

Exception Handling

Date	3 November 2023
Team ID	NM2023TMID09736
Project Name	How to create a Reel using Canva

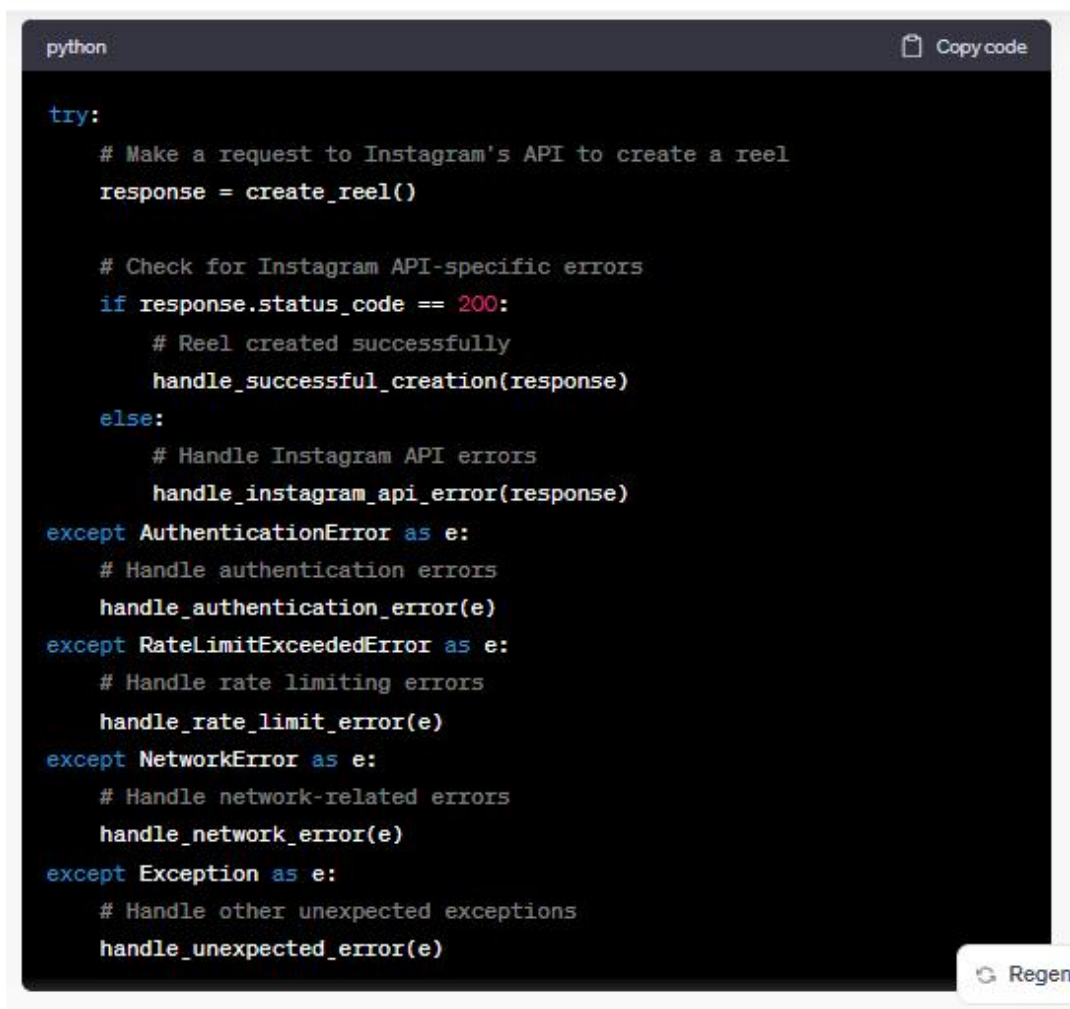
Exception handling is an important aspect of software development, including when you're working with Instagram's API to create reels. Instagram's API may change over time, so be sure to refer to the latest documentation and error handling practices provided by Instagram. Here's a general outline of how you can handle exceptions when creating Instagram reels:

- 1. Authentication and Authorization Errors:** Handle exceptions related to authentication and authorization, such as invalid access tokens or permission errors. These are crucial to ensure that your app can interact with Instagram's API.
- 2. API Rate Limiting:** Instagram may impose rate limits on API requests to prevent abuse. Handle rate-limiting exceptions by respecting the limits and implementing retry mechanisms with backoff strategies if necessary.
- 3. Network Errors:** Handle network-related exceptions, such as connection timeouts or request failures. You can use try-catch blocks to capture these exceptions and implement retry logic as needed.
- 4. Validation Errors:** Ensure that the data you're sending to Instagram's API is valid. Handle validation errors by checking the data before making the request and responding appropriately if the data is invalid.
- 5. Instagram API Errors:** Instagram's API may return specific error codes and messages. Handle these errors by checking the response from the API and taking appropriate actions based on the error code and message.
- 6. Logging and Monitoring:** Implement comprehensive logging to record exceptions and errors, making it easier to diagnose and fix issues. Additionally, consider setting up monitoring and alerting systems to detect and respond to errors in real-time.
- 7. Graceful Degradation:** In the event of an exception, implement a graceful degradation strategy. For example, if you're creating Instagram reels, you might

queue failed requests for later retry, notify the user of the failure, or handle the situation in a way that doesn't disrupt the user experience.

8. User Feedback: Provide clear and helpful error messages to users when exceptions occur. This can help them understand the issue and take appropriate actions.

Here's a pseudo-code example of exception handling when creating an Instagram reel:



```
python                                                                    Copy code

try:
    # Make a request to Instagram's API to create a reel
    response = create_reel()

    # Check for Instagram API-specific errors
    if response.status_code == 200:
        # Reel created successfully
        handle_successful_creation(response)
    else:
        # Handle Instagram API errors
        handle_instagram_api_error(response)
except AuthenticationError as e:
    # Handle authentication errors
    handle_authentication_error(e)
except RateLimitExceededError as e:
    # Handle rate limiting errors
    handle_rate_limit_error(e)
except NetworkError as e:
    # Handle network-related errors
    handle_network_error(e)
except Exception as e:
    # Handle other unexpected exceptions
    handle_unexpected_error(e)

Regen
```

Remember that this is a simplified example, and you should tailor your exception handling to your specific use case and the Instagram API version you're working with. Always consult the latest API documentation for the most up-to-date information on error handling and best practices.