Spring Professional Exam Tutorial v5.0

Question 07

Question 07 - How does Spring Boot simplify writing tests?

Spring Boot simplifies writing tests in following way:

- Provides `@SpringBootTest` annotation - alternative to `@ContextConfiguration`, creates `ApplicationContext` through `SpringApplication`, Enables Tests Auto-Configuration, Enables Spring Boot Test Features
- Provides `@MockBean` annotation – easy creation and injection of Mockito mock
- Provides `@SpyBean` annotation - easy creation and injection of Mockito spy
- Provides `@WebMvcTest` annotation - useful when test focuses only on Spring MVC components, disables full auto-configuration and applies only configuration relevant to MVC tests
- Provides Web Environments
  - MOCK (default)
  - RANDOM_PORT
  - DEFINED_PORT
  - NONE

Question 07 - How does Spring Boot simplify writing tests?

- Provides algorithm for Tests Environment Auto-Configuration
  - Based on defined dependencies, beans, properties, resources provides beans necessary for integration tests
  - Allows you to focus on test content instead of focusing on how to configure specified technology for integration test

- Allows to explicitly use Auto-Configurations:
  - `@JsonTest` - Auto-configured JSON Tests
  - `@WebMvcTest` - Auto-configured Spring MVC Tests (context limited to MVC)
  - `@JdbcTest` - Auto-configured JDBC Tests
  - `@DataJpaTest` - Auto-configured Data JPA Tests
  - `@JooqTest` - Auto-configured jOOQ Tests
  - `@DataMongoTest` - Auto-configured Data MongoDB Tests
  - `@RestClientTest` - Auto-configured REST Clients
  - ...

Question 07 - How does Spring Boot simplify writing tests?

- Provides `spring-boot-starter-test` module, which includes:
  - JUnit
  - Spring Test
  - Spring Boot Test
  - AssertJ - fluent assertion library.
  - Hamcrest - library of matcher objects
  - Mockito - mocking framework.
  - JSONassert - An assertion library for JSON
  - JsonPath - XPath for JSON

- Provides `@Conditional` annotations
  - `@ConditionalOnClass`
  - `@ConditionalOnMissingBean`
  - `@ConditionalOnProperty`
  - `@ConditionalOnResource`
  - `@ConditionalOnExpression`