# 3. Morpheus Custom Stages: 기본 구조 및 메시지 업데이트

## 프로젝트 개요

Morpheus의 핵심인 \*\*사용자 정의 스테이지(Custom Stage)\*\*를 작성하는 방법을 학습했습니다. 메시지를 변형 없이 통과시키는 기본 Passthrough 스테이지부터, 메시지 메타데이터와 페이로드의 데이터를 실제로 **업데이트**하는 커스텀 로직까지 구현했습니다.

## ⚙️ 사용 기술 및 역할

* **주요 기술:** Morpheus SinglePortStage, GpuAndCpuMixin, PassThruTypeMixin, MessageMeta.mutable\_dataframe().
* **핵심 역할:**
  1. **Custom Stage 클래스 정의:** SinglePortStage를 상속받아 name, supports\_cpp\_node, on\_data 메서드를 정의하는 기본 구조 확립.
  2. **데이터 업데이트:** on\_data 메서드 내에서 mutable\_dataframe() 컨텍스트 관리자를 사용하여 메시지의 cuDF 페이로드 데이터를 안전하게 읽고 쓰는(업데이트) 로직 구현.

## 🛠️ 주요 구현 내용 (22-update-message-meta.ipynb, 23-update-message.ipynb 기반)

1. **Passthrough Stage (21-custom-passthrough-stage.ipynb):** 입력 메시지를 그대로 반환하는 on\_data(self, message) -> message 로직을 구현하여 커스텀 스테이지의 기본적인 데이터 흐름을 확인했습니다.
2. **MessageMeta 업데이트:** 사용자 인증 로그 데이터에서 user 컬럼의 값을 \*\*대문자(Uppercase)\*\*로 변경하는 process 함수를 정의하고, 이를 커스텀 스테이지의 on\_data 메서드 내에서 적용했습니다.
3. **안전한 데이터 수정:** **with message.mutable\_dataframe() as df:** 구문을 사용하여 GPU 메모리에 있는 데이터프레임에 대한 \*\*경쟁 조건(Race Condition)\*\*을 피하고 안전하게 데이터를 수정했습니다.

## 📌 주요 코드 로직 (메시지 페이로드 업데이트 추상화)

from morpheus.pipeline.single\_port\_stage import SinglePortStage  
from morpheus.messages import MessageMeta # MessageMeta 또는 ControlMessage  
  
# 메시지 페이로드(DataFrame)를 수정하는 커스텀 스테이지  
class UpdateMessageStage(SinglePortStage):  
  
 # on\_data 메서드에서 실제 데이터 처리 로직 수행  
 def on\_data(self, message: MessageMeta) -> MessageMeta:  
 # mutable\_dataframe() 컨텍스트 관리자를 사용하여 안전하게 데이터 수정  
 with message.mutable\_dataframe() as df:  
 # 예시: 'user' 컬럼의 모든 값을 대문자로 변경  
 if 'user' in df.columns:  
 df['user'] = df['user'].str.upper()  
   
 return message