**Testing Documentation**

This document explains how to run tests, what scenarios are covered, and lists the HTTP APIs with sample requests and responses **matching the current tests**

**1. How to Run**

**Run only API module tests**

| # from repo root ./mvnw -pl api test # Windows .\mvnw.cmd -pl api test |
| --- |

**Run a specific test class/method**

| ./mvnw -pl api test -Dtest=ConcurrentAddStockTests |
| --- |

**Run the app (API module)**

| # Run API (multi-module) from root ./mvnw -pl api -am spring-boot:**run** # Windows .\mvnw.cmd -pl api -am spring-boot:**run** |
| --- |

**Profiles**: tests run with api/src/test/resources/**application-test.properties**. **2. End-to-End Test Scenarios**

All the classes below are located in **api/src/test/java/com/example/api**

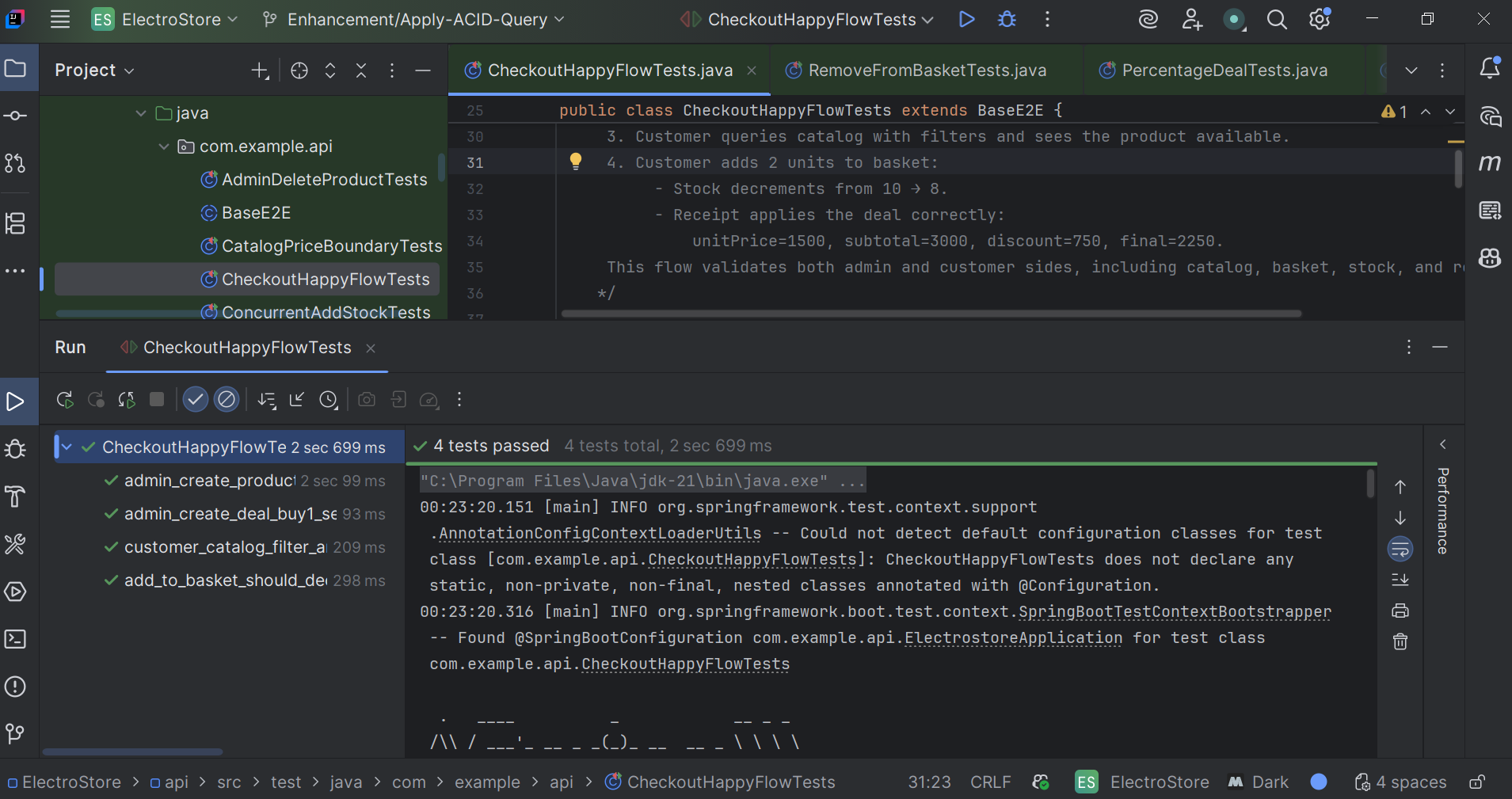
**2.1 CheckoutHappyFlowTests**

**Goal**: End-to-end happy path: create product(s) → add to basket → apply deals → receipt total correct.

**Steps**:

1. Admin creates product (iPhone 15, price=1500, stock=10).
2. Admin creates B1G50\_2ND deal (future expiry).
3. Customer catalogs the product with filters (category=PHONE, price range).
4. Customer adds 2 units to basket → stock decrements to 8.
5. Receipt shows discount math: original=3000, discount=750, final=2250.

**Expect**: All steps 200; catalog & receipt values match.

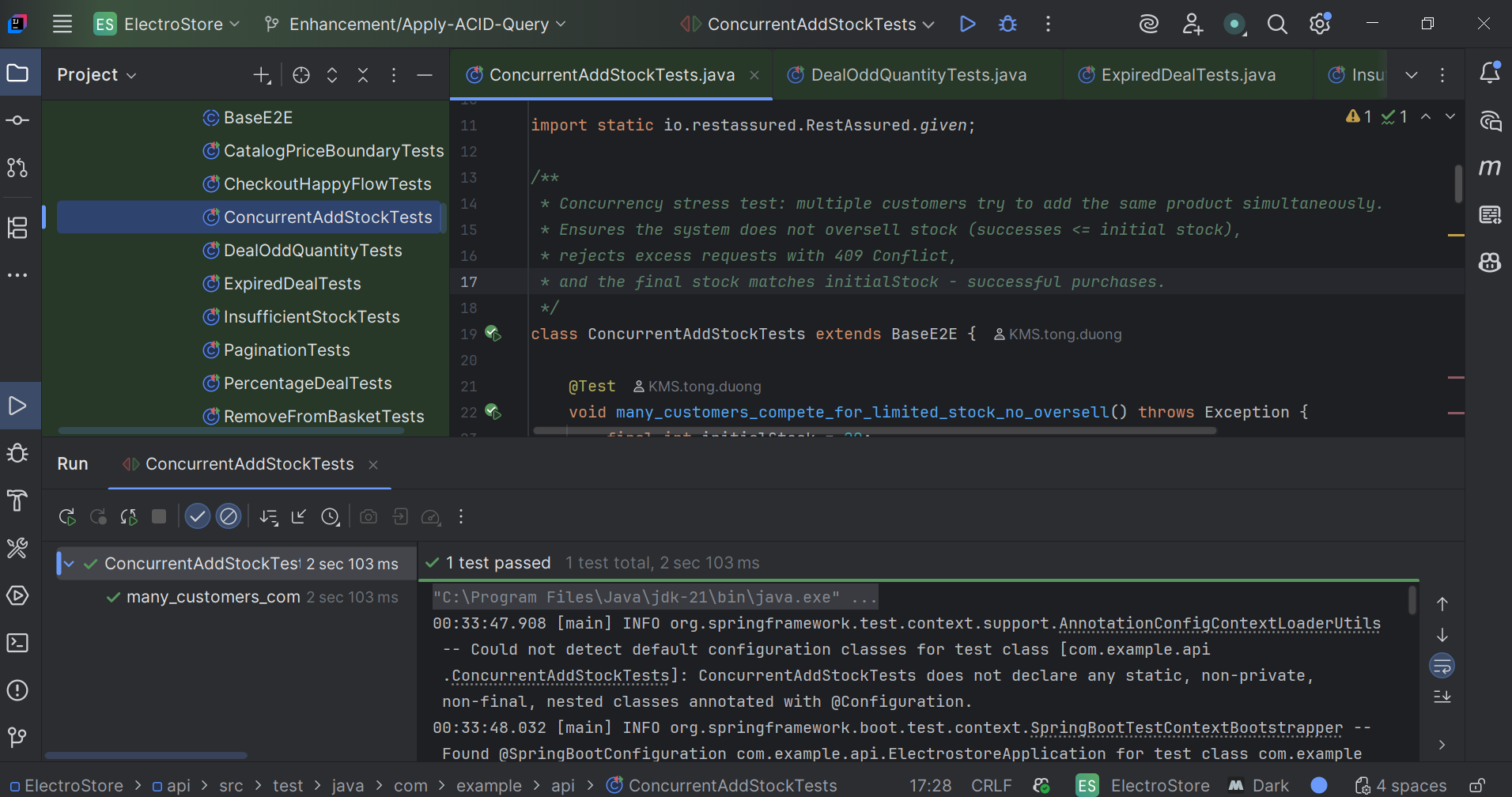


**2.2. ConcurrentAddStockTests**

**Scenario**: Concurrency stress, no oversell.

**Steps**: seed stock=20; 50 concurrent customers add 1 unit; count 200 vs 409; final stock by GET /api/admin/products/{id}.

**Expect**: successes ≤ 20; rest 409; final stock = 20 − successes.

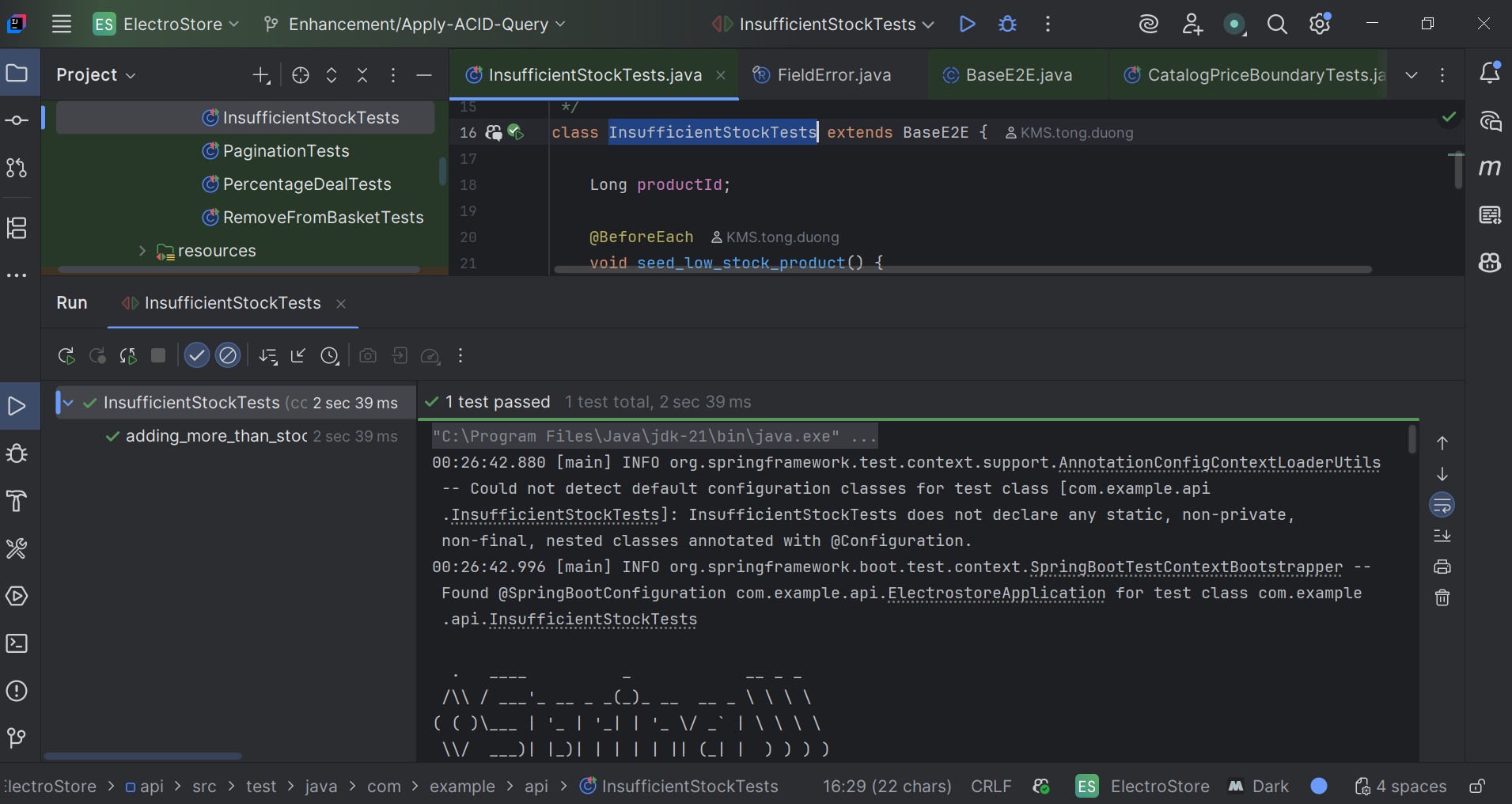


**2.3. InsufficientStockTests**

**Scenario**: Add beyond stock returns 409 and does not change stock.

**Steps**: stock=1; try add 2; then verify stock still 1 via admin list.

**Expect**: 409 on add; stock unchanged.

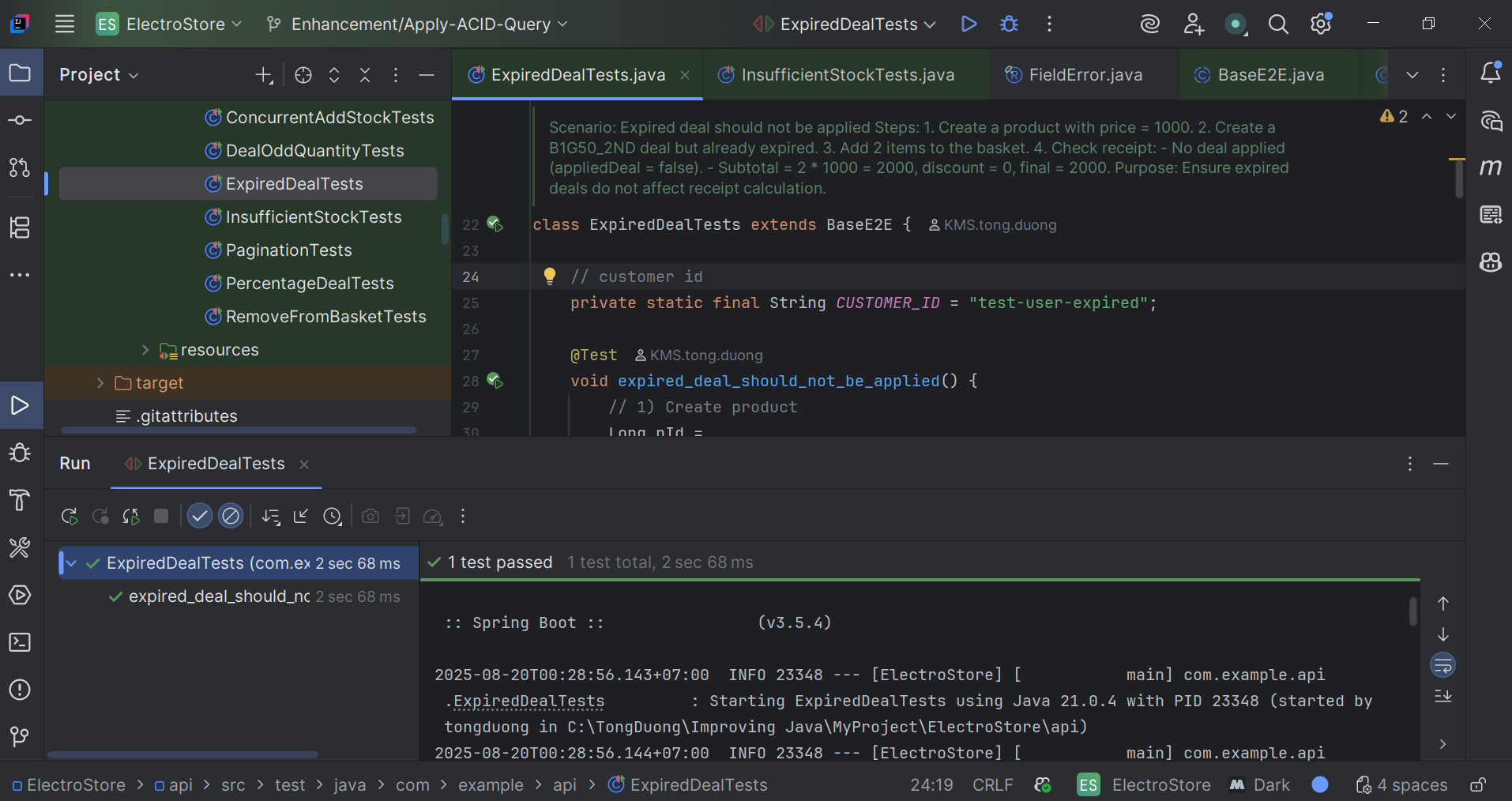


**2.4. ExpiredDealTests**

**Scenario**: Expired deal must not apply.

**Steps**: create product (1000), create expired B1G50\_2ND, add qty=2, get receipt.

**Expect**: no discount; final=2000; appliedDeal = false.



**2.5 PercentageDealTests**

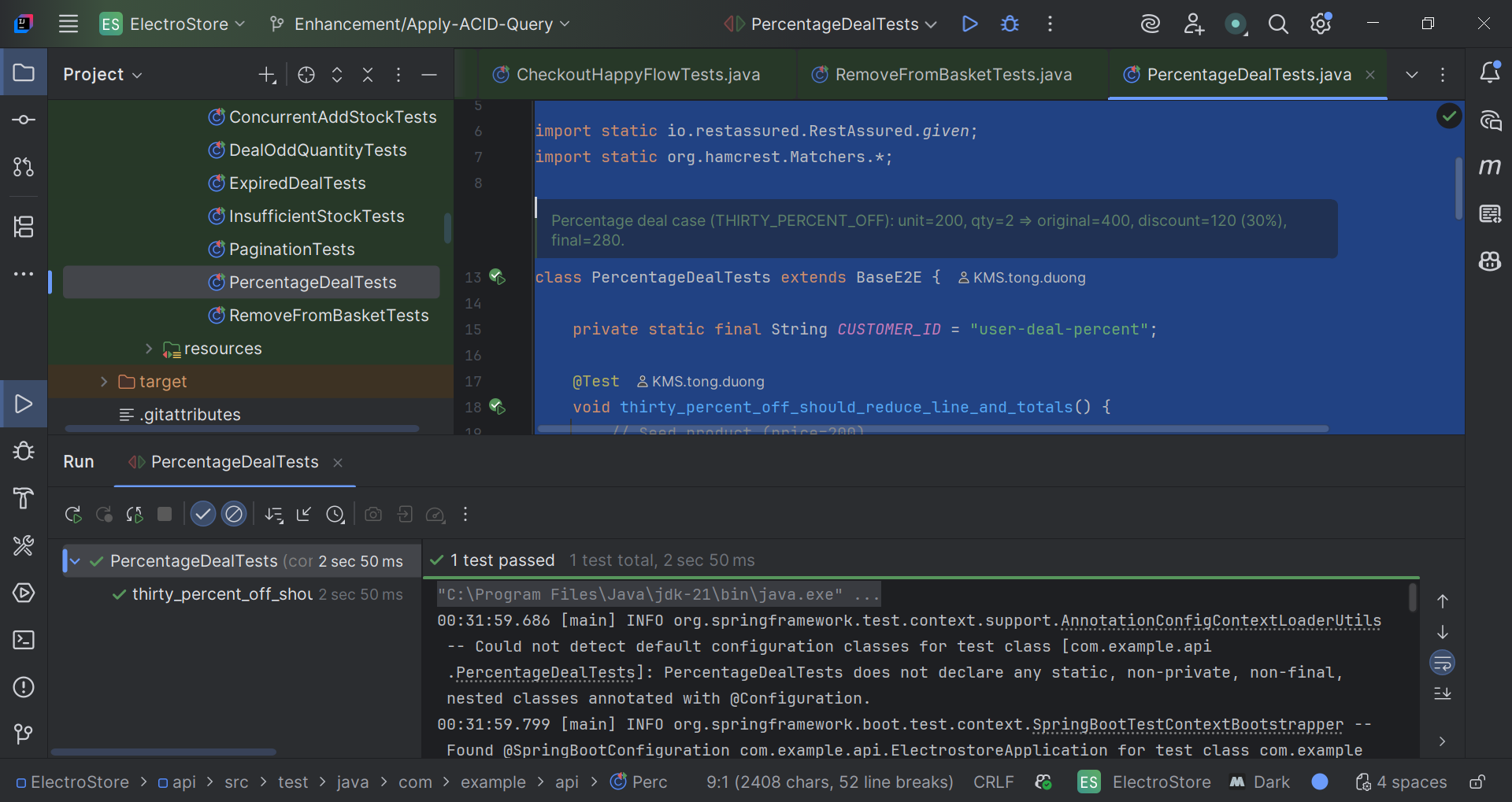
**Scenario**: 30% discount deal is correctly applied.

**Steps**:

1. Create a product (unit price = 200, stock = 10).
2. Apply a future THIRTY\_PERCENT\_OFF deal.
3. Add 2 units to basket (with customer ID).
4. Get receipt.

**Expect**:

* Original subtotal = 400
* Discount = 120 (30%)
* Final = 280
* appliedDeal = true
* Totals match: totalBeforeDiscount = 400, totalDiscount = 120, totalFinal = 280

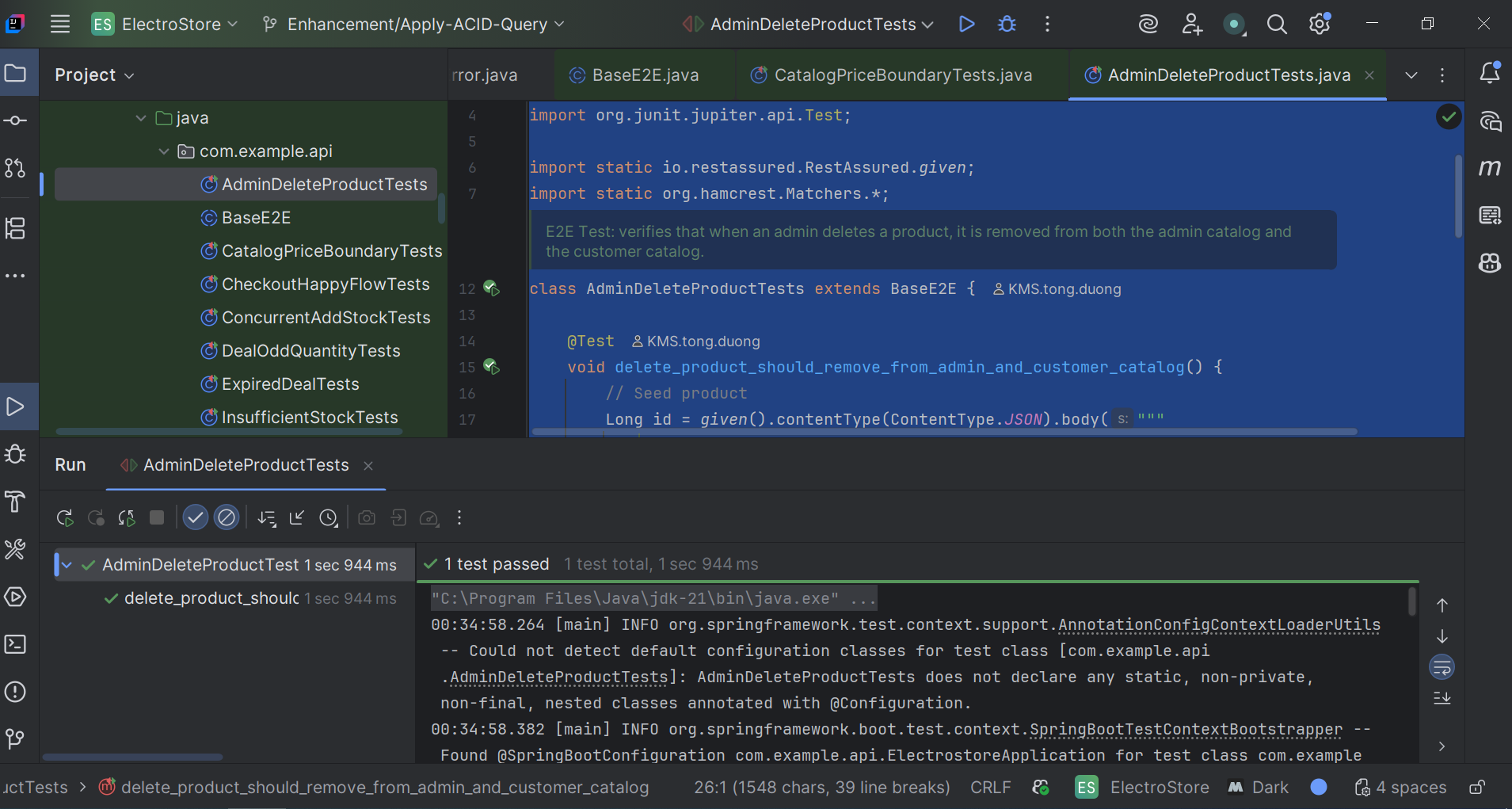


**2.6. AdminDeleteProductTests**

**Scenario**: After admin deletes a product, it no longer appears in both admin and customer catalogs.

**Steps**: create → delete → list admin → list customer.

**Expect**: 200 on delete; product absent in both lists

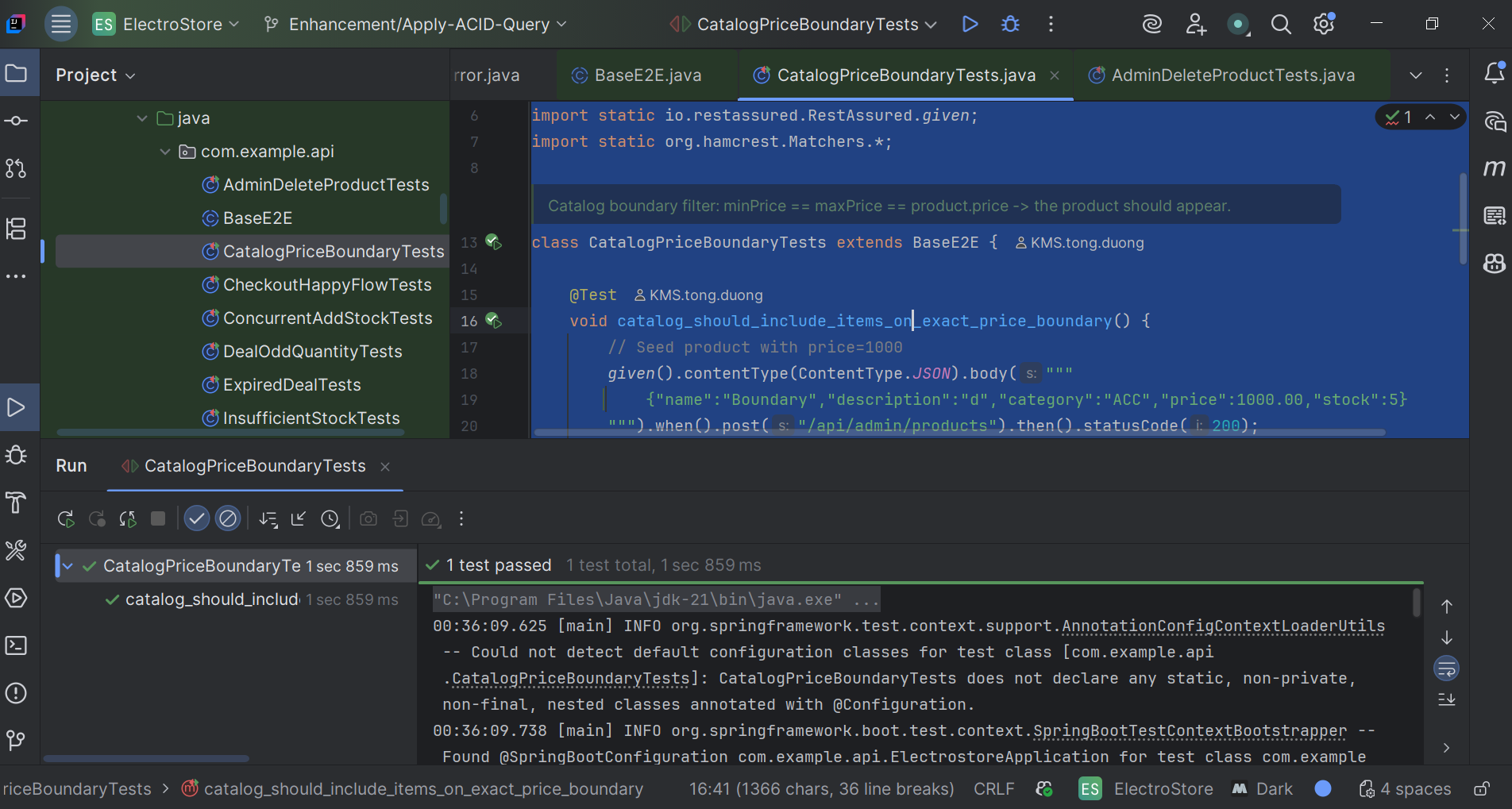


**2.7. CatalogPriceBoundaryTests**

**Scenario**: Boundary filtering on price (minPrice == maxPrice == price).

**Steps**: seed product at 1000; search with min=max=1000.

**Expect**: product appears; page envelope correct; price equals 1000.00.

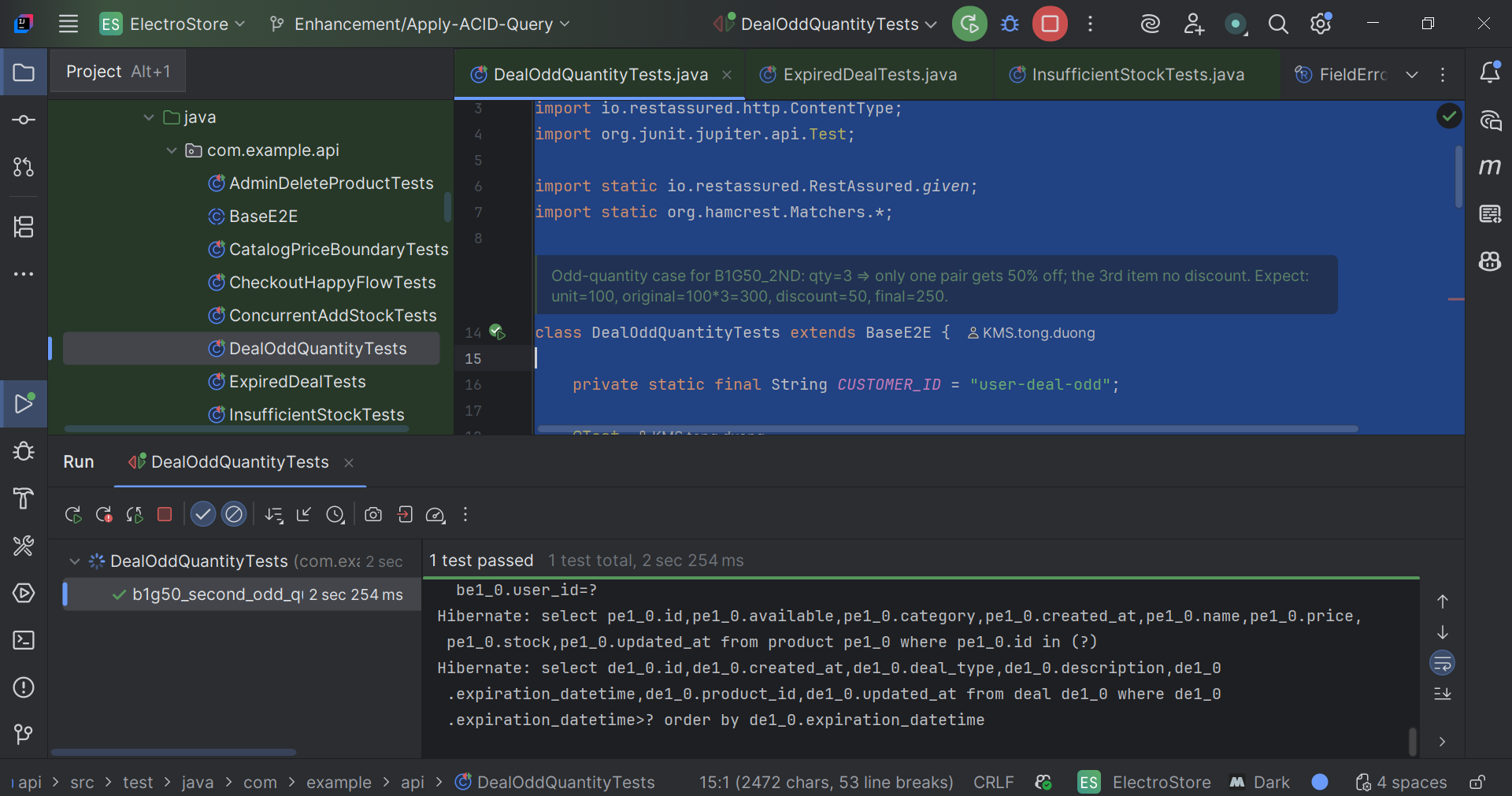


**2.8. DealOddQuantityTests**

**Scenario**: Odd quantity with B1G50\_2ND.

**Steps**: price=100, qty=3; one item gets 50% off.

**Expect**: original=300, discount=50, final=250; appliedDeal = true.



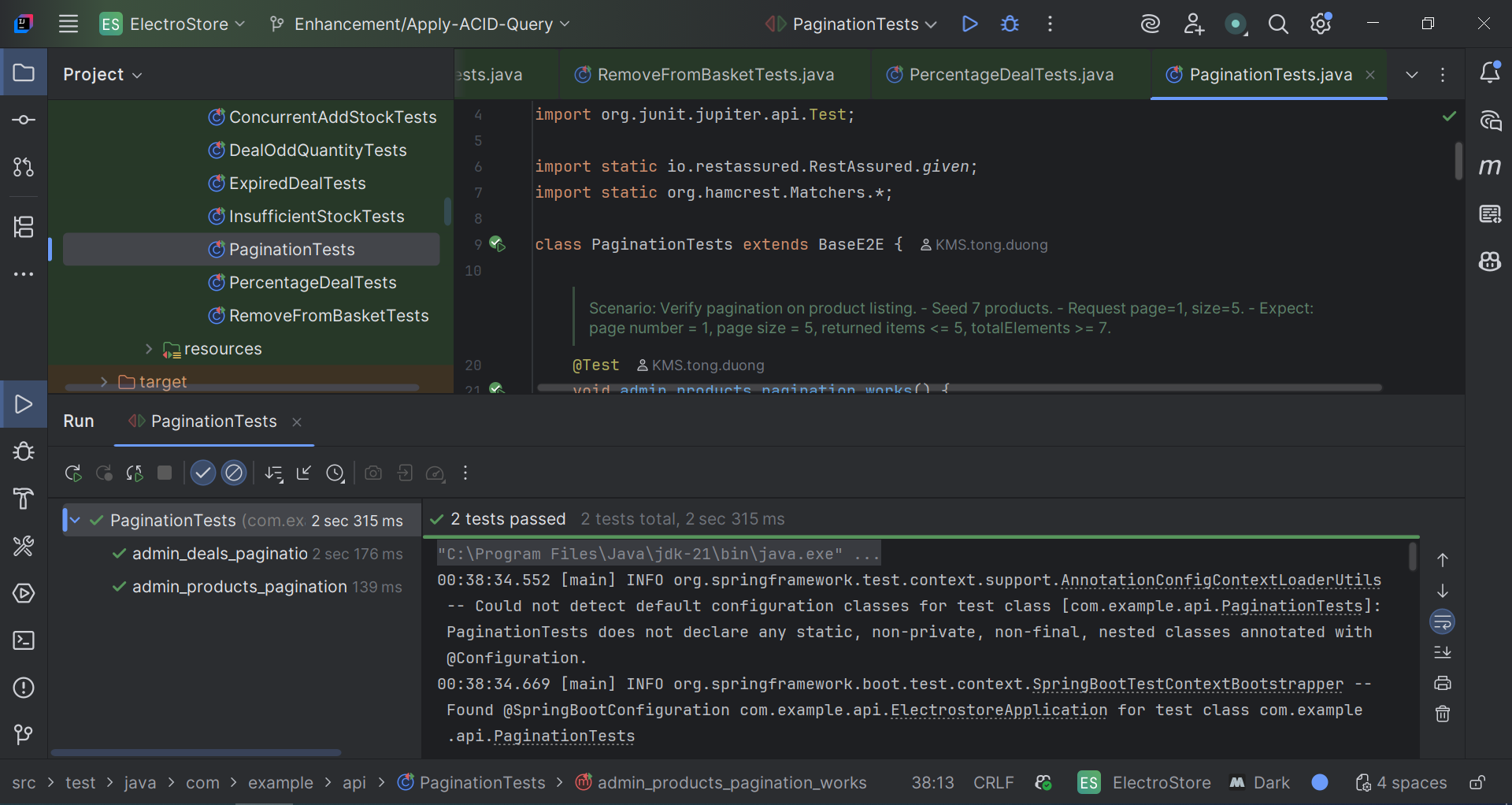
**2.9. PaginationTests**

**Scenario** 1 (products): Seed 7 products; request page=1, size=5.

**Expect**: data.number=1, data.size=5, content.size() ≤ 5, totalElements ≥ 7.

**Scenario** 2 (deals): Seed 6 deals; request page=0, size=5.

**Expect**: data.number=0, data.size=5, content.size() == 5, totalElements ≥ 6.

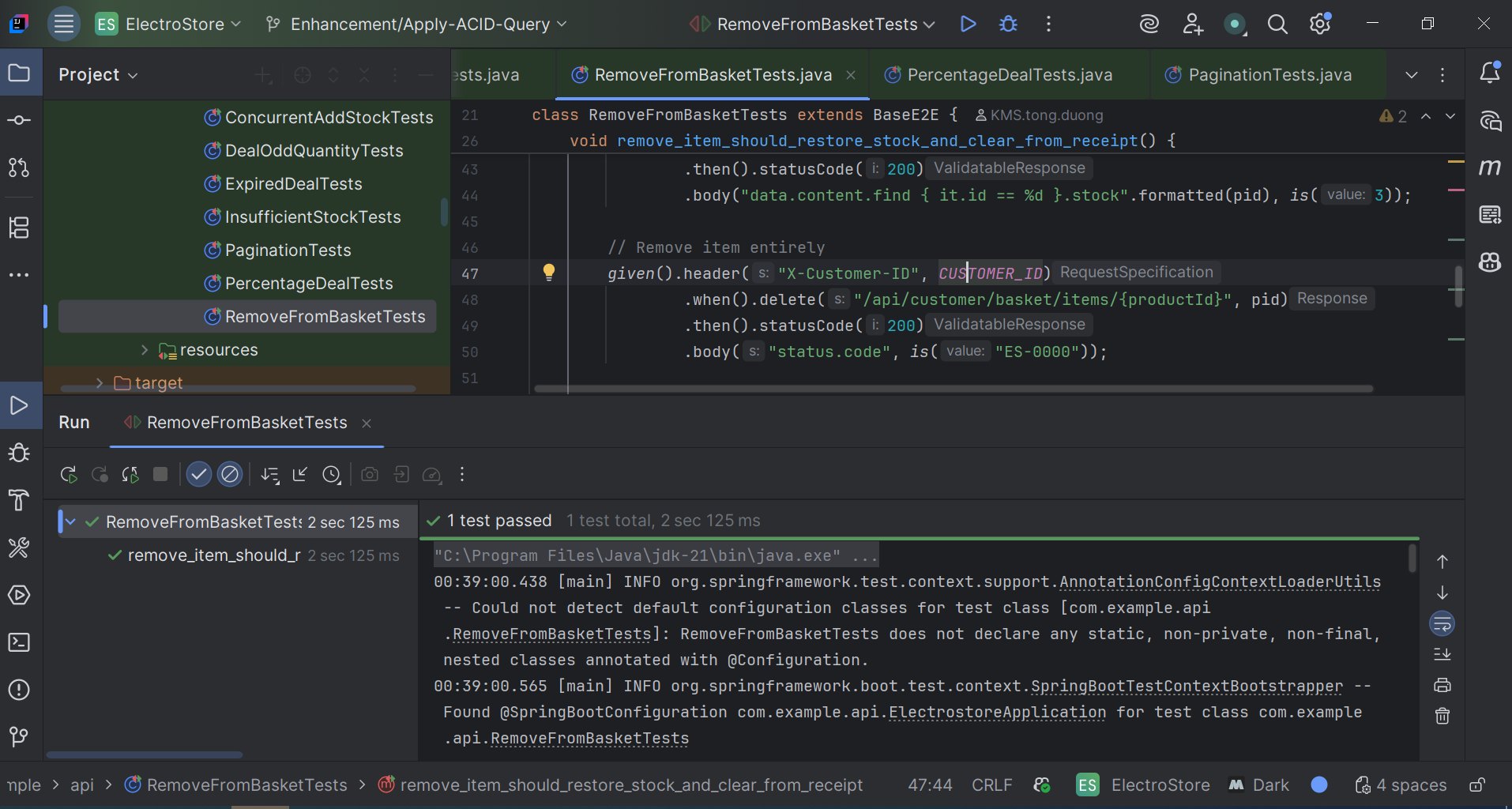


**2.10. RemoveFromBasketTests**

**Scenario**: Removing item restores stock and clears receipt line.

**Steps**: stock=5; add 2 → stock 3; DELETE /api/customer/basket/items/{productId}; check stock back to 5; receipt has no line & total 0.

**Expect**: admin stock shows 3 then 5; receipt empty or total 0.



**3. API List**

**3.1.AdminProductController**

| **Path** | **Description** | **Request** | **Response** |
| --- | --- | --- | --- |
| POST /api/admin/products | Create product | {  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "description": "A sample mouse",  "stock": 10  } | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "id": 1,  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "stock": 10,  "available": **true**  },  "metaData": {  "timestamp": 1755611962681  }  } |
| GET /api/admin/products?page=1&size=5&name=iPhone&category=PHONE&minPrice=500&maxPrice=1000 | Get product with filter | http://localhost:8080/api/admin/products?page=0&size=5&name=mouse&category=Electronics&minPrice=0&maxPrice=10 | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "content": [  {  "id": 1,  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "stock": 10,  "available": **true**  }  ],  "number": 0,  "size": 5,  "totalElements": 1  },  "metaData": {  "timestamp": 1755625612551  }  } |
| DELETE /api/admin/products/{productId} | Delete product by product id | http://localhost:8080/api/admin/products/1 | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": **null**,  "metaData": {  "timestamp": 1755612114066  }  } |

**3.2. AdminDealController**

| **Path** | **Description** | **Request** | **Response** |
| --- | --- | --- | --- |
| POST /api/admin/deals | Create deal | {  "productId": 1,  "dealType": "THIRTY\_PERCENT\_OFF",  "expirationDateTime": "2025-08-31T23:59:59"  } | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "id": 1,  "productId": 1,  "dealType": "THIRTY\_PERCENT\_OFF",  "expirationDateTime": "2025-08-31T23:59:59"  },  "metaData": {  "timestamp": 1755625826581  }  } |
| GET /api/admin/deals?page=0&size=10 | List all deals with pagination | http://localhost:8080/api/admin/deals?page=0&size=10 | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "content": [  {  "id": 1,  "productId": 1,  "dealType": "THIRTY\_PERCENT\_OFF",  "expirationDateTime": "2025-08-31T23:59:59"  },  {  "id": 3,  "productId": 2,  "dealType": "B1G50\_2ND",  "expirationDateTime": "2025-08-31T23:59:59"  }  ],  "number": 0,  "size": 10,  "totalElements": 2  },  "metaData": {  "timestamp": 1755625958963  }  } |
| DELETE /api/admin/deals/{dealId} | Delete deal by deal id | http://localhost:8080/api/admin/deals/2 | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": **null**,  "metaData": {  "timestamp": 1755625905343  }  } |

**3.3. BasketController**

| **Path** | **Description** | **Request** | **Response** |
| --- | --- | --- | --- |
| POST /api/customer/basket/items | Add item to basket | **Header**  **X-Customer-ID**: <value>  {  "productId": 2,  "quantity": 2  } | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": **null**,  "metaData": {  "timestamp": 1755626063492  }  } |
| GET /api/customer/basket | View current basket | <http://localhost:8080/api/customer/basket>  **Header**  **X-Customer-ID**: <value> | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "items": [  {  "productId": 1,  "name": "mouse",  "quantity": 1,  "unitPrice": 3.99  },  {  "productId": 2,  "name": "mouse",  "quantity": 2,  "unitPrice": 3.99  }  ],  "totalAmount": 11.97  },  "metaData": {  "timestamp": 1755626109895  }  } |
| DELETE api/customer/basket/items/{productId} | Remove from current basket | <http://localhost:8080/api/customer/basket/items/2>  **Header**  **X-Customer-ID**: <value> | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": **null**,  "metaData": {  "timestamp": 1755626236086  }  } |

**3.4. CustomerCatalogController**

| **Path** | **Description** | **Request** | **Response** |
| --- | --- | --- | --- |
| GET /api/customer/products?name=mouse&category=&minPrice=0&maxPrice=30&available=true&page=0&size=10 | Get product catalog for customer | **Header**  **X-Customer-ID**: <value> | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "content": [  {  "id": 4,  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "available": **true**  },  {  "id": 2,  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "available": **true**  },  {  "id": 1,  "name": "mouse",  "category": "Electronics",  "price": 3.99,  "available": **true**  }  ],  "number": 0,  "size": 10,  "totalElements": 3  },  "metaData": {  "timestamp": 1755626399930  }  } |

**3.5. ReceiptController**

| **Path** | **Description** | **Request** | **Response** |
| --- | --- | --- | --- |
| GET /api/customer/receipt | Generate receipt for current basket | **Header**  **X-Customer-ID**: <value> | {  "status": {  "code": "ES-0000",  "message": "Success"  },  "data": {  "items": [  {  "productId": 1,  "productName": "mouse",  "quantity": 1,  "unitPrice": 3.99,  "originalSubtotal": 3.99,  "discount": 1.20,  "finalSubtotal": 2.79,  "appliedDeal": **true**  },  {  "productId": 2,  "productName": "mouse",  "quantity": 2,  "unitPrice": 3.99,  "originalSubtotal": 7.98,  "discount": 2.00,  "finalSubtotal": 5.98,  "appliedDeal": **true**  }  ],  "totalBeforeDiscount": 11.97,  "totalDiscount": 3.20, |