TestContainer 설정 가이드

# 1. PostgreSQL 설정

```java

@Container

static PostgreSQLContainer<?> postgresContainer = new PostgreSQLContainer<>("postgres:13.2-alpine")

.withDatabaseName("testdb")

.withUsername("test")

.withPassword("test");

@DynamicPropertySource

static void registerPgProperties(DynamicPropertyRegistry registry) {

registry.add("spring.datasource.url", postgresContainer::getJdbcUrl);

registry.add("spring.datasource.username", postgresContainer::getUsername);

registry.add("spring.datasource.password", postgresContainer::getPassword);

registry.add("spring.jpa.properties.hibernate.dialect",

() -> "org.hibernate.dialect.PostgreSQLDialect");

}

# 2. MySQL 설정

```

@Container

static MySQLContainer<?> mysqlContainer = new MySQLContainer<>("mysql:8.0")

.withDatabaseName("testdb")

.withUsername("test")

.withPassword("test")

.withCommand(

"--character-set-server=utf8mb4",

"--collation-server=utf8mb4\_unicode\_ci"

);

@DynamicPropertySource

static void registerMySqlProperties(DynamicPropertyRegistry registry) {

registry.add("spring.datasource.url", mysqlContainer::getJdbcUrl);

registry.add("spring.datasource.username", mysqlContainer::getUsername);

registry.add("spring.datasource.password", mysqlContainer::getPassword);

registry.add("spring.jpa.properties.hibernate.dialect",

() -> "org.hibernate.dialect.MySQLDialect");

}

```

# 3. MongoDB 설정

```

@Container

static MongoDBContainer mongoDBContainer = new MongoDBContainer("mongo:6.0")

.withExposedPorts(27017);

@DynamicPropertySource

static void registerMongoProperties(DynamicPropertyRegistry registry) {

registry.add("spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);

}

```

# 4. Redis 설정

```

@Container

static GenericContainer<?> redisContainer = new GenericContainer<>("redis:7.0-alpine")

.withExposedPorts(6379);

@DynamicPropertySource

static void registerRedisProperties(DynamicPropertyRegistry registry) {

registry.add("spring.redis.host", redisContainer::getHost);

registry.add("spring.redis.port", redisContainer::getFirstMappedPort);

}

```

# 5. 여러 DB를 동시에 사용하는 경우 설정

```

@TestConfiguration

@Testcontainers

public class TestContainerConfig {

@Container

static PostgreSQLContainer<?> postgresContainer =

new PostgreSQLContainer<>("postgres:13.2-alpine");

@Container

static MySQLContainer<?> mysqlContainer =

new MySQLContainer<>("mysql:8.0");

@Container

static MongoDBContainer mongoDBContainer =

new MongoDBContainer("mongo:6.0");

@Container

static GenericContainer<?> redisContainer =

new GenericContainer<>("redis:7.0-alpine")

.withExposedPorts(6379);

@DynamicPropertySource

static void registerProperties(DynamicPropertyRegistry registry) {

// PostgreSQL properties

registry.add("spring.datasource.postgres.url",

postgresContainer::getJdbcUrl);

registry.add("spring.datasource.postgres.username",

postgresContainer::getUsername);

registry.add("spring.datasource.postgres.password",

postgresContainer::getPassword);

// MySQL properties

registry.add("spring.datasource.mysql.url",

mysqlContainer::getJdbcUrl);

registry.add("spring.datasource.mysql.username",

mysqlContainer::getUsername);

registry.add("spring.datasource.mysql.password",

mysqlContainer::getPassword);

// MongoDB properties

registry.add("spring.data.mongodb.uri",

mongoDBContainer::getReplicaSetUrl);

// Redis properties

registry.add("spring.redis.host",

redisContainer::getHost);

registry.add("spring.redis.port",

redisContainer::getFirstMappedPort);

}

}

```

# 6. build.gradle 의존성 설정

```

dependencies {

// DB별 TestContainer

testImplementation 'org.testcontainers:postgresql:1.19.3'

testImplementation 'org.testcontainers:mysql:1.19.3'

testImplementation 'org.testcontainers:mongodb:1.19.3'

// Redis 클라이언트

testImplementation 'org.springframework.boot:spring-boot-starter-data-redis'

// TestContainer 공통

testImplementation 'org.testcontainers:testcontainers:1.19.3'

testImplementation 'org.testcontainers:junit-jupiter:1.19.3'

}

```

# 7. DB별 주요 특징과 설정 포인트

- PostgreSQL:

- Alpine 기반 경량 이미지 권장

- 기본 포트: 5432

- 특별한 추가 설정 없이 기본 설정으로 동작

- MySQL:

- 문자셋 설정 필수 (utf8mb4)

- 기본 포트: 3306

- 타임존 설정 필요시 추가

- MongoDB:

- ReplicaSet URL 사용

- 기본 포트: 27017

- 인증 필요시 추가 설정

- Redis:

- Alpine 기반 경량 이미지 권장

- 기본 포트: 6379

- 클러스터 모드 필요시 추가 설정

# 8. 주의사항:

- 컨테이너 버전은 실제 운영 환경과 일치시키기

- 메모리 사용량을 고려한 컨테이너 설정

- 테스트 실행 시간 최적화를 위한 재사용 설정

- 테스트 격리를 위한 데이터 초기화 전략 수립