

REST Assured Banking Use Cases

Here's how each **real-time use case in banking** works using REST APIs with **Rest Assured** examples and **error validation** scenarios (like for invalid users):

✓ 1. Login API → Token Retrieval

Purpose: Authenticate user and get a JWT token/session ID for secured APIs.

► Request

```
http

POST /api/v1/login
Content-Type: application/json

{
  "username": "user123",
  "password": "securePass!"
}
```

► Response

```
json

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ii..."
}
```

► Rest Assured Code (Token Retrieval)

```
java

Response response = given()
    .contentType("application/json")
    .body("{ \"username\": \"user123\", \"password\": \"securePass!\" }")
    .when()
    .post("/api/v1/login");

String token = response.jsonPath().getString("token");
System.out.println("Token: " + token);
```

✓ 2. Balance Check API

Purpose: To retrieve the account balance for the logged-in user.

► Request

```
http

GET /api/v1/accounts/balance
Authorization: Bearer <token>
```

► Response

```
json

{
  "accountNumber": "1234567890",
  "balance": 10320.45,
  "currency": "INR"
}
```

► Rest Assured Code

```
java

given()
  .header("Authorization", "Bearer " + token)
.when()
  .get("/api/v1/accounts/balance")
.then()
  .statusCode(200)
  .body("currency", equalTo("INR"));
```

✓ 3. Fund Transfer API

Purpose: Transfer funds from one account to another.

► Request

```
http

POST /api/v1/fundtransfer
Authorization: Bearer <token>
Content-Type: application/json

{
  "toAccount": "9876543210",
  "amount": 500.00,
  "remarks": "Rent Payment"
}
```

► Response

```
json

{
  "transactionId": "TXN784512",
  "status": "SUCCESS"
}
```

► Rest Assured Code

```
java

given()
    .header("Authorization", "Bearer " + token)
    .contentType("application/json")
    .body("{ \"toAccount\": \"9876543210\", \"amount\": 500.00, \"remarks\": \"Rent Payment\" }")
.when()
    .post("/api/v1/fundtransfer")
.then()
    .statusCode(200)
    .body("status", equalTo("SUCCESS"));
```

✓ 4. Transaction History API

Purpose: Fetch past transactions for an account.

► Request

```
http

GET /api/v1/accounts/transactions?from=2024-01-01&to=2024-07-28
Authorization: Bearer <token>
```

► Response

```
json

[
  {
    "date": "2024-07-15",
    "amount": -500.00,
    "description": "Rent Payment"
  },
  {
    "date": "2024-07-10",
    "amount": 10000.00,
    "description": "Salary"
  }
]
```

► Rest Assured Code

```
java

given()
    .header("Authorization", "Bearer " + token)
    .queryParams("from", "2024-01-01")
    .queryParams("to", "2024-07-28")
.when()
    .get("/api/v1/accounts/transactions")
.then()
    .statusCode(200)
    .body("[0].description", equalTo("Rent Payment"));
```

✓ 5. Mini-Statement Download API

Purpose: Download a PDF or JSON of recent 5–10 transactions.

► Request

```
http

GET /api/v1/accounts/ministatement
Authorization: Bearer <token>
```

► Response

Returns: PDF or JSON

```
json

{
  "transactions": [
    { "date": "2024-07-28", "amount": -100, "desc": "Fuel" },
    ...
  ]
}
```

► Rest Assured Code

```
java

given()
  .header("Authorization", "Bearer " + token)
.when()
  .get("/api/v1/accounts/ministatement")
.then()
  .statusCode(200)
  .body("transactions.size()", greaterThan(0));
```

✗ 6. Validate Error Codes – Invalid Users

Purpose: Check response code and message for wrong login or unauthorized access.

● Invalid Login

```
http

POST /api/v1/login
{
  "username": "invalidUser",
  "password": "wrongPass"
}
```

✓ Expected Response:

```
json

{
  "error": "Invalid credentials"
```

```
}

```

Rest Assured Validation

```
java

given()
    .contentType("application/json")
    .body("{ \"username\": \"invalidUser\", \"password\": \"wrongPass\" }")
.when()
    .post("/api/v1/login")
.then()
    .statusCode(401)
    .body("error", equalTo("Invalid credentials"));

```

Unauthorized Access

```
h

GET /api/v1/accounts/balance
// No token or invalid token

```

Expected Response:

```
json

{
  "error": "Unauthorized"
}

```

Rest Assured Validation

```
java

given()
    .header("Authorization", "Bearer invalidToken")
.when()
    .get("/api/v1/accounts/balance")
.then()
    .statusCode(401)
    .body("error", equalTo("Unauthorized"));

```

Would you like to set up a **mock banking API** to practice these cases hands-on with Rest Assured locally or on Postman?