1. **Utworzenie projektu**

Utwórz nowy projekt mule-szkolenie-5a z poniższymi parametrami:

**Runtime**: CE 3.9.0

**Maven**: TAK

**Group ID**: pl.com.sages.training

1. Przeciągnij plik **weather.raml** do katalogu src/main/api
2. Wygeneruj API z pliku weather.raml

*APIKit wygeneruje plik weather.xml. W nim będzie główny przepływ z APIkit Routerem oraz prywatne przepływu do implementacji operacji.*

1. Usuń plik konfiguracyjny **mule-szkolenie-5a.xml**
2. W pliku weather.xml przejdź na zakładkę **Global Elements**
3. Zmień numer portu HTTP Listener Configuration na 8091
4. Wróć na zakładkę Message Flow i usuń przepływ **weather-console**

*Konsola ta pozwala na testowanie API, jednak wygodniej jest używać oprogramowania trzeciego jak np. Postamn*

1. Kliknij na komponent **Set Payload** we flow post:/countries/{country}/cities:application/json:weather-config

*Jak można zauważyć payload został ustawiony zgodnie z dostępnym przykładem, który został wprowadzony w pliku weather.raml*

1. Kliknij na komponent **Set Payload** we flow get:/countries/{country}/cities:weather-config i ustaw payload na zgodny z dostarczonym schematem w weather.raml

*Zgodnie z definicji raml wynika że GET /countries/{country}/cities zwraca tablicę obiektów typu city.*

Ustaw następującą zawartość:

[

{

"name": "Warszawa",

"code": "WA",

"location": {

"lat": 10,

"long": 40

}

},

{

"name": "Kraków",

"code": "KR",

"location": {

"lat": 10,

"long": 40

}

}

]

1. Przeciągnij komponent logger za **Set Payload** i wyświetli w nim następujący komunikat, zastępując odpowiednie placeholdery wyrażeniami MEL

**Komunikat**: Response for country {kraj z URI}: {payload}

Odpowiedź poniżej:

Response for country #[flowVars.country]: #[payload]

1. Kliknij na komponent **Set Payload** we flow get:/weather:weather-config i ustaw payload na zgodny z dostarczonym schematem w weather.raml

Zgodnie z definicji raml wynika że GET /weather zwraca obiekt typu forecast.

Ustaw następującą zawartość, zastępując placeholder wyrażeniem MEL:

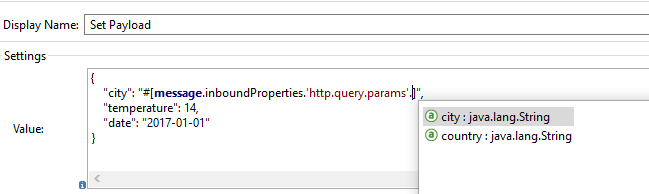
{

"city": "{kraj z query parameter}",

"temperature": 14,

"date": "2017-01-01"

}



1. Przeciągnij komponent **logger** za Set Payload i wyświetli w nim następujący komunikat, zastępując odpowiednie placeholdery wyrażeniami MEL

Komunikat: Response for {kraj}-{miasto}: {payload}

Odpowiedź poniżej:

Response for #[message.inboundProperties.'http.query.params'.country]-#[message.inboundProperties.'http.query.params'.city]: #[payload]

1. Uruchom projekt

Uruchom aplikację Postman. Wykonaj operację GET na zasobie <http://localhost:8091/api/countries/Polska/cities>

Oczekiwany rezultat:

**Body**: "[ { \"name\": \"Warszawa\", \"code\": \"WA\", \"location\": { \"lat\": 10, \"long\": 40 } }, { \"name\": \"Kraków\", \"code\": \"KR\", \"location\": { \"lat\": 10, \"long\": 40 } } ] "

**Log**: Response for country Polska: [ { "name": "Warszawa", "code": "WA", "location": { "lat": 10, "long": 40 } }, { "name": "Kraków", "code": "KR", "location": { "lat": 10, "long": 40 } } ]

Uruchom aplikację Postman. Wykonaj operację GET na zasobie <http://localhost:8091/api/weather?city=Krakow&country=Polska>

Oczekiwany rezultat:

**Body**: {

"city": "Krakow",

"temperature": 14,

"date": "2017-01-01"

}

**Log**: Response for Polska-Krakow: { "city": "Krakow", "temperature": 14, "date": "2017-01-01" }

Uruchom aplikację Postman. Wykonaj operację POST na zasobie <http://localhost:8091/api/countries/PL/cities> z body:

{

"name": "Krakow",

"code": "KR",

"location": {

"lat": 12.2,

"long": 55.0

}

}

Oczekiwany rezultat:

**Body**: {

"id": "Warszawa - Wa",

"geo": "(52.2, 21.0)"

}

Uruchom aplikację Postman. Wykonaj operację POST na zasobie <http://localhost:8091/api/countries/PL/cities> z body:

{

"name": "Krakow",

"code": "KR",

"location": {

"lat": 12.2

}

}

Oczekiwany rezultat:

**Body**: {

"message": "Bad request"

}

**Status HTTP**: 400

**Log**: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Message : Error validating JSON. Error: - Missing required field "long"

1. Kliknij komponent **Set Payload** w get:/countries/{country}/cities:weather-config i ustaw MIME Type na application/json
2. Uruchom projekt

Uruchom aplikację Postman. Wykonaj operację GET na zasobie <http://localhost:8091/api/countries/Polska/cities>

Oczekiwany rezultat:

**Body**: [

{

"name": "Warszawa",

"code": "WA",

"location": {

"lat": 10,

"long": 40

}

},

{

"name": "Kraków",

"code": "KR",

"location": {

"lat": 10,

"long": 40

}

}

]

**Log**: Response for country Polska: [ { "name": "Warszawa", "code": "WA", "location": { "lat": 10, "long": 40 } }, { "name": "Kraków", "code": "KR", "location": { "lat": 10, "long": 40 } } ]