

## Project: Three Part Two

([https://3-dot-engine-985.appspot.com/API\\_library](https://3-dot-engine-985.appspot.com/API_library))

### **1. URL Structure**

The url structure I used was routing: [in API\_main.yaml], all use localhost:#### for me the #### is 9080 but for the instructor/grader it may be different.

The first is to the main library page; this lists all the keys to the currently loaded libraries ["http://localhost:9080/API\_library"]. The second gives all information specific to the Library whose key was entered ["http://localhost:9080/API\_library/5772779643207680"]. The third does nothing because GET is not allowed - the search implementation uses curl (please see last two pages for curl input details). The fourth lists all the keys to the currently loaded books ["http://localhost:9080/API\_book"]. The fifth gives all information specific to the Book whose key was entered ["http://localhost:9080/API\_book/5209829689786368"]. Lastly the sixth is not functioning correctly it is supposed to add a Library (second key) to the given Book (first key). But it is saying "405 Method Not Allowed: The method GET is not allowed for this resource" for "http://localhost:9080/API\_book/5209829689786368/API\_library/5772779643207680" when I try it in my browser url space. However it does work if I try it in curl (please see last two pages for curl input details).

### **2. RESTful Constraints**

Client Server: Yes

My client is not a fancy UI instead it uses curl, but in any case has zero concern with the data, it is just a medium to pass data to the portion that cares about data - whereas the server only cares about my data (libraries and books).

Stateless: Yes

There are no sessions - no tracking of client identifications. The server maintains my state using information imbedded in the resource provided by the server. The states are maintained by my 'API\_library' and 'API\_book' via server monitoring of data modified or removed.

Cacheable: No

In mimicking the instructor's video my system also does not implement caching. My client does not tell the server portion how long the resources are available for in its request for usage.

Layered: Yes

While there is only one server, there is only one access point for the client.

Uniform Interface: Yes

In mimicking the instructor's video, my system requests application/json formats. My messages also state and they are incompatible with anything other than application/json MIME types.

URI Navigation:

No.

### **3. Changes**

I made two changes to the Book type objects: I was originally going to use the books required ISBN as the unique key for a book (to allow for connectivity through my Library\_Books structure) but this however proved slightly more complicated than I thought and I ended up just making a 'bid' key for each book instead. I also originally had ISBN as an integer input, but when I was trying to run the curl commands to add a book I kept getting:

```
BadValueError: Entity has uninitialized properties: isbn
  when I entered:
    curl --data "title= Inkspell" --data "author= Funke" --data "isbn=" 987125864 -
H "Accept: application/json" http://localhost:9080/API_book
    curl --data "title= Inkspell" --data "author= Funke" --data '{"isbn":
987125864}' -H "Accept: application/json" http://localhost:9080/API_book
    $ curl --data "title= Inkspell" --data "author= Funke" -d '{"isbn": 987125864
}' -H "Accept: application/json" http://localhost:9080/API_book
    $ curl --data "title= Inkspell" --data "author= Funke" --data "isbn=" 987125864
-H "Accept: application/json" http://localhost:9080/API_book
```

```
BadValueError: Expected integer, got u' 987125864'
  when I entered:
    curl --data "title= Inkspell" --data "author= Funke" --data "isbn= 987125864" -
H "Accept: application/json" http://localhost:9080/API_book
```

I could not figure out in the *curl --help* or Google searching how to pass an integer, so I changed the value in API\_nb\_models [line 17] from: *isbn = ndb.IntegerProperty(required=True)* to: *isbn = ndb.StringProperty(required=True)* and then it worked.

I also did not do a fully alone function for Library\_Books as I intended to, it ended up being simpler to have a LibraryBooks class within API\_book.py.

### **4. What would I do differently?**

I would have made a nice UI for the application because figuring out curl was more complicated/time consuming than I thought - and this affected my output program because curl was my testing medium and because it took so long to download viably and took so long for me to figure out the minimal usage my testing was delayed and thus my program is not as functional/in depth as it could have been. I also would have taken more than an hour to try to figure out how to use the ISBN as the book key and I would have spent more time researching how to pass an integer in curl for data values.

## CURL Commands

### Book Creation

**input:** (add book with only required elements)

```
$ curl --data "title= Inkspell" --data "author= Funke" --data "isbn= 987125864" -H "Accept: application/json" http://localhost:9080/API_book
```

**output:**

```
{"isbn": " 987125864", "author": " Funke", "key": 5209829689786368, "genres": [], "cover": [], "title": " Inkspell", "ownership": [], "libs": []}
```

**input:** (add book with all elements but libs)

```
$ curl --data "title= Inkspell" --data "author= Funke" --data "isbn= 987125864" --data "cover= Hard" --data "ownership= want" --data "genres[]= 'Fantasy', 'Mystery'" -H "Accept: application/json" http://localhost:9080/API_book
```

**output:**

```
{"genres": [" 'Fantasy', 'Mystery'"], "libs": [], "cover": " Hard", "author": " Funke", "ownership": " want", "title": " Inkspell", "key": 6265360852451328, "isbn": " 987125864"}
```

**input:** (add book with all elements)

```
$ curl --data "title= Inkspell" --data "author= Funke" --data "isbn= 987125864" --data "cover= Hard" --data "ownership= want" --data "genres[]= 'Fantasy', 'Mystery'" -d "libs[]=4928354713075712" -H "Accept: application/json" http://localhost:9080/API_book
```

**output:**

```
{"key": 5983885875740672, "author": " Funke", "ownership": " want", "cover": " Hard", "title": " Inkspell", "isbn": " 987125864", "genres": [" 'Fantasy', 'Mystery'"], "libs": [4928354713075712]}
```

Datastore Viewer

Entity Kind

Book ▼ List Entities Create New Entity Select a different namespace

<input type="checkbox"/>	Key	Write Ops	ID	Key Name	author	cover	genres	isbn	libs	ownership	title
<input type="checkbox"/>	agpkZXZ...	8	5209829689786368	Funke				987125864			Inkspell
<input type="checkbox"/>	agpkZXZ...	16	5983885875740672	Funke	Hard	[u'Fantasy', 'Mystery']		987125864	[datastore_types.Key.from_p...	Want	Inkspell
<input type="checkbox"/>	agpkZXZ...	14	6265360852451328	Funke	Hard	[u'Fantasy', 'Mystery']		987125864		Want	Inkspell

Delete Flush Memcache

Results 1 - 3 of 3

### Library Creation

**input:** (add library with all required elements)

```
$ curl --data "name= Mac" --data "location= McMinnville" --data "accessability= Public" -H "Accept: application/json" http://localhost:9080/API_library
```

**output:**

```
{"location": " McMinnville", "name": " Mac", "key": 6335729596628992, "accessability": " Public", "hours": []}
```

**input:** (add library with all elements)

```
$ curl --data "name= Mac" --data "location= 3rd St McMinnville OR 97128" --data "accessability= Public" --data "hours= M-F 8:00am to 5:00pm Saturday Noon to 6:00pm Sunday Closed" -H "Accept: application/json" http://localhost:9080/API_library
```

**output:**

```
{"hours": " M-F 8:00am to 5:00pm Saturday Noon to 6:00pm Sunday Closed",  
"accessability": " Public", "location": " 3rd St McMinnville OR 97128", "key":  
4928354713075712, "name": " Mac"}
```

Datastore Viewer

Entity Kind

Library ▼ [List Entities](#) [Create New Entity](#) [Select a different namespace](#)

<input type="checkbox"/>	Key	Write Ops	ID	Key Name	accessability	hours	location	name
<input type="checkbox"/>	agpkZXZ-...	10	4928354713075712		Public	M-F 8:00am to 5:00pm Satur...	3rd St McMinnville OR 97128	Mac
<input type="checkbox"/>	agpkZXZ-...	8	5772779643207680		Public		McMinnville	Mac

[Delete](#) [Flush Memcache](#)

Results 1 - 2 of 2

---

## Book Editing Libs [Libraries]

---

**input:** (add library to book)

```
$ curl -X PUT -H "Accept: application/json"  
http://localhost:9080/API_book/5209829689786368/API_library/5772779643207680
```

**output:**

```
{"libs": [5772779643207680], "author": " Funke", "genres": [], "title": " Inkspell",  
"ownership": null, "isbn": " 987125864", "key": 5209829689786368, "cover": null}
```

Datastore Viewer

Entity Kind

Book ▼ [List Entities](#) [Create New Entity](#) [Select a different namespace](#)

<input type="checkbox"/>	Key	Write Ops	ID	Key Name	author	cover	genres	isbn	libs	ownership	title
<input type="checkbox"/>	agpkZXZ-...	14	5209829689786368	Funke	None			987125864	[datastore_types.Key.from_p...	None	Inkspell
<input type="checkbox"/>	agpkZXZ-...	16	5983885875740672	Funke	Hard		[u" 'Fantasy', 'Mystery']	987125864	[datastore_types.Key.from_p...	Want	Inkspell
<input type="checkbox"/>	agpkZXZ-...	14	6265360852451328	Funke	Hard		[u" 'Fantasy', 'Mystery']	987125864		Want	Inkspell

[Delete](#) [Flush Memcache](#)

Results 1 - 3 of 3

---

## Library Searching

---

**input:** (search library)

```
$ curl --data "name= Mac" -H "Accept: application/json"  
http://localhost:9080/API_library/search
```

**output:**

```
{"keys": [4928354713075712, 5772779643207680]}
```

**input:** (search library)

```
$ curl --data "accessability= Public" -H "Accept: application/json"  
http://localhost:9080/API_library/search
```

**output:**

```
{"keys": [4928354713075712, 5772779643207680]}
```