

Integration Tests Report

Project Information:

- **Project Name:** NBP Crasher
- **Project Manager:** Przemysław Kowalski
- **Report Date:** 18.01.2025
- **Test Environment:** Fedora Linux (41)

Scope of Testing:

- **Objective:** Verify the handling of errors and data validation between the frontend and backend when accessing the endpoint /get_stats using the GET method.
- **Key Focus Areas:**
 - API error handling and response validation
 - Frontend behavior and messaging when backend errors occur
 - Correctness of status codes and error messages

Test Cases Overview:

Description	Expected Result	Actual Result	Status
Valid requests for periods predefined periods	Status code 200, correct data displayed.	Status code 200, data displayed.	Passed
Valid request for custom date	Status code 200, correct data displayed.	Status code 200, data displayed.	Passed
Request with equal currencies (currencies_are_equal)	Status code 400, error message shown.	Status code 400, error message shown.	Passed
Request with invalid date format (invalid_date_format)	Status code 400, error message shown.	Status code 400, error message shown.	Passed
Request with unsupported date (date_not_supported)	Status code 400, error message shown.	Status code 400, error message shown.	Passed

Description	Expected Result	Actual Result	Status
Request with invalid data (<code>invalid_data</code>)	Status code 400, error message shown.	Status code 400, error message shown.	Passed
Simulated server error (No NBP connection) (<code>internal_server_error</code>).	Status code 500, error message shown.	Status code 500, error message shown.	Passed

Key Observations:

- **Error Handling:** All backend errors were properly propagated to the frontend with appropriate status codes and messages.
- **Validation Errors:** Status codes were correctly returned for malformed requests, with proper frontend display.
- **Frontend Behavior:** Each error scenario was clear and aligned with expected behavior.

Conclusion:

- Integration testing for the `/api/get_stats` endpoint was successfully completed. All error scenarios and validation cases were properly handled, and the frontend responded as expected, ensuring a robust and user-friendly experience.