The PrimeCheckApp Case Study: Implementing the Server Components

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 the PrimeCheckController & PrimeCheckService classes that run in the PrimeCheckApp server.

See [github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1](https://github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1)
Implementing the PrimeCheckApp Server

```java
@RestController
@ResponseBody
public class PrimeCheckController {
    /**
     * This auto-wired field connects the {@link PrimeCheckController} to the
     * PrimeCheckService.
     */
    @Autowired
    PrimeCheckService mService;

    /**
     * Checks the @code primeCandidate} param for primality,
     * returning 0 if it's prime or the smallest factor if it's not.
     * Spring WebMVC maps HTTP GET requests sent to the @code
     * CHECK_IF_PRIME endpoint to this method.
     * @param primeCandidate The @link Integer to check for
     *     primality
     * @return An @link Integer that is 0 if the @code
     *     primeCandidate} is prime and its smallest factor if
     *     it's not prime
     */
    @GetMapping(CHECK_IF_PRIME)
    public Integer checkIfPrime(Integer primeCandidate) {
```
End of the PrimeCheckApp Case Study: Implementing the Server Components