The PrimeCheckApp Case Study

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 Lesson

• Understand how functional programming & Java parallel streams are applied in a case study that uses Spring WebMVC to check primality of large integers

See 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.
Overview of the PrimeCheckApp Case Study

- This case study shows how Spring WebMVC can be used to send & receive HTTP GET requests
- These requests are mapped to endpoint handler methods that determine the primality of large random integers

```java
@RestController
public class PrimeCheckController {
    @GetMapping(CHECK_IF_PRIME)
    public Integer checkIfPrime(Integer primeCandidate) {...}

    @GetMapping(CHECK_IF_PRIME_LIST)
    public List<Integer> checkIfPrimeList(@RequestParam List<Integer> primeCandidates, Boolean parallel) {...}
}
```
Overview of the PrimeCheckApp Case Study

- This case study uses synchronous two-way calls & the Java parallel streams framework

```java
List<Integer> checkIfPrimeList
    (List<Integer> primeCandidates,
     Boolean parallel) {
    var stream = primeCandidates
        .stream();

    if (parallel)
        stream.parallel();

    return stream
        .map(this::isPrime)
        .collect(toList());
}
```

See www.manning.com/books/modern-java-in-action
Overview of the PrimeCheckApp Case Study

- The source code is organized into several packages
Overview of the PrimeCheckApp Case Study

- The source code is organized into several packages
- Server
• The source code is organized into several packages
  • Server
  • Client
Overview of the PrimeCheckApp Case Study

- The source code is organized into several packages
  - Server
  - Client
  - Test driver
Overview of the PrimeCheckApp Case Study

- The source code is organized into several packages
  - Server
  - Client
  - Test driver
  - Common & utils
Overview of the PrimeCheckApp Case Study

```java
@RestController
@ResponseBody
class PrimeController {
    @Autowired
    PrimeUserService mService;

    @GetMapping("/primeCheck")
    public Integer checkPrime(Integer primeCandidate) {
        // Code to check primality of primeCandidate
    }
}
```

See [github.com/douglasraegschmidt/LiveLessons/tree/master/WebMVC/ex1](https://github.com/douglasraegschmidt/LiveLessons/tree/master/WebMVC/ex1)
End of the PrimeCheck App Case Study