The LockManager App Case Study: Test Driver & Client Implementation

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 LockManagerTest class & associated client code that invoke asynchronous methods on the LockManagerController

See [github.com/douglascraigschmidt/LiveLessons/tree/master/WebFlux/ex1](https://github.com/douglascraigschmidt/LiveLessons/tree/master/WebFlux/ex1)
Implementing the LockManagerTest Driver
Implementing the LockManagerTest Driver

```
/**
 * This program tests the features of the LockManagerApplication microservice, which uses Spring WebFlux to provide a distributed lock manager for asynchronous Spring applications using asynchronous controller methods that return Project Reactor Mono and Flux reactive objects. It also shows how to use the as HTTP interface features in Spring framework 6, which enables the definition of declarative HTTP services using Java interfaces.
 */

@SpringBootTest(classes = LockManagerApplication.class,
               webEnvironment = SpringBootTest.WebEnvironment.DEFINED_PORT)

class LockManagerTests {
    /**
     * The auto-wired LockAPI that accesses the LockManagerApplication microservice.
     */
    @Autowired
```
End of the LockManager App Case Study: Test Driver & Client Implementation