# 1. Morpheus Custom Stage 구현: 데이터 변환 및 업데이트

## 프로젝트 개요

NVIDIA Morpheus 파이프라인에서 메시지 내부의 **데이터 페이로드를 수정하는 커스텀 스테이지**를 구현했습니다. 이는 파이프라인의 특정 지점에서 데이터 변환, 정규화 또는 특정 비즈니스 로직을 삽입하는 기본 역량을 보여줍니다. (노트북: 23-update-message.ipynb, 23-update-message (2).ipynb 내용 통합)

## ⚙️ 사용 기술 및 역할

* **주요 기술:** NVIDIA Morpheus SDK (Python), MessageMeta, ControlMessage
* **핵심 역할:**
  + PassThruTypeMixin, SinglePortStage를 상속받아 파이프라인 구조 유지.
  + on\_data 메서드 내에서 MessageMeta 또는 ControlMessage의 데이터프레임(cuDF)에 접근하여 데이터를 수정하는 로직 구현. (예: 특정 컬럼 값을 대문자로 변경)

## 💡 핵심 구현 내용

1. **데이터 가변성 확보:** 메시지의 페이로드(MessageMeta 또는 ControlMessage)에 접근하여 **Mutable DataFrame**을 사용해 데이터 수정의 효율성과 안정성을 확보했습니다.
2. **컬럼 업데이트 로직:** 데이터프레임의 특정 컬럼에 대해 문자열 조작 함수(예: .str.upper())를 적용하여 데이터를 일괄적으로 변환했습니다.
3. **파이프라인 흐름:** FileSource -> Deserialize -> **Custom Stage** -> InMemorySink의 표준 선형 파이프라인에 통합하여 작동을 검증했습니다.

## 📌 주요 코드 로직 (추상화)

from morpheus.pipeline import PassThruTypeMixin, SinglePortStage  
from morpheus.messages import MessageMeta # 또는 ControlMessage  
  
class UpdateMessageStage(PassThruTypeMixin, GpuAndCpuMixin, SinglePortStage):  
  
 def \_\_init\_\_(self, config: Config):  
 super().\_\_init\_\_(config)  
  
 @property  
 def name(self) -> str:  
 return "update-message-stage"  
  
 def on\_data(self, message: MessageMeta) -> MessageMeta:  
 # 데이터프레임에 접근하여 변경 가능한(mutable) 상태로 만듭니다.  
 # 이를 통해 데이터프레임의 내용을 안전하게 수정할 수 있습니다.  
 with message.mutable\_dataframe() as df:  
 # 예시: 'user' 컬럼의 모든 값을 대문자로 변경하여 저장 (IP/Port 변환 등의 전처리 로직 대체)  
 if 'user' in df.columns:  
 df['user'] = df['user'].str.upper()  
   
 return message  
  
# 파이프라인에 커스텀 스테이지 통합  
# pipeline.add\_stage(UpdateMessageStage(config))