To handle the HTTP requests and fetch data from the given URLs, we'll use the requests library. We'll also implement a timeout mechanism to ensure that the service respects the specified timeout of 500 milliseconds. First, install Flask and requests by running the following command in your terminal or command prompt.

Here's the implementation:

```
# pip install Flask requests

from flask import Flask, request, jsonify
import requests
import concurrent.futures

app = Flask(__name__)

def fetch_data(url):
    try:
        response = requests.get(url, timeout=0.5)
        if response.status_code == 200:
            return response.json().get("numbers", [])
    except requests.Timeout:
        pass
    except Exception as e:
        print(f"Error fetching data from {url}: {e}")

    return []

@app.route('/numbers')
def get_numbers():
    urls = request.args.getlist('url')

    # Using ThreadPoolExecutor to fetch data from multiple URLs
concurrently
    with concurrent.futures.ThreadPoolExecutor() as executor:
        results = list(executor.map(fetch_data, urls))

    # Merge and sort the integers from all the fetched URLs
    merged_numbers = sorted(set(number for numbers in results for number
in numbers))
```

```
    return jsonify({"numbers": merged_numbers})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8008)
```

Save the file and run the microservice using the following command:
– ***python app.py***

The microservice will start running on http://localhost:8008/numbers.

Now, you can test the microservice with the provided test case using any API testing tool like curl or Postman:
curl -X GET
"http://localhost:8008/numbers?url=http://20.244.56.144/numbers/primes&url=http://20.244.56.144/numbers/fibo&url=http://20.244.56.144/numbers/odd"

ScreenShots