Testing Spring

Beginner to Guru
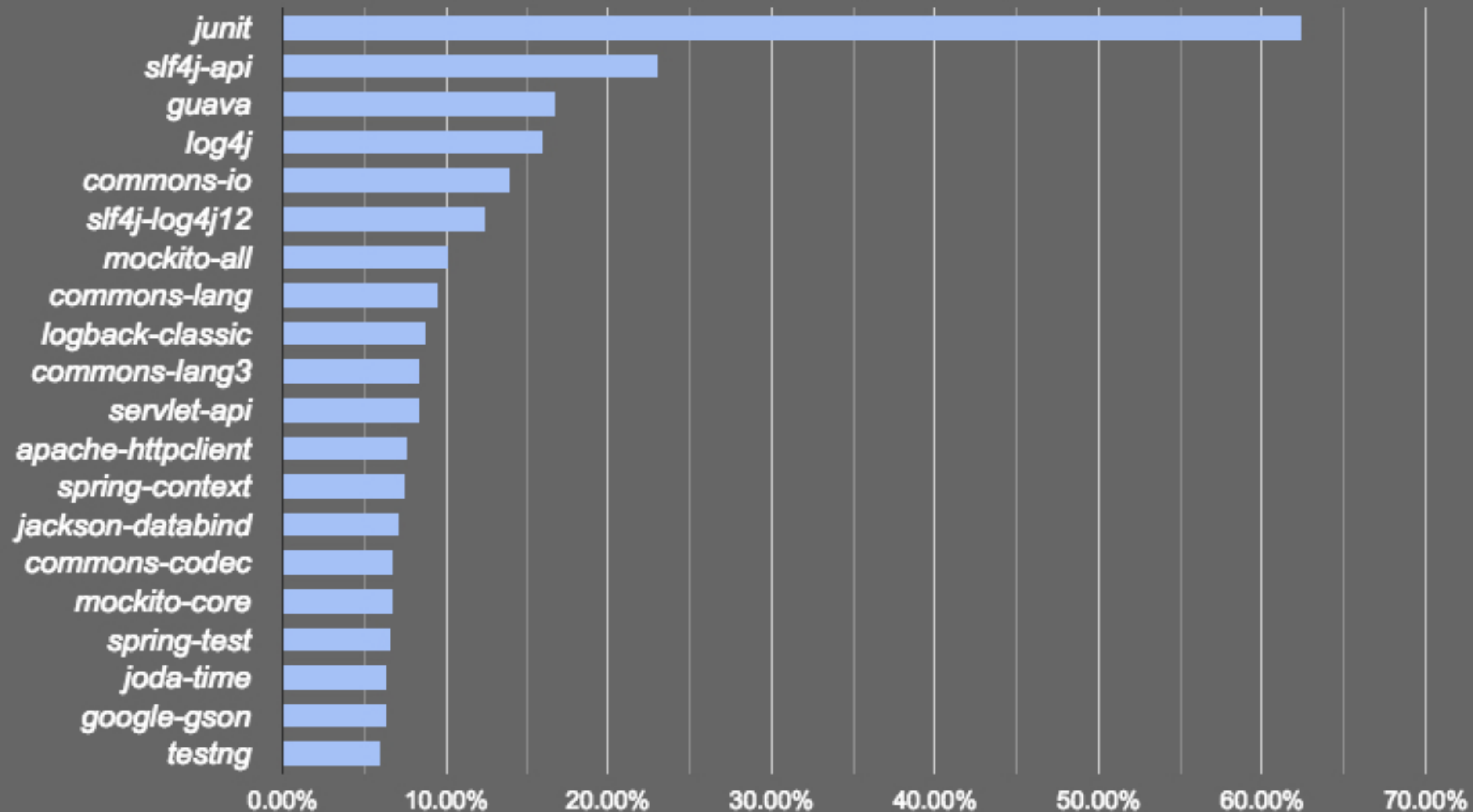
Introduction to JUnit 5

JUnit History

- Kent Beck - known as the creator of Extreme Programming and one of the 17 original signers of the Agile Manifesto and well known for being a proponent of Test Driven Development:
    - "Write new code only if an automated test has failed."
- Kent Beck in the 1990's wrote SUnit, a testing framework for Smalltalk.
- Using SUnit as a starting point Beck partnered with Erich Gamma to write JUnit in 1998
    - Gamma is one of the Gang of Four (aka GoF)
- JUnit has become the most popular Java library in history

Top Java Libraries of 2018

The Top 20 Libraries Used by Github's Most Popular Java Projects

Library	Percentage
junit	~63%
slf4j-api	~23%
guava	~17%
log4j	~16%
commons-io	~14%
slf4j-log4j12	~13%
mockito-all	~10%
commons-lang	~10%
logback-classic	~9%
commons-lang3	~9%
servlet-api	~9%
apache-httpclient	~8%
spring-context	~8%
jackson-databind	~7%
commons-codec	~7%
mockito-core	~7%
spring-test	~7%
joda-time	~7%
google-gson	~7%
testng	~6%

JUnit 4

- JUnit 4 originally released in February 2006

   - Based on Java 5

   - Java 6 Released in December 2006

- ~12 Minor Releases since 2006

- Considered the de factor standard for Java unit testing

- Framework was starting to age considering features added to Java since Java 5

Road to JUnit 5

- Started as a project called JUnit Lambda

  - Crowdfunding campaign in 2015 raised €54,000

  - Corporate Sponsors include Pivotal and American Express

- First GA Release of JUnit 5 was on September 10th, 2017

Goals of JUnit 5

- Leverage features of Java 8

    - Lambda expressions

    - Streams

    - Java 8 or higher is required

- Redesigned for better integration and extensibility

JUnit 5 Modules

- **JUnit Platform** - The foundation for launching testing frameworks on the JVM. Allows tests to be run from a Console Launcher, or build tools such as Maven and Gradle

- **JUnit Jupiter** - Programming model for writing tests and extensions to JUnit

- **JUnit Vintage** - Provides a test engine for running JUnit 3 and JUnit 4 tests.

JUnit Annotations

Annotation	Description
@Test	Marks a method as a test method
@ParameterizedTest	Marks method as a parameterized test
@RepeatedTest	Repeat test N times
@TestFactory	Test Factory method for dynamic tests
@TestInstance	Used to configure test instance lifecycle

JUnit Annotations

Annotation	Description
@TestTemplate	Creates a template to be used by multiple test cases
@DisplayName	Human friendly name for test
@BeforeEach	Method to run before each test case
@AfterEach	Method to run after each test case
@BeforeAll	Static method to run before all test cases in current class

JUnit Annotations

Annotation	Description
@AfterAll	Static method to run after all test cases in current class
@Nested	Creates a nested test class
@Tag	Declare 'tags' for filtering tests
@Disabled	Disable test or test class
@ExtendWith	Used to register extensions

SPRING FRAMEWORK GURU