

Prime Generator

Building and Running the Prime Generator

Perform the following steps to build and run:

```
$ mvn clean package
$ java -cp target/primeserver-1.0-jar-with-dependencies.jar
primeserver.Main
```

Example Queries and Responses

Return all primes between 2 and 100

Enter the following URL in your browser:

<http://localhost:1080/primes?max=100>

```
{"data": [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97], "max": 100, "time": 6265538, "status": "OK", "algorithm": "Sieve Of Sundaram"}
```

Return all primes between 2 and 100 using the Trial Division algorithm

Enter the following URL in your browser:

<http://localhost:1080/primes?max=100&algorithm=trialdivision>

```
{"data": [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97], "max": 100, "time": 212519, "status": "OK", "algorithm": "Trial Division"}
```

Return all primes between 2 and 100 using an invalid algorithm (error condition)

Enter the following URL in your browser:

<http://localhost:1080/primes?max=100&algorithm=foo>

```
{"message": "Unknown algorithm: [foo]", "status": "error"}
```

Algorithm Profiles

Profiles of each algorithm. Timings in μ s per number examined.

Max	TrialDivision	SieveOfEratosthenes	SieveOfSundaram
10	0	0	0
100	0	0	0
1000	2	3	1
10000	0.8	0.5	0.3
100000	0.95	0.13	0.13
1000000	5.216	0.068	0.04
10000000	36.5547	0.0841	0.0768