#### Starting a New Sefaria Project 101

- 1. Get Sefaria working locally (instructions here: <a href="https://github.com/Sefaria/Sefaria-Project">https://github.com/Sefaria/Sefaria-Project</a>; some python2 code has to be updated to python3 for pip to work)
- 2. Set your account with the roles of staff, admin, superuser this way:
  - a) Start your local Django server
  - b) Use the localhost site to register a new user (click "Sign up")
  - c) Start the Sefaria CLI (./cli in the root of Sefaria-Project)
  - d) Enter following commands. Replace <my email> with the email you used to sign up

```
from django.contrib.auth.models import User user = User.objects.get(email="<my email>") user.is_staff = True user.is_admin = True user.is_superuser = True user.save()
```

- e) Use the ID of this user for the user parameter. To find out the user ID, type: print(user.id)
- 3. Go to MongoDBCompass, find collection "apikeys" and add a new document with your self-defined "key"
- 4. Add new categries by:
  - a) Go to the CLI by using command ".\cli"
  - b) Enter following commands

```
import django
django.setup()
from sefaria.model import *
from sefaria.helper.category import create_category
<new root category name>_root_cat = create_category(["<category path layer 1>",
    "<category path layer 2>..."], "<category name English>", "<category name Hebrew>")
examples:
kangyur_root_cat = create_category(["Kangyur"], "Kangyur", "Kangyur_")
```

Which results in:

1. Create an object in sefaria.category

```
_id: ObjectId('646037e12331a7c06a048d9f')
lastPath: "Kangyur"

path: Array
0: "Kangyur"
depth: 1
sharedTitle: "Kangyur"
```

2. Create object in sefaria.term

3. Which can be edited using MongoDB Compass by adding a enShortDesc if there is to be a description featured under the category name thus:

```
_id: ObjectId('646042232331a7c06a048da7')
lastPath: "Kangyur"

path: Array
    0: "Kangyur"
depth: 1
sharedTitle: "Kangyur"
enShortDesc: "The collected scriptures: the Tibetan translations of the Indian texts..."
```

4. Which on the frontend homepage will look like

## Kangyur

The collected scriptures: the Tibetan translations of the Indian texts that are considered to be the words of the Buddha.

```
Then adding the subcategory of discipline/vinaya:

Command entered:

discipline_cat = create_category(["Kangyur", "Discipline"], "Discipline", "Discipline_")
```

Which results in, upon clicking on Kangyur:

# KANGYUR

DISCIPLINE

5. New texts can be added to the index using the following python script (modified from api post example on <a href="https://github.com/Sefaria/Sefaria-Project/wiki/API-Documentation#post-apiindextitle">https://github.com/Sefaria/Sefaria-Project/wiki/API-Documentation#post-apiindextitle</a>)

```
import urllib
import urllib.error
from urllib.error import URLError, HTTPError
import urllib.parse
import urllib.request
import json
index = {
     "title": "<textTitle>",
     "heTitle": "<textTitle>e_",
     "titleVariants": ["<variantName1>", "<variantName2>"...],
     "sectionNames": ["Chapter", "Paragraph"],
     "categories": ["level1CategoryName", "level2CategoryName"...], (e.g.,["Kangyur",
"Discipline"])
def post_index(index):
     url = 'http://127.0.0.1:8000/api/index/' + index["title"].replace(" ", "_")
     indexJSON = json.dumps(index)
     values = {
          'json': indexJSON,
          'apikey': "apikey"
     data = urllib.parse.urlencode(values)
     binary_data = data.encode('ascii')
     req = urllib.request.Request(url, binary_data)
     try:
          response = urllib.request.urlopen(req)
     except (HTTPError):
          print("error")
          print ('Error code: ', e.code)
post index(index)
```

Which results in the name of the text appearing under the subchapter:

## KANGYUR

DISCIPLINE

## Chapters on Monastic Discipline

6. Paragraphs are added one chapter at a time with the following:

```
import urllib
import urllib.error
from urllib.error import URLError, HTTPError
import urllib.parse
import urllib.request
import json
```

```
text = {
     "versionTitle": "<versionTitle>",
     "versionSource": "<versionSource, could be a url>",
     "language": "<en or he; he for tibetan>",
     "text": [
          "<textOfParagraph1>",
          "<textOfParagraph2>",
          "<textOfParagraph3>"<add more lines for more paragraphs>
     ]
}
def post_text(ref, text):
     textJSON = json.dumps(text)
     ref = ref.replace(" ", " ")
     url = 'http://localhost:8000/api/texts/%s' % ref
     values = {'json': textJSON, 'apikey': 'apikey'}
     data = urllib.parse.urlencode(values)
     binary_data = data.encode('ascii')
     req = urllib.request.Request(url, binary_data)
     try:
          response = urllib.request.urlopen(req)
          print(response.read())
     except (HTTPError):
          print("error")
          print('Error code: ', e.code)
          print(e.read())
```

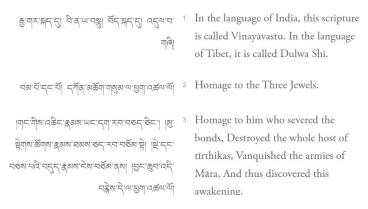
post\_text("<textTitle> <paragraph number>", text)

Running the above one time each for the English and Tibetan versions results in:



#### Chapters on Monastic Discipline

#### Chapter 1



#### 7. Issues to resolve

### a) Chapter numbers not showing Chapters on Monastic Discipline

KANGYUR

Continue Reading

Contents Versions

Chapter

#### b) Chapter title is number instead of text

#### Chapter 1

ক্র'শ্বমান্ত্র বির্থেবস্থা ব্রমান্ত্র ব্রথবে া In the language of India, this scripture is called Vinayavastu. In the language of Tibet, it is called Dulwa Shi.