The PrimeCheckApp Case Study: Overview

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Understand how functional programming & the Java Streams framework are applied in a case study using Spring WebMVC to check primality of large integers

See [github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1](https://github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1)
Overview of the Prime CheckApp Case Study
Overview of the PrimeCheckApp Case Study

• This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from sequential and/or parallel clients & servers
Overview of the PrimeCheckApp Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from sequential and/or parallel clients & servers

```java
PrimeCheckTest
```

The client can send requests individually or in bulk, as well as sequentially or in parallel using Java Streams

See WebMVC/ex1/src/main/java/client
Overview of the PrimeCheckApp Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from sequential and/or parallel clients & servers.

The server can receive requests individually or in bulk, as well as process the requests sequentially or in parallel using Java Streams.

See WebMVC/ex1/src/main/java/server
Overview of the PrimeCheckApp Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from sequential and/or parallel clients & servers

The PrimeCheckController automatically converts HTTP GET requests into Java types & forwards them to the PrimeCheckService for processing

See WebMVC/ex1/src/main/java/server/PrimeCheckController.java
Overview of the PrimeCheckApp Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests to/from sequential and/or parallel clients & servers.

The PrimeCheckService checks the primality of one or more Integers passed to it from the controller using either sequential or parallel streams.

See WebMVC/ex1/src/main/java/server/PrimeCheckService.java
Structure of the PrimeCheckApp
The PrimeCheckApp source code is organized into several packages.
The PrimeCheckApp source code is organized into several packages:

**Server**
- Contains the “app” entry point, the controller, & the service implementation.
Structure of the PrimeCheckApp Source Code

- The PrimeCheckApp source code is organized into several packages
  - Server
  - Client
    - Sends HTTP GET requests to the server using either Java sequential or parallel streams
The PrimeCheckApp source code is organized into several packages:

- Server
- Client
- Test driver

Measures the time taken by the client to send/receive requests/responses to/from the server & displays the results.
Structure of the PrimeCheckApp Source Code

- The PrimeCheckApp source code is organized into several packages:
  - Server
  - Client
  - Test driver
  - Common & utils
    - Consolidates various helper classes
End of the PrimeCheck App Case Study: Overview