

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": "SieveOfSundaram"}
```

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": "TrialDivision"}
```

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	0.000000	0.000000	0.000000
100	0.000000	0.010000	0.000000
1000	0.002000	0.003000	0.001000
10000	0.000800	0.000500	0.000300
100000	0.000950	0.000130	0.000130
1000000	0.005216	0.000068	0.000040
10000000	0.036555	0.000084	0.000077