# ‘Prime Number’ Rest Service

## 1 About :

This service exposes 2 APIs : ‘isPrime(number)’ and ‘computePrimes(limit)’. For demo purposes, 3 algorithms are implemented to compute the primes. Spring framework has been used. Potentially, more APIs like - computeNthPrime,... can be added.

## 2 Class level details :

1. **com.example.Application** – Main class for startup
2. **com.example.controller.PrimeNumberRangeController** – Controller class that exposes rest service APIs. For demo purpose, it exposes different APIs for different algorithms
3. **com.example.strategy.\*** -

Contains different algorithms to compute prime numbers range  
(a) **BasicPrimeRangeGenStrategy** – Uses parallel streams to iterate over the range and then the isPrime check is done for each number in the range

(b) **ForkJoinAndOptimisedPrimeCheckPrimeRangeGenStrategy** – Uses Forkjoin pool to iterate over the range and optimized isPrime check .  
(c) **EulerSievePrimeRangeGenStrategy** – Implements Euler Sieve algorithm to compute the range

1. **com.example.validate.Validator** – APIs for input validation
2. **com.example.data.Result** – Result object

## 3 How to run :

**3.1 Server :** **3.1.1 IDE :**

Import the project as maven project. Then, right click on Application.java and choose ‘Run As’-> Java Application option.

### 3.1.2 Maven :

Go to base directory of project. One can build the JAR file with mvn clean package and run the JAR by typing:

java -jar target/rest-service-assignment-1.0.jar

**3.2 Client :** Browser can be used to access rest service.   
 Sample URLs :

Euler - <http://localhost:8080/findPrimeEuler?number=100>

Basic - <http://localhost:8080/findPrimeBasic?number=100>

JoinFork - <http://localhost:8080/findPrimeJoinFork?number=100>

Prime check - <http://localhost:8080/isPrime?number=100>