-Type: application/x-www-form-urlencoded		{ "comment": "my comment string", "username": "maxine", "review": { "key": 5639445604728832 }, "status": "200 - OK", "key": 5629499534213120, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 } }	{ "comment": "my comment string", "username": "maxine", "review": { "key": 5639445604728832 }, "status": "200 - OK", "key": 5629499534213120, "provider": { "first_name": "Kelly", "last_name": "Washburn", "key": 5649050225344512 } }
empty, no id provided in URI	DELETE	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid. No reply id provided." }	{ "status": "400 - Invalid. No reply id provided." }
basic, delete existing record	DELETE	/reply/5707702298738688/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "200 - Delete reply successful." }	{ "status": "200 - Delete reply successful." }
duplicate, delete already deleted record	DELETE	/reply/5707702298738688/	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for reply id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for reply id." }
basic, invalid id	DELETE	/reply	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	{ "status": "400 - Invalid input. Unable to delete entity. No match for reply id." }	{ "status": "400 - Invalid input. Unable to delete entity. No match for reply id." }

1 All URI's in the table above are preceded by <http://malenah-api.com>

Deletion of Reply entity removes its reference as well

The following screen shots of Postman and the Google App Engine Administrator Console show that when a correctly formed HTTP delete request is sent to my web application for a Reply entity, the result is that the entity and references to it are deleted.

Postman HTTP Delete Request



Google Cloud Administrator Console before and after

Kind: Reply

Name/ID	comment	provider	review	username
id=5144752345317376	painless experience	Key('Provider', 5649050225344512)	Key('Review', 5639445604728832)	farah
id=5668600916475904	my comment string	Key('Provider', 5649050225344512)	Key('Review', 5639445604728832)	maxine
id=5629499534213120	reply to a review	Key('Provider', 5629499534213120)	Key('Review', 5707702298738688)	farza
id=5668600916475904	another comment string!	Key('Provider', 5629499534213120)	Key('Review', 5707702298738688)	tomg

Before (4 Reply entities)

Kind: Reply

Name/ID	comment	provider	review	username
id=5144752345317376	painless experience	Key('Provider', 5649050225344512)	Key('Review', 5639445604728832)	farah
id=5629499534213120	reply to a review	Key('Provider', 5629499534213120)	Key('Review', 5707702298738688)	farza
id=5668600916475904	another comment string!	Key('Provider', 5629499534213120)	Key('Review', 5707702298738688)	tomg

After (3 Reply entities) Notice that the Reply entity with id 5668600916475904 has been removed

Explanation of Failed Tests

All tests pass, except for routes that are not handled by the web application. These include alphabetic characters entered into the URI's provider ID, review ID, or reply ID fields. For instance, the addition of the 'a' character results in a route that is not handled by the application, as shown in one of the test results below.

Type	HTTP Request	URI ¹	HTTP request header	HTTP request body	Expected Outcome	Actual Outcome
basic invalid id wrong data type	DELETE	/provider/123	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	404 Not Found The resource could not be found.	404 Not Found The resource could not be found.
basic invalid URI wrong data type	GET	http://malenah-api.appspot.com/provider/review	Accept: application/json Content-Type: application/x-www-form-urlencoded	None	404 Not Found The resource could not be found.	404 Not Found The resource could not be found.

Sources/Citations

- [1] <http://stackoverflow.com/questions/4007969/application-x-www-form-urlencoded-or-multipart-form-data>
- [2] Welford, Justin. CS496 Winter 2016, Notes
- [3] <http://www.restapitutorial.com/lessons/whatisrest.html#>
- [4] https://en.wikipedia.org/wiki/Representational_state_transfer#Architectural_properties