

# Test Cases Catalog

Total test cases: 114

## Backend — Access Logs API (13)

### **TestAccessLogEndpoints.test\_get\_access\_logs\_exceeds\_max\_limit**

(backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs exceeding maximum limit
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_filter\_by\_date\_range**

(backend/tests/test\_access\_log\_endpoints.py)

- Test filtering access logs by date range
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_filter\_by\_endpoint**

(backend/tests/test\_access\_log\_endpoints.py)

- Test filtering access logs by endpoint
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_filter\_by\_operation\_type**

(backend/tests/test\_access\_log\_endpoints.py)

- Test filtering access logs by operation type
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_filter\_by\_user\_id**

(backend/tests/test\_access\_log\_endpoints.py)

- Test filtering access logs by user ID
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_invalid\_date\_format**

(backend/tests/test\_access\_log\_endpoints.py)

- Test filtering with invalid date format
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_statistics** (backend/tests/test\_access\_log\_endpoints.py)

- Test getting access log statistics
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_statistics\_unauthorized**

(backend/tests/test\_access\_log\_endpoints.py)

- Test getting statistics without authentication
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_success** (backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs with authentication
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_unauthorized** (backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs without authentication
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_with\_limit** (backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs with limit parameter
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_with\_max\_limit**

(backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs with maximum limit
- Test suite for /api/access-logs/\* endpoints

### **TestAccessLogEndpoints.test\_get\_access\_logs\_with\_offset** (backend/tests/test\_access\_log\_endpoints.py)

- Test getting access logs with offset parameter
- Test suite for /api/access-logs/\* endpoints

## **Backend — Auth API (14)**

### **TestAuthEndpoints.test\_activate\_account\_invalid\_token** (backend/tests/test\_auth\_endpoints.py)

- Test account activation with invalid token
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_change\_password\_success** (backend/tests/test\_auth\_endpoints.py)

- Test successful password change
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_change\_password\_unauthorized** (backend/tests/test\_auth\_endpoints.py)

- Test password change without authentication
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_change\_password\_wrong\_current** (backend/tests/test\_auth\_endpoints.py)

- Test password change with wrong current password
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_get\_current\_user\_invalid\_token** (backend/tests/test\_auth\_endpoints.py)

- Test getting current user with invalid token
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_get\_current\_user\_success** (backend/tests/test\_auth\_endpoints.py)

- Test getting current user info with valid token
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_get\_current\_user\_unauthorized** (backend/tests/test\_auth\_endpoints.py)

- Test getting current user without token
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_login\_invalid\_credentials** (backend/tests/test\_auth\_endpoints.py)

- Test login with invalid password
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_login\_nonexistent\_user** (backend/tests/test\_auth\_endpoints.py)

- Test login with non-existent user
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_login\_success** (backend/tests/test\_auth\_endpoints.py)

- Test successful login
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_register\_duplicate\_email** (backend/tests/test\_auth\_endpoints.py)

- Test registration with duplicate email
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_register\_invalid\_email** (backend/tests/test\_auth\_endpoints.py)

- Test registration with invalid email format
- Test suite for /api/auth/\* endpoints

### **TestAuthEndpoints.test\_register\_missing\_fields** (backend/tests/test\_auth\_endpoints.py)

- Test registration with missing required fields
- Test suite for /api/auth/\* endpoints

**TestAuthEndpoints.test\_register\_success** (backend/tests/test\_auth\_endpoints.py)

- Test successful user registration
- Test suite for /api/auth/\* endpoints

## Backend — End-to-end Integration Flows (17)

**TestAuthenticationIntegration.test\_authentication\_required\_flow** (backend/tests/test\_integration\_auth\_flow.py)

- Integration test: Verify authentication required for protected endpoints
- Steps:

**TestAuthenticationIntegration.test\_complete\_registration\_and\_login\_flow**

(backend/tests/test\_integration\_auth\_flow.py)

- Integration test: Complete registration and login flow
- Steps:

**TestAuthenticationIntegration.test\_login\_and\_access\_wardrobe\_flow**

(backend/tests/test\_integration\_auth\_flow.py)

- Integration test: Login and access wardrobe flow
- Steps:

**TestCompleteUserJourney.test\_complete\_journey\_search\_pagination\_and\_suggestion**

(backend/tests/test\_integration\_complete\_user\_journey.py)

- Integration test: Complete journey with search, pagination, and suggestion
- Steps:

**TestCompleteUserJourney.test\_complete\_journey\_wardrobe\_to\_outfit\_to\_history**

(backend/tests/test\_integration\_complete\_user\_journey.py)

- Integration test: Complete journey from wardrobe to outfit to history
- Steps:

**TestCompleteUserJourney.test\_complete\_journey\_with\_wardrobe\_matching**

(backend/tests/test\_integration\_complete\_user\_journey.py)

- Integration test: Complete journey with wardrobe matching
- Steps:

**TestOutfitSuggestionIntegration.test\_complete\_flow\_cost\_calculation**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Verify cost calculation in complete flow
- Steps:

**TestOutfitSuggestionIntegration.test\_complete\_flow\_duplicate\_detection\_workflow**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Complete duplicate detection workflow
- Steps:

**TestOutfitSuggestionIntegration.test\_complete\_flow\_from\_wardrobe\_item**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Complete flow starting from wardrobe item
- Steps:

**TestOutfitSuggestionIntegration.test\_complete\_flow\_with\_duplicate\_get\_new\_suggestion**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Complete flow with duplicate detection but get new AI suggestion
- Steps:

### **TestOutfitSuggestionIntegration.test\_complete\_flow\_with\_duplicate\_use\_existing**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Complete flow with duplicate detection and using existing suggestion
- Steps:

### **TestOutfitSuggestionIntegration.test\_complete\_flow\_with\_model\_image\_generation**

(backend/tests/test\_integration\_outfit\_flow.py)

- Integration test: Complete flow with model image generation enabled
- Steps:

### **TestWardrobeManagementIntegration.test\_complete\_wardrobe\_management\_flow**

(backend/tests/test\_integration\_wardrobe\_flow.py)

- Integration test: Complete wardrobe management flow
- Steps:

### **TestWardrobeManagementIntegration.test\_wardrobe\_duplicate\_detection\_flow**

(backend/tests/test\_integration\_wardrobe\_flow.py)

- Integration test: Wardrobe duplicate detection flow
- Steps:

### **TestWardrobeManagementIntegration.test\_wardrobe\_pagination\_integration**

(backend/tests/test\_integration\_wardrobe\_flow.py)

- Integration test: Wardrobe pagination integration
- Steps:

### **TestWardrobeManagementIntegration.test\_wardrobe\_search\_and\_filter\_integration**

(backend/tests/test\_integration\_wardrobe\_flow.py)

- Integration test: Wardrobe search and filter integration
- Steps:

### **TestWardrobeManagementIntegration.test\_wardrobe\_summary\_integration**

(backend/tests/test\_integration\_wardrobe\_flow.py)

- Integration test: Wardrobe summary integration
- Steps:

## **Backend — Outfit API (18)**

### **TestOutfitEndpoints.test\_check\_duplicate\_authenticated** (backend/tests/test\_outfit\_endpoints.py)

- Test duplicate check with authentication
- Test suite for /api/suggest-outfit and related endpoints

### **TestOutfitEndpoints.test\_check\_duplicate\_missing\_image** (backend/tests/test\_outfit\_endpoints.py)

- Test duplicate check without image
- Test suite for /api/suggest-outfit and related endpoints

### **TestOutfitEndpoints.test\_check\_duplicate\_no\_auth** (backend/tests/test\_outfit\_endpoints.py)

- Test duplicate check without authentication (should work for anonymous)
- Test suite for /api/suggest-outfit and related endpoints

### **TestOutfitEndpoints.test\_delete\_outfit\_history\_authenticated** (backend/tests/test\_outfit\_endpoints.py)

- Test deleting outfit history entry
- Test suite for /api/suggest-outfit and related endpoints

### **TestOutfitEndpoints.test\_delete\_outfit\_history\_nonexistent** (backend/tests/test\_outfit\_endpoints.py)

- Test deleting non-existent outfit history entry
- Test suite for /api/suggest-outfit and related endpoints

### **TestOutfitEndpoints.test\_delete\_outfit\_history\_unauthorized** (backend/tests/test\_outfit\_endpoints.py)

- Test deleting outfit history without authentication

- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_get\_outfit\_history\_authenticated** (backend/tests/test\_outfit\_endpoints.py)

- Test getting outfit history with authentication
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_get\_outfit\_history\_unauthorized** (backend/tests/test\_outfit\_endpoints.py)

- Test getting outfit history without authentication
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_get\_outfit\_history\_with\_limit** (backend/tests/test\_outfit\_endpoints.py)

- Test getting outfit history with limit parameter
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_health\_check** (backend/tests/test\_outfit\_endpoints.py)

- Test health check endpoint
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_root\_endpoint** (backend/tests/test\_outfit\_endpoints.py)

- Test root endpoint
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_authenticated** (backend/tests/test\_outfit\_endpoints.py)

- Test outfit suggestion with authentication
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_from\_wardrobe\_item\_authenticated**

(backend/tests/test\_outfit\_endpoints.py)

- Test suggesting outfit from wardrobe item with authentication
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_from\_wardrobe\_item\_nonexistent**

(backend/tests/test\_outfit\_endpoints.py)

- Test suggesting outfit from non-existent wardrobe item
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_from\_wardrobe\_item\_unauthorized**

(backend/tests/test\_outfit\_endpoints.py)

- Test suggesting outfit from wardrobe item without authentication
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_missing\_image** (backend/tests/test\_outfit\_endpoints.py)

- Test outfit suggestion without image
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_unauthorized** (backend/tests/test\_outfit\_endpoints.py)

- Test outfit suggestion without authentication (should work for anonymous)
- Test suite for /api/suggest-outfit and related endpoints

**TestOutfitEndpoints.test\_suggest\_outfit\_with\_model\_image** (backend/tests/test\_outfit\_endpoints.py)

- Test outfit suggestion with model image generation
- Test suite for /api/suggest-outfit and related endpoints

## Backend — Outfit ↔ Wardrobe Integration (4)

**TestOutfitWardrobeIntegration.test\_suggest\_outfit\_from\_wardrobe\_item\_with\_matching**

(backend/tests/test\_outfit\_wardrobe\_integration.py)

- Test outfit suggestion from wardrobe item includes matching items
- Test suite for outfit suggestions with wardrobe matching

### **TestOutfitWardrobeIntegration.test\_suggest\_outfit\_handles\_empty\_wardrobe**

(backend/tests/test\_outfit\_wardrobe\_integration.py)

- Test outfit suggestion works with empty wardrobe
- Test suite for outfit suggestions with wardrobe matching

### **TestOutfitWardrobeIntegration.test\_suggest\_outfit\_response\_format**

(backend/tests/test\_outfit\_wardrobe\_integration.py)

- Test that outfit suggestion response has correct format
- Test suite for outfit suggestions with wardrobe matching

### **TestOutfitWardrobeIntegration.test\_suggest\_outfit\_with\_wardrobe\_matching**

(backend/tests/test\_outfit\_wardrobe\_integration.py)

- Test outfit suggestion includes matching wardrobe items
- Test suite for outfit suggestions with wardrobe matching

## **Backend — Wardrobe API (24)**

### **TestWardrobeEndpoints.test\_add\_wardrobe\_item\_missing\_fields** (backend/tests/test\_wardrobe\_endpoints.py)

- Test adding wardrobe item with missing required fields
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_add\_wardrobe\_item\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test successfully adding a wardrobe item
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_add\_wardrobe\_item\_unauthorized** (backend/tests/test\_wardrobe\_endpoints.py)

- Test adding wardrobe item without authentication
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_analyze\_wardrobe\_image\_missing\_image**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test analyzing wardrobe image without image file
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_analyze\_wardrobe\_image\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test analyzing wardrobe image
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_analyze\_wardrobe\_image\_unauthorized**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test analyzing wardrobe image without authentication
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_check\_wardrobe\_duplicate\_success**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test checking wardrobe duplicate
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_check\_wardrobe\_duplicate\_unauthorized**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test checking wardrobe duplicate without authentication
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_check\_wardrobe\_duplicate\_with\_existing**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test checking duplicate with existing wardrobe item
- Test suite for /api/wardrobe/\* endpoints

### **TestWardrobeEndpoints.test\_delete\_wardrobe\_item\_nonexistent** (backend/tests/test\_wardrobe\_endpoints.py)

- Test deleting non-existent wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_delete\_wardrobe\_item\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test successfully deleting a wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_delete\_wardrobe\_item\_unauthorized**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test deleting wardrobe item without authentication
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_item\_nonexistent** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting non-existent wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_item\_other\_user** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe item belonging to another user
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_item\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting specific wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_item\_unauthorized** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting specific wardrobe item without authentication
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe items with paginated response
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_summary\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe summary
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_summary\_unauthorized**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe summary without authentication
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_unauthorized** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe without authentication
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_get\_wardrobe\_with\_category\_filter** (backend/tests/test\_wardrobe\_endpoints.py)

- Test getting wardrobe items filtered by category
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_update\_wardrobe\_item\_nonexistent**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test updating non-existent wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_update\_wardrobe\_item\_success** (backend/tests/test\_wardrobe\_endpoints.py)

- Test successfully updating a wardrobe item
- Test suite for /api/wardrobe/\* endpoints

**TestWardrobeEndpoints.test\_update\_wardrobe\_item\_unauthorized**

(backend/tests/test\_wardrobe\_endpoints.py)

- Test updating wardrobe item without authentication
- Test suite for /api/wardrobe/\* endpoints

## Backend — Wardrobe Pagination & Search (18)

**TestWardrobePagination.test\_get\_wardrobe\_default\_limit** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that default limit is 10 items
- Test suite for wardrobe pagination

**TestWardrobePagination.test\_get\_wardrobe\_limit\_max** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that limit cannot exceed 100
- Test suite for wardrobe pagination

**TestWardrobePagination.test\_get\_wardrobe\_offset\_beyond\_total**

(backend/tests/test\_wardrobe\_pagination\_search.py)

- Test offset beyond total items returns empty list
- Test suite for wardrobe pagination

**TestWardrobePagination.test\_get\_wardrobe\_pagination\_with\_category**

(backend/tests/test\_wardrobe\_pagination\_search.py)

- Test pagination with category filter
- Test suite for wardrobe pagination

**TestWardrobePagination.test\_get\_wardrobe\_with\_custom\_limit**

(backend/tests/test\_wardrobe\_pagination\_search.py)

- Test wardrobe with custom limit
- Test suite for wardrobe pagination

**TestWardrobePagination.test\_get\_wardrobe\_with\_offset** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test wardrobe with offset
- Test suite for wardrobe pagination

**TestWardrobeSearch.test\_search\_by\_category\_keyword** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test searching by category keyword (shirt, trouser, etc.)
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_by\_color** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test searching by color
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_by\_description** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test searching by description
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_case\_insensitive** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that search is case insensitive
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_empty\_results** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test search that returns no results
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_multi\_word\_and\_logic** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test multi-word search with AND logic
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_with\_category\_filter** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test search combined with category filter
- Test suite for wardrobe search functionality

**TestWardrobeSearch.test\_search\_with\_pagination** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test search with pagination



- Test suite for wardrobe search functionality

**TestWardrobeSummary.test\_summary\_by\_color** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that summary includes color counts
- Test suite for wardrobe summary with category counts

**TestWardrobeSummary.test\_summary\_category\_counts** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that summary returns correct category counts
- Test suite for wardrobe summary with category counts

**TestWardrobeSummary.test\_summary\_category\_normalization**

(backend/tests/test\_wardrobe\_pagination\_search.py)

- Test that categories are normalized to lowercase in summary
- Test suite for wardrobe summary with category counts

**TestWardrobeSummary.test\_summary\_empty\_wardrobe** (backend/tests/test\_wardrobe\_pagination\_search.py)

- Test summary with empty wardrobe
- Test suite for wardrobe summary with category counts

## Frontend — App (1)

**App.test#1** (frontend/src/App.test.tsx)

- Renders AI Outfit Suggestor app
- File: App.test.tsx

## Frontend — Wardrobe (integration) (2)

**Wardrobe.integration.test#1** (frontend/src/views/components/Wardrobe.integration.test.tsx)

- Loads and renders items from the wardrobe API
- File: Wardrobe.integration.test.tsx

**Wardrobe.integration.test#2** (frontend/src/views/components/Wardrobe.integration.test.tsx)

- Shows a user-visible error (not a blank screen) when the wardrobe API fails
- File: Wardrobe.integration.test.tsx

## Frontend — Wardrobe (unit) (3)

**Wardrobe.test#1** (frontend/src/views/components/Wardrobe.test.tsx)

- Loads the wardrobe page and shows the page header
- File: Wardrobe.test.tsx

**Wardrobe.test#2** (frontend/src/views/components/Wardrobe.test.tsx)

- Shows empty list state when there are no wardrobe items
- File: Wardrobe.test.tsx

**Wardrobe.test#3** (frontend/src/views/components/Wardrobe.test.tsx)

- Shows wardrobe item list when items exist
- File: Wardrobe.test.tsx