The PrimeCheck App Case Study: Test Driver Implementation & Performance

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 the implementation of a test driver program that runs the PCClient* classes & measures the performance of various methods

PrimeCheckTest

PCServerApplication

HTTP GET requests/responses

PCServer Controller

PCServer Service

PCCF Strategy

PCPS Strategy

PCSC Strategy
Implementing the PrimeCheckTest Driver
Implementing the PrimeCheckTest Driver

```java
package primechecker;

import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

@SpringBootTest(classes = PCsServerApplication.class,
               webEnvironment = SpringBootTest.WebEnvironment.DEFINED_PORT)
@RunWith(SpringRunner.class)
public class PrimeCheckTest {

    /**
     * This object connects [PrimeCheckTest] to the [PrimeCheckClient].
     * The [PrimeCheckClient] annotation ensures
     * this field is initialized via Spring dependency injection,
     * where an object receives another object it depends on (e.g., by
     * creating a [PCClientParallelStream]).
     */

    @Autowired
    private PCClientParallelStream testClientPS;

    @Autowired
    private PCClientCompletableFuture testClientCF;

    @Autowired
    private PCClientStructuredConcurrency testClientSC;
```

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