Implementing Nowcasting Intelligent System for Media

Implementing Nowcasting Intelligent System for Media

Team Members

NAME	NUID
PRIYANKA DILIP SHINDE	001524484

KSHITIJ ZUTSHI	001021288
DWITHIKA SHETTY	002114630

Part1: Get familiar with the Gretel design and run the tutorials

NAME	NUID
PRIYANKA DILIP SHINDE	001524484
KSHITIJ ZUTSHI	001021288
DWITHIKA SHETTY	002114630

Create the synthetic data configuration

Load the default configuration template. This template will work well for most datasets. View other templates at https://github.com/gretelai/gretel-blueprints/tree/main/config_templates/gretel/synthetics

In [5]:
```
import json
from smart_open import open
import yaml

with open("https://raw.githubusercontent.com/gretelai/gretel-blueprints/main/config_templates/gretel/synthetics/default
    config = yaml.safe_load(stream)

# Set the model epochs to 50
config['models'][0]['synthetics']['params']['epochs'] = 50

print(json.dumps(config, indent=2))
```

```
{
  "schema_version": "1.0",
  "models": [
    {
      "synthetics": {
        "data_source": "__tmp__",
        "params": {
          "epochs": 50,
          "batch_size": 64,
          "vocab_size": 20000,
          "reset_states": false,
          "learning_rate": 0.01,
          "rnn_units": 256,
          "dropout_rate": 0.2,
          "overwrite": true,
          "early_stopping": true,
          "gen_temp": 1.0,
          "predict_batch_size": 64,
```

Load and preview the source dataset

Specify a data source to train the model on. This can be a local file, web location, or HDFS file.

In [6]:
```
# Load and preview the DataFrame to train the synthetic model on.
import pandas as pd

dataset_path = '/content/train.csv'
# dataset_path = 'https://gretel-public-website.s3-us-west-2.amazonaws.com/datasets/USAdultIncome5k.csv'
df = pd.read_csv(dataset_path)
df.to_csv('training_data.csv', index=False)
df
```

Out[6]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	tot
0	OH	107	area_code_415	no	yes	26	161.6	123	27.47	
1	NJ	137	area_code_415	no	no	0	243.4	114	41.38	
2	OH	84	area_code_408	yes	no	0	299.4	71	50.90	
3	OK	75	area_code_415	yes	no	0	166.7	113	28.34	
4	MA	121	area_code_510	no	yes	24	218.2	88	37.09	
...	...	...	...	...	...	...	...	...	...	...
4245	MT	83	area_code_415	no	no	0	188.3	70	32.01	
4246	WV	73	area_code_408	no	no	0	177.9	89	30.24	
4247	NC	75	area_code_408	no	no	0	170.7	101	29.02	
4248	HI	50	area_code_408	no	yes	40	235.7	127	40.07	
4249	VT	86	area_code_415	no	yes	34	129.4	102	22.00	

4250 rows × 20 columns

Train the synthetic model

In this step, we will task the worker running in the Gretel cloud, or locally, to train a synthetic model on the source dataset.

In [9]:
```
from gretel_client.helpers import poll

model = project.create_model_obj(model_config=config)
model.data_source = 'training_data.csv'
model.submit(upload_data_source=True)

poll(model)
```

```
INFO: Starting poller
```


```
{
    "uid": "621d3ae508ddd4b2c4ab007f",
    "guid": "model_25ktDAIFvmOPfVwBPuu81v3bHyo",
    "model_name": "shaggy-quizzical-dingo",
    "runner_mode": "cloud",
    "user_id": "621d391bbff6213002668507",
    "user_guid": "user_25ksHQGND5uMVGKEdPGyWXSx93n",
    "billing_domain": null,
    "billing_domain_guid": null,
    "project_id": "621d3ada2199163c444ddcf9",
    "project_guid": "proj_25ktBiepgKXQkyKtAakQpBEEhxA",
    "status_history": {
        "created": "2022-02-28T21:13:09.159589Z"
    },
    "last_modified": "2022-02-28T21:13:09.166345Z",
    "status": "created",
    "last_active_hb": null,
    "duration_minutes": null
```

View the generated synthetic data

In [10]:
```
# View the synthetic data

synthetic_df = pd.read_csv(model.get_artifact_link("data_preview"), compression='gzip')

synthetic_df
```

Out[10]:

	state	account_length	area_code	international_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day_charge	tot
0	NY	107	area_code_415	no	no	0	255.4	97	43.42	
1	ND	59	area_code_415	no	no	0	201.4	98	34.24	
2	CT	41	area_code_408	no	no	0	166.0	100	28.22	
3	CO	95	area_code_408	yes	no	0	121.4	104	20.64	
4	CO	152	area_code_408	no	no	0	176.3	84	29.95	
...	...	...	...	...	...	...	...	...	...	
4995	OH	159	area_code_415	no	no	0	189.6	101	32.23	
4996	AZ	131	area_code_415	yes	no	0	75.6	119	13.57	
4997	NV	98	area_code_510	no	no	0	124.4	110	21.15	
4998	CO	172	area_code_510	no	no	0	184.6	111	31.38	
4999	LA	42	area_code_415	yes	yes	26	194.5	114	33.07	

5000 rows × 20 columns

View the synthetic data quality report

In [11]:
```
# Generate report that shows the statistical performance between the training and synthetic data

import IPython
from smart_open import open

IPython.display.HTML(data=open(model.get_artifact_link("report")).read())
```

Out[11]: Gretel synthetics report

Gretel
Synthetic Report

Generated 02/28/2022, 21:13

Synthetic Data Quality Score

Privacy Protection Level

Data Summary Statistics

Field Correlation Stability

Deep Structure Stability

Field Distribution Stability

	Training Data	Synthetic Data
Row Count	4250	4250
Column Count	20	20

Training Lines Duplicated	--	0

What do these values mean?

Privacy Protection Summary

	Default Privacy Protections	Advanced Protections

Outlier Filter

Similarity Filter

Overfitting Prevention

Differential Privacy

Training field overview

Field	Unique	Missing	Ave. Length	Type	Distribution Stability
account_length	215	0	2.49	Categorical	
total_day_calls	120	0	2.51	Categorical	Excellent
total_eve_calls	123	0	2.52	Categorical	
state	51	0	2.00	Categorical	Excellent
total_night_calls	128	0	2.51	Categorical	
total_day_charge	1843	0	4.88	Numeric	Excellent
total_day_minutes	1843	0	4.93	Numeric	
total_eve_minutes	1773	0	4.98	Numeric	Excellent
total_night_charge	992	0	4.23	Numeric	
state	51	0	2.00	Categorical	Excellent
total_night_calls	128	0	2.51	Categorical	
total_day_charge	1843	0	4.88	Numeric	Excellent
total_day_minutes	1843	0	4.93	Numeric	
total_eve_minutes	1773	0	4.98	Numeric	Excellent
total_night_charge	992	0	4.23	Numeric	
total_eve_charge	1572	0	4.85	Numeric	Excellent
number_vmail_messages	46	0	1.26	Categorical	
total_night_minutes	1757	0	4.98	Numeric	Excellent
total_intl_charge	168	0	3.90	Numeric	
total_intl_minutes	168	0	3.56	Numeric	Excellent
total_intl_calls	21	0	1.04	Categorical	
area_code	3	0	13.00	Categorical	Excellent
churn	2	0	2.14	Binary	
number_customer_service_calls	10	0	1.00	Categorical	Excellent
international_plan	2	0	2.09	Binary	
voice_mail_plan	2	0	2.26	Binary	Excellent

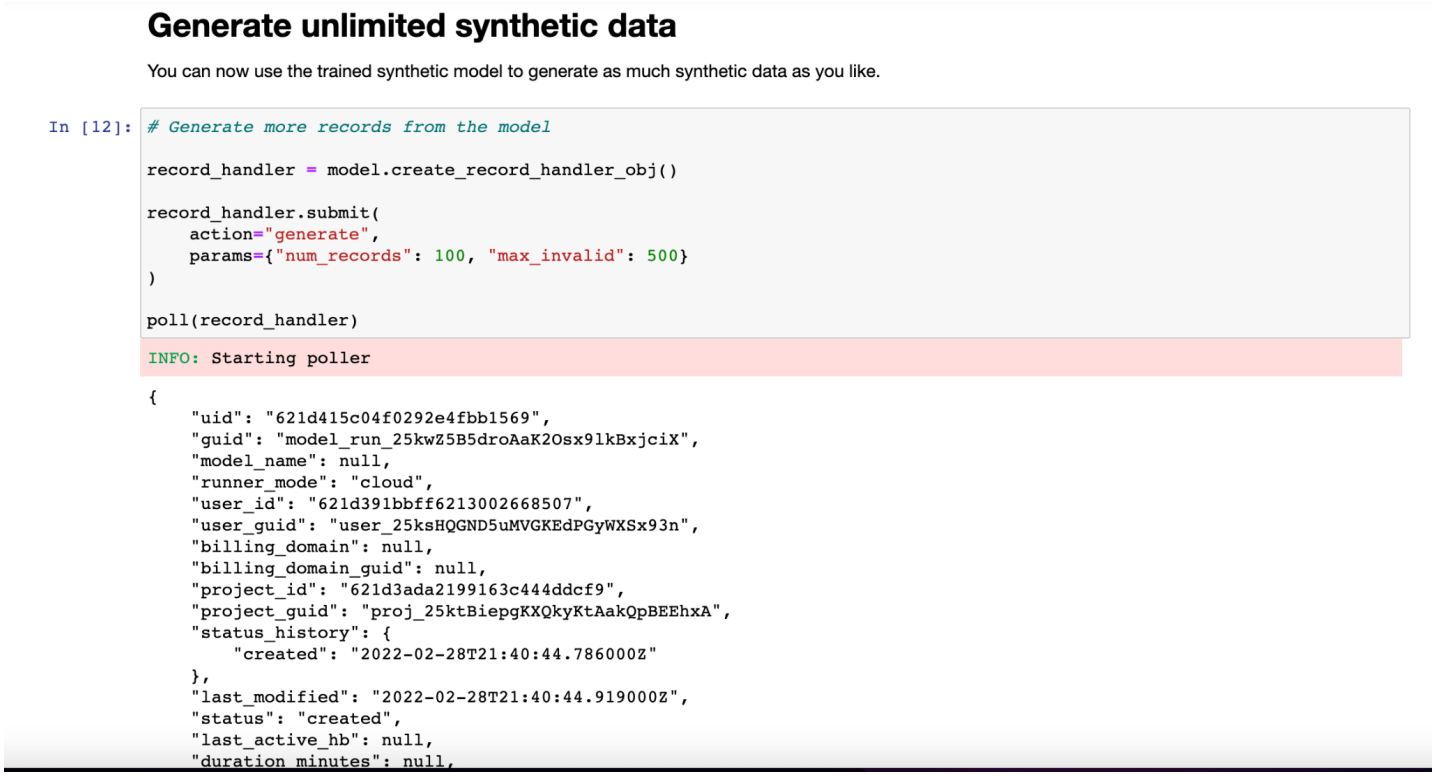
Training and Synthetic Data Correlation

Principal Component Analysis

Field Distribution Comparisons

Training Data Synthetic Data

Copyright © 2021 Gretel Labs. All rights reserved.



Generate unlimited synthetic data

You can now use the trained synthetic model to generate as much synthetic data as you like.

In [12]:

```
# Generate more records from the model

record_handler = model.create_record_handler_obj()

record_handler.submit(
    action="generate",
    params={"num_records": 100, "max_invalid": 500}
)

poll(record_handler)
```

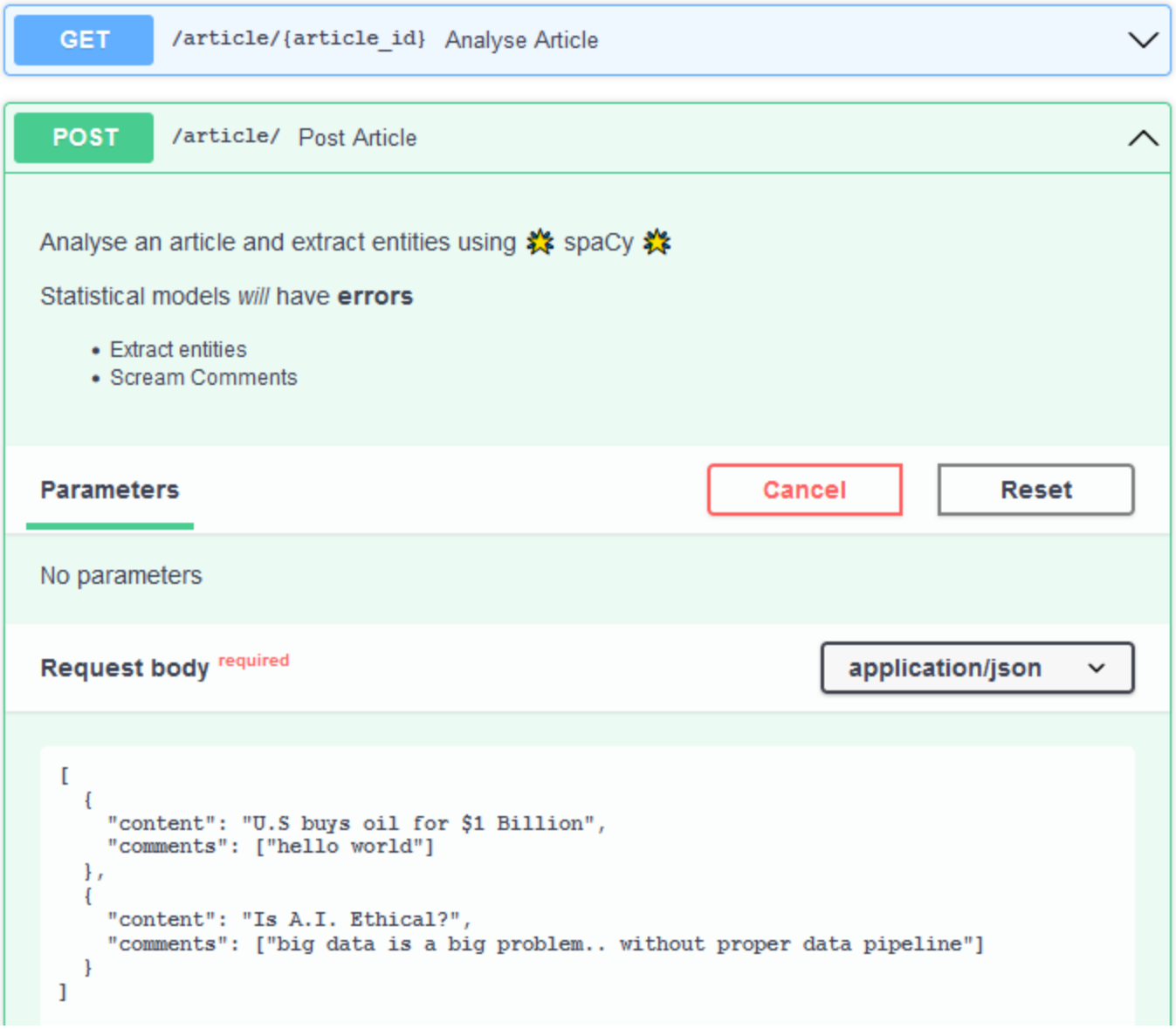
```
INFO: Starting poller

{
    "uid": "621d415c04f0292e4fbb1569",
    "guid": "model_run_25kwZ5B5droAaK2Osx9lkBxjciX",
    "model_name": null,
    "runner_mode": "cloud",
    "user_id": "621d391bbff6213002668507",
    "user_guid": "user_25ksHQGND5uMVGKEdPGyWXSx93n",
    "billing_domain": null,
    "billing_domain_guid": null,
    "project_id": "621d3ada2199163c444ddcf9",
    "project_guid": "proj_25ktBiepgKXQkyKtAakQpBEEhxA",
    "status_history": {
        "created": "2022-02-28T21:40:44.786000Z"
    },
    "last_modified": "2022-02-28T21:40:44.919000Z",
    "status": "created",
    "last_active_hb": null,
    "duration_minutes": null,
```

Part 2: Designing the REST API

Introduction to FAST API

To try Fast API software as part of assignment, we have tried a tutorial as seen in the screenshot below where in the endpoint i.e article_id will be provided by the user. If an integer is provided it will display the same integer.



In the below screenshot , when we pass a string “U.S buys oil for 1$ Billion”, by using pre pre-trained machine learning model (nlp) and finding out the entities of the string passed.

Code | Details

200

Response body


```
{
  "ents": [
    {
      "text": "U.S",
      "label": "GPE"
    },
    {
      "text": "$1 Billion",
      "label": "MONEY"
    },
    {
      "text": "A.I. Ethical",
      "label": "ORG"
    }
  ],
  "comments": [
    "HELLO WORLD",
    "BIG DATA IS A BIG PROBLEM.. WITHOUT PROPER DATA PIPELINE"
  ]
}
```

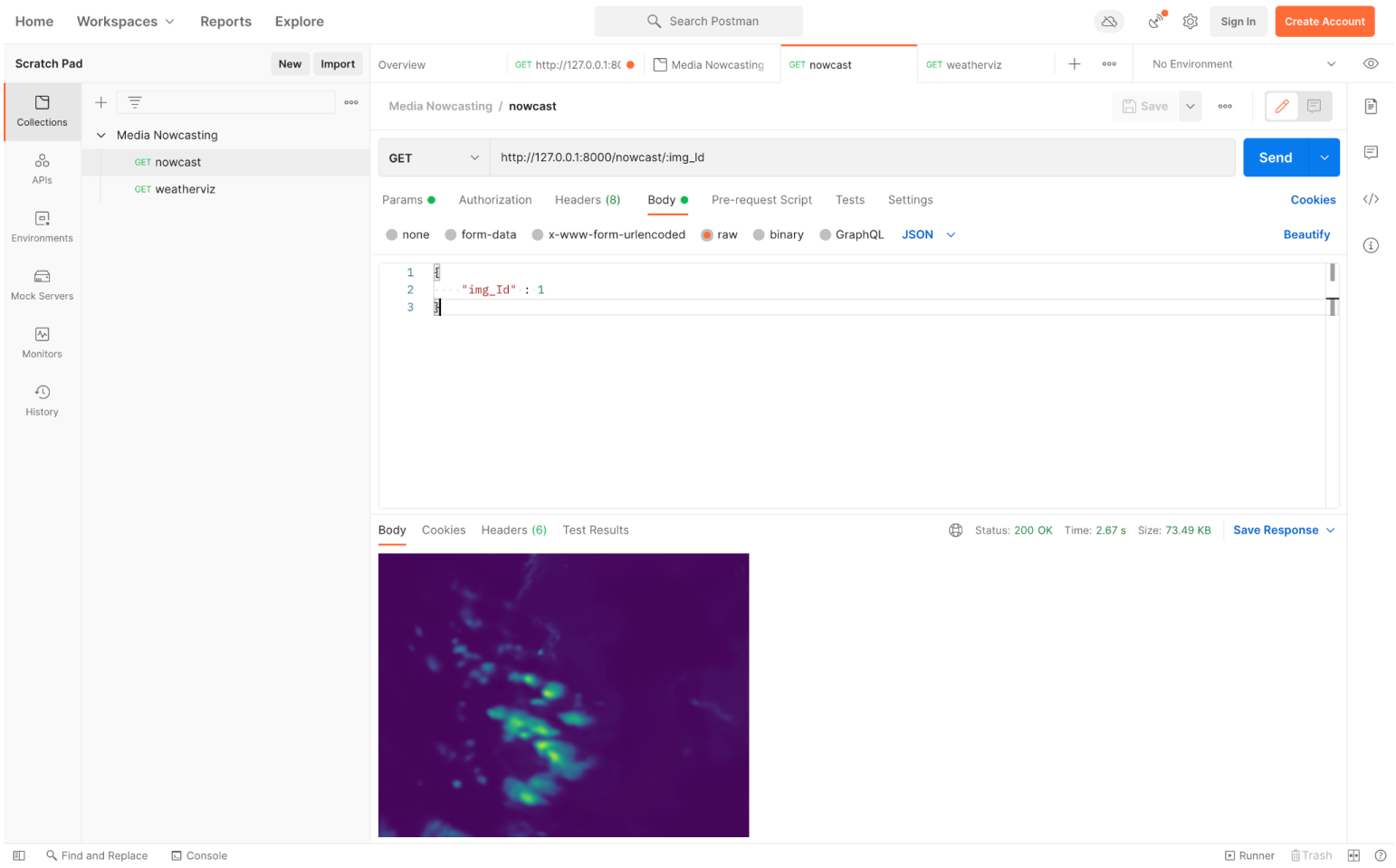

Download

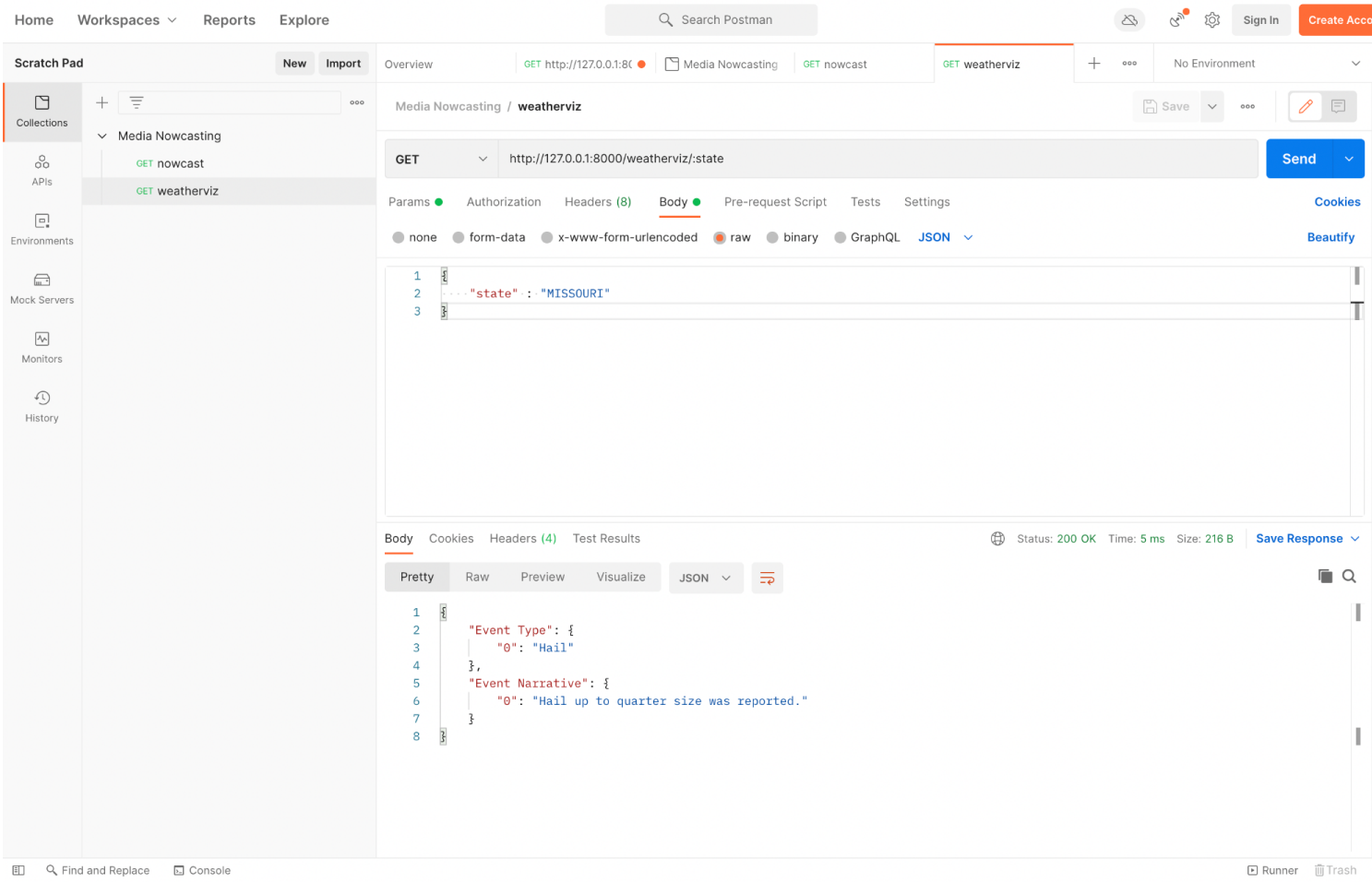
Response headers


```
content-length: 201
content-type: application/json
date: Wed,09 Mar 2022 17:34:43 GMT
server: uvicorn
```


Postman Application JSON Execution

Executing model in postman by passing JSON


Home
Workspaces
Reports
Explore
Search Postman
Sign In
Create Account
Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Media Nowcasting
GET nowcast
GET weatherviz
Overview
GET http://127.0.0.1:8(
Media Nowcasting
GET nowcast
GET weatherviz
No Environment
Media Nowcasting / nowcast
Save
GET
http://127.0.0.1:8000/nowcast/:img_Id
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1 {
2 "img_Id" : 1
3 }
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 2.67 s
Size: 73.49 KB
Save Response
Find and Replace
Console
Runner
Trash



Operations supported by API

1. Endpoint 1: **/nowcast/{img_Id}**
   To render next hour's nowcasting 12 output images based on their image id that ranges from 0-11 where each image id represents 5 min interval of nowcasted image.

2. Endpoint 2: **/weatherviz/{state}**
   To display the event type and event narrative for selected states from the drop down menu.

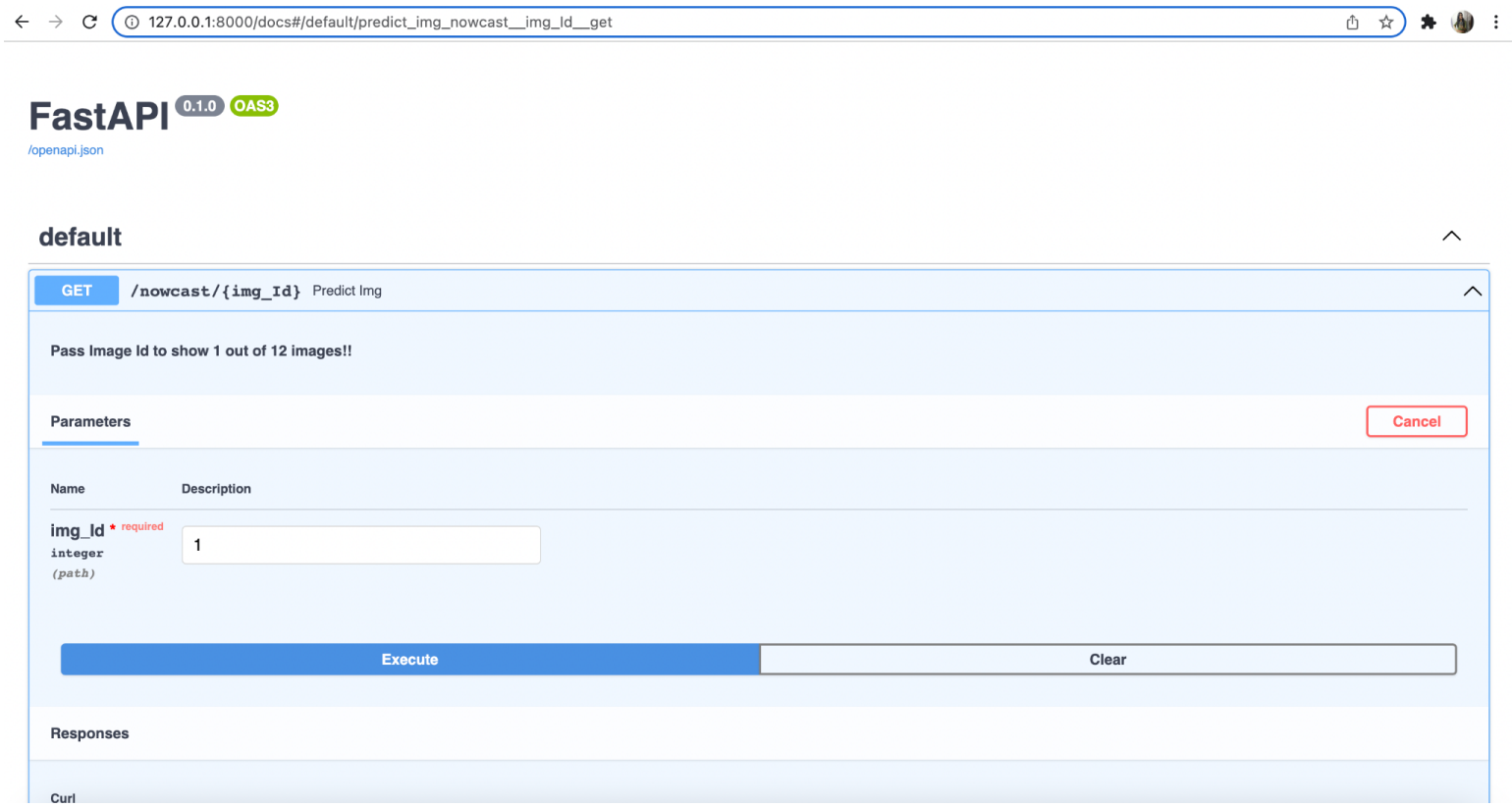
Assumptions for Media Nowcasting

1. We will be generating next hours nowcast and allow users to render those 12 nowcasted images for every 5 min interval.

2. Users can get the event type and event narrative for the selected state from the drop down menu.

Implementation

Deployment of FAST API for Media Nowcasting



127.0.0.1:8000/docs#/default/predict_img_nowcast__img_Id__get

Responses

Curl

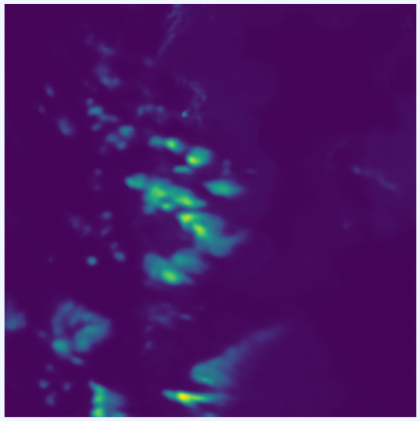
```
curl -X 'GET' \
  'http://127.0.0.1:8000/nowcast/1' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/nowcast/1
```

Server response

Code	Details
200	Response body



Response headers

127.0.0.1:8000/docs#/default/filter_nowcast_weatherviz__state__get

GET `/weatherviz/{state}` Filter Nowcast

Get weather visualization stats based on state selection!

Parameters Cancel

Name	Description
state * required string *(path)*	MISSOURI

Execute Clear

Responses

Curl

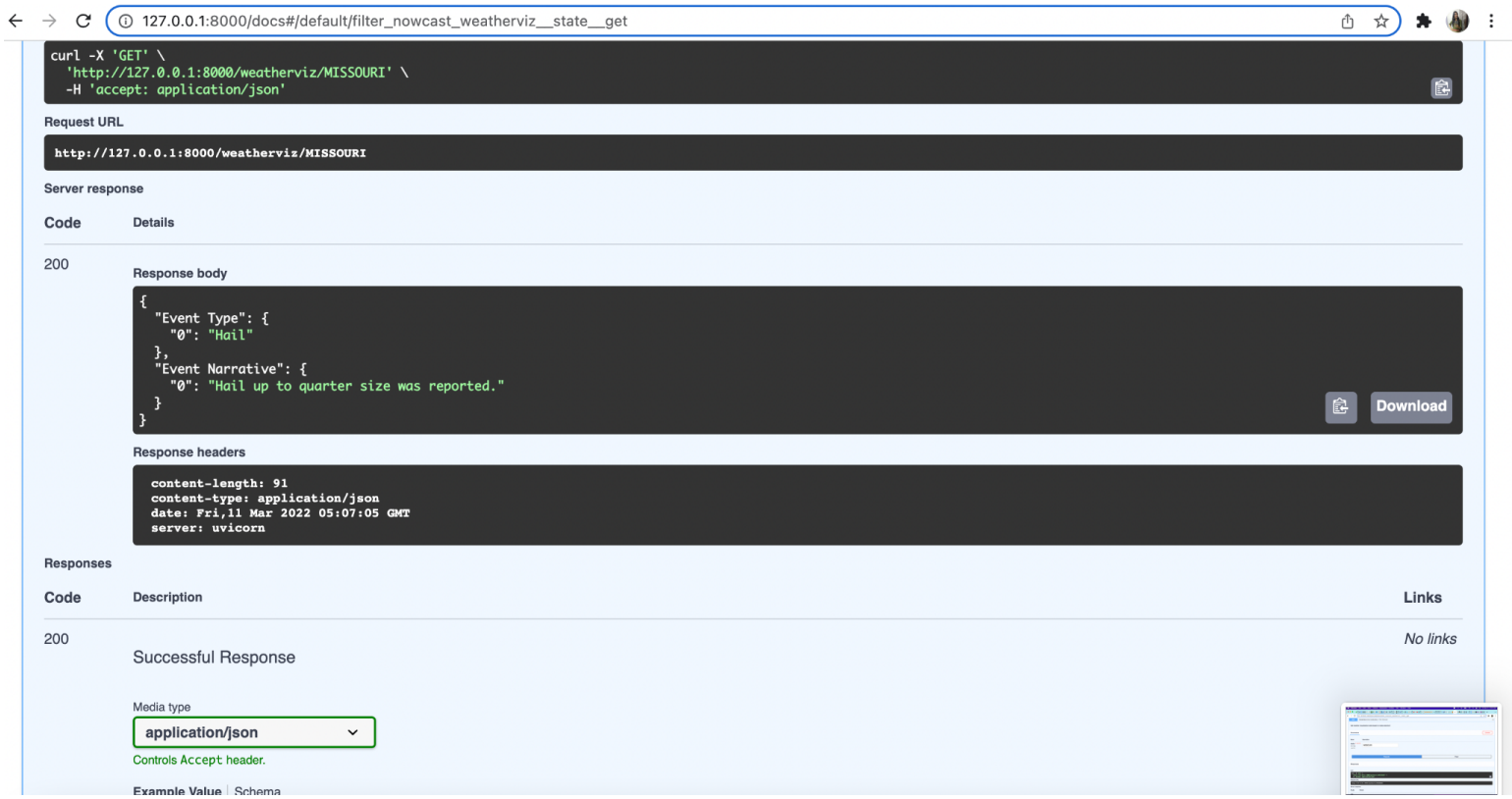
```
curl -X 'GET' \
  'http://127.0.0.1:8000/weatherviz/MISSOURI' \
  -H 'accept: application/json'
```

Request URL

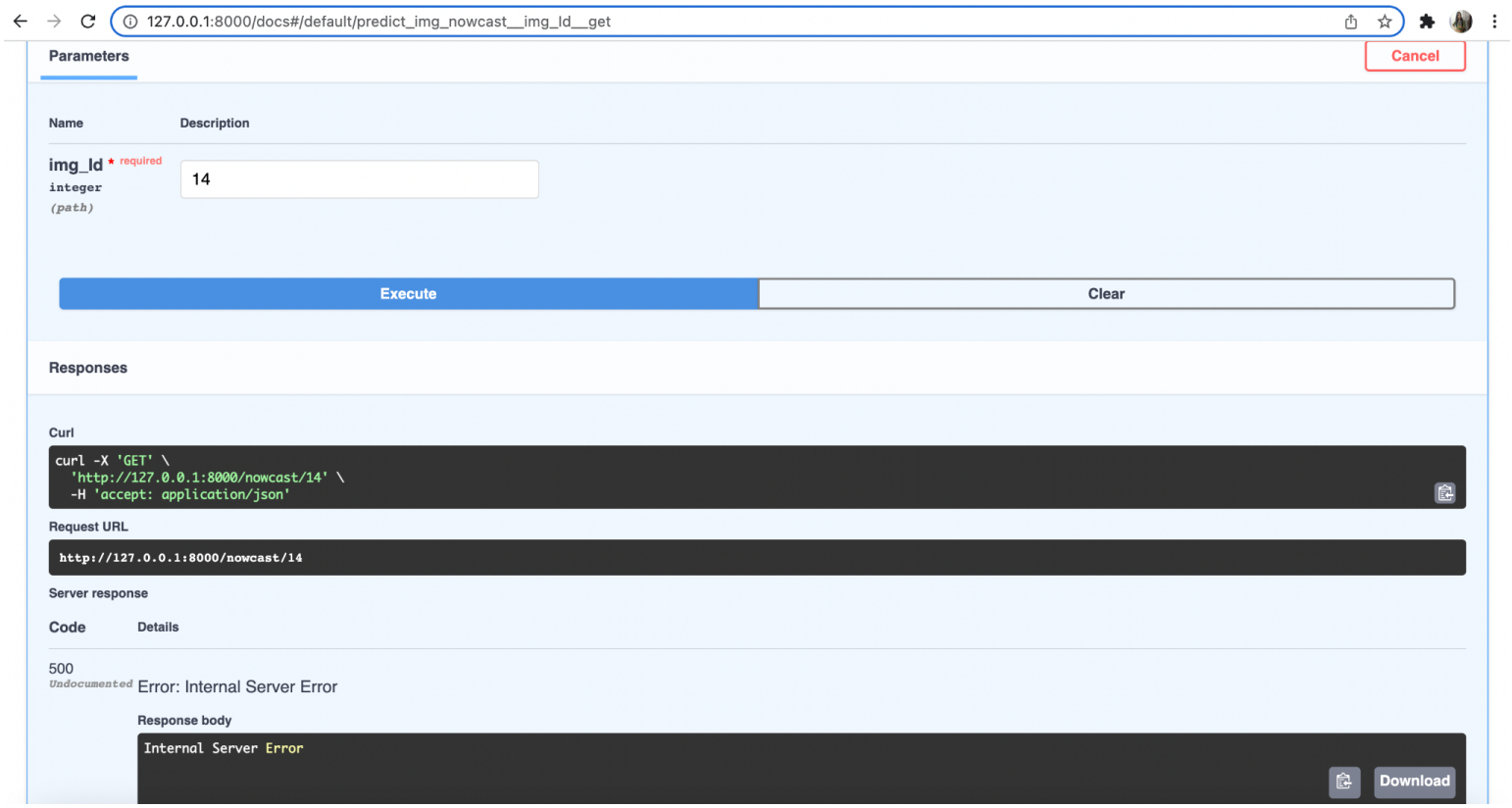
```
http://127.0.0.1:8000/weatherviz/MISSOURI
```

Server response

Code	Details



Test cases to illustrate how to use the API


127.0.0.1:8000/docs#/default/predict_img_nowcast__img_Id__get
Parameters
Cancel
Name
Description
img_Id * required
integer
(path)
14
Execute
Clear
Responses
Curl
curl -X 'GET' \
'http://127.0.0.1:8000/nowcast/14' \
-H 'accept: application/json'
Request URL
http://127.0.0.1:8000/nowcast/14
Server response
Code
Details
500
Undocumented
Error: Internal Server Error
Response body
Internal Server Error
Download

Conclusion

1. We have generated nowcast for the next few hours and rendered 12 nowcast images.
2. Enabled users to filter the event type and event narrative for the selected state from the drop down menu.

References

1. https://realpython.com/fastapi-python-web-apis/#path-parameters-get-an-item-by-id
2. https://learning.postman.com/docs/getting-started/introduction/
3. https://www.youtube.com/watch?v=vpTAqnAbowo
4. https://github.com/gretelai/gretel-blueprints
5. https://nbviewer.org/github/MIT-AI-Accelerator/eie-sevir/blob/master/examples/SEVIR_Tutorial.ipynb
6. https://gretel.ai/blog/walkthrough-create-synthetic-data-from-a-dataframe-or-csv