Spring Professional Exam Tutorial v5.0

Question 08

Question 08 - What does @SpringBootTest do? How does it interact with @SpringBootApplication and @SpringBootConfiguration?

`@SpringBootTest` annotation provides following features over regular Spring Test Context:

- Automatically searches for `@SpringBootConfiguration`
  - ...unless nested `@Configuration` is detected or explicit `@SpringBootTest(classes=...)` is specified
  - In most cases `@SpringBootConfiguration` is not explicitly used, it is inherited from `@SpringBootApplication` used in production code to indicate starting place for application
  - `@SpringBootConfiguration` is an alternative to standard `@Configuration`, advantage is that `@SpringBootConfiguration` can be found automatically in tests
- Sets default `ContextLoader` to `SpringBootContextLoader`
  - ...unless one is explicitly specified in `@ContextConfiguration(loader=...)`
  - `SpringBootContextLoader` is specific `ContextLoader` that starts tests using `SpringApplication`
- Provides Web Environments
  - MOCK (default)
  - RANDOM_PORT
  - DEFINED_PORT
  - NONE
- Allows to easily define `Environment` properties
  - `properties` field of `@SpringBootTest` annotation can be used to define `key=value` pairs that will be added to `Environment` before tests execution
- Registers `TestRestTemplate` and `WebTestClient`

Question 08 - What does @SpringBootTest do? How does it interact with @SpringBootApplication and @SpringBootConfiguration?

`@SpringBootTest` annotation allows you to set following fields:

- `classes` - annotated classes to use for loading an `ApplicationContext`
  - If not set, `@SpringBootTest` will automatically search for `@SpringBootConfiguration` which is usually inherited from `@SpringBootApplication`
- `properties` - `key=value` pairs that will be added to `Environment` before tests execution
- `webEnvironment` - one of specified web environment used for web layer testing:
  - MOCK (default)
  - RANDOM_PORT
  - DEFINED_PORT
  - NONE

Question 08 - What does @SpringBootTest do? How does it interact with @SpringBootApplication and @SpringBootConfiguration?

`@SpringBootTest` annotation interacts with `@SpringBootApplication` and `@SpringBootConfiguration` through `SpringBootTestContextBootstrapper` and `SpringBootContextLoader`.

Goal of `SpringBootContextLoader` is to transform initial `ContextConfiguration` to `ApplicationContext`. `SpringBootContextLoader` will get as input `class` annotated with `@SpringBootConfiguration`, which will be located by `SpringBootTestContextBootstrapper`.

Algorithm that searches for class annotated with `@SpringBootConfiguration` will start at package where test class is located and will scan this package and all parent packages in search for `@SpringBootApplication`.

Question 08 - What does @SpringBootTest do? How does it interact with @SpringBootApplication and @SpringBootConfiguration?

`@SpringBootTest` annotation is used in following way:

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class ApplicationServiceIntegrationTest {

    ...

}
```

Discovers

```
@SpringBootApplication
public class Runner {

    public static void main(String[] args) {
        SpringApplication.run(Runner.class, args);
    }
}
```

```
...
@SpringBootConfiguration
...
public @interface SpringBootApplication {
...
```

`@SpringBootApplication` definition

`@SpringBootApplication` inherits `@SpringBootConfiguration`