1. **Dodaj nowy przepływ**

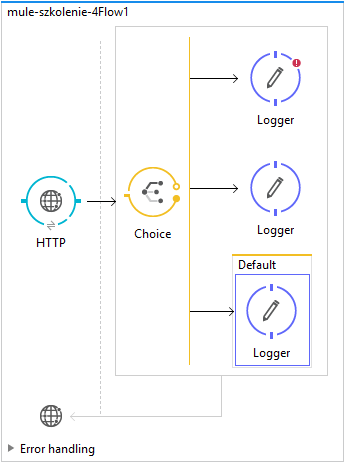
Z Mule Palette wyszukujemy komponent Flow i przeciągamy na obszar roboczy

1. **Dodaj HTTP Input**

Dodaj HTTP input source, ustaw ścieżkę na /lesson4, z operacją POST

1. **Dodaj komponent Choice**

Dodaj komponent Choice do przepływu, weryfikujący czy query param o nazwie type jest STR czy BIN. Dodaj 3 loggery, jeden z komentarzem “wysyłamy string”, drugi “wysyłamy tablice byte[]”. Ostatni z komentarzem “brak query param lub niewłaściwa wartość”.



Dodaj odpowiednie kryteria dla poszczególnych terminali:

message.inboundProperties.'http.query.params'.type=="STR"

message.inboundProperties.'http.query.params'.type=="BIN"

1. **Konwersja payloada do stringa**

Nasz payload odebrany przez HttpInput jest typu org.glassfish.grizzly.utils.BufferInputStream, należy pamiętać że HTTP Input zawsze dostarcza payload w formie strumienia. W celu dokonania konwersji do stringa używamy ObjectToString. Konwersję wykonujemy dla scenariusza STR.

1. **Wyślij komunikat string na kolejkę LESSON4.Q.IN**

W tym scenariuszu używamy komponentu JMS do przekazania naszego komunikatu na wcześniej zbudowany Flow. Dodaj komponent JMS, ustaw mu kolejkę oraz wybierz wcześniej zdefiniowaną konfigurację Active MQ.

1. **Konwersja payloada do tablicy byte**

Nasz payload odebrany przez HttpInput jest typu org.glassfish.grizzly.utils.BufferInputStream, należy pamiętać że HTTP Input zawsze dostarcza payload w formie strumienia. W celu dokonania konwersji do byte[] używamy ObjectToByteArray. Konwersję wykonujemy dla scenariusza BIN.

1. **Wyślij komunikat string na kolejkę LESSON4.Q.IN**

W tym scenariuszu używamy komponentu JMS do przekazania naszego komunikatu na wcześniej zbudowany Flow. Dodaj komponent JMS, ustaw mu kolejkę oraz wybierz wcześniej zdefiniowaną konfigurację Active MQ.

1. **Dodaj ‘ok’ jako payload zwrotny**

Dodaj komponent Set Paylod i ustaw mu “ok” za komponentem Choice