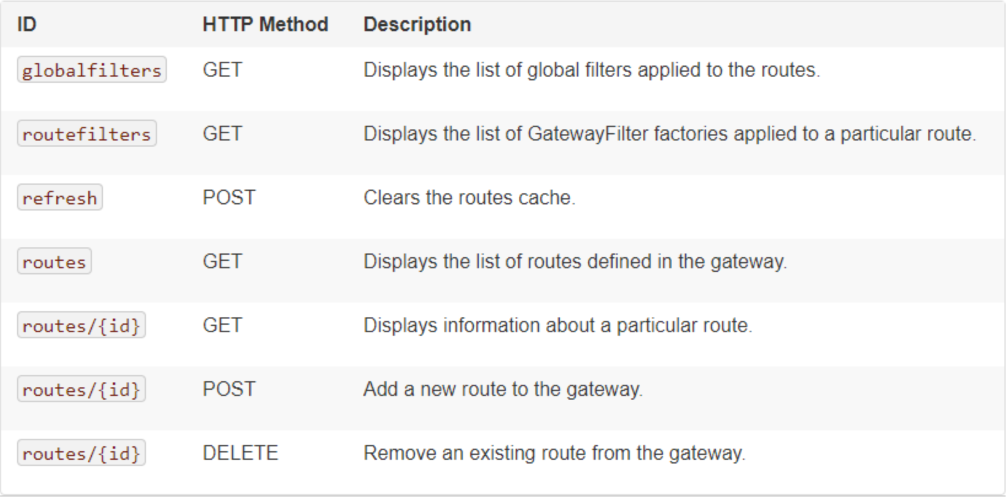
scg crud test

Spring Cloud Gateway actuator endpoints 요약

- base-path : /actuator/gateway



ID	HTTP Method	Description
`globalfilters`	GET	Displays the list of global filters applied to the routes.
`routefilters`	GET	Displays the list of GatewayFilter factories applied to a particular route.
`refresh`	POST	Clears the routes cache.
`routes`	GET	Displays the list of routes defined in the gateway.
`routes/{id}`	GET	Displays information about a particular route.
`routes/{id}`	POST	Add a new route to the gateway.
`routes/{id}`	DELETE	Remove an existing route from the gateway.

1) route filters 조회

- Global Filters

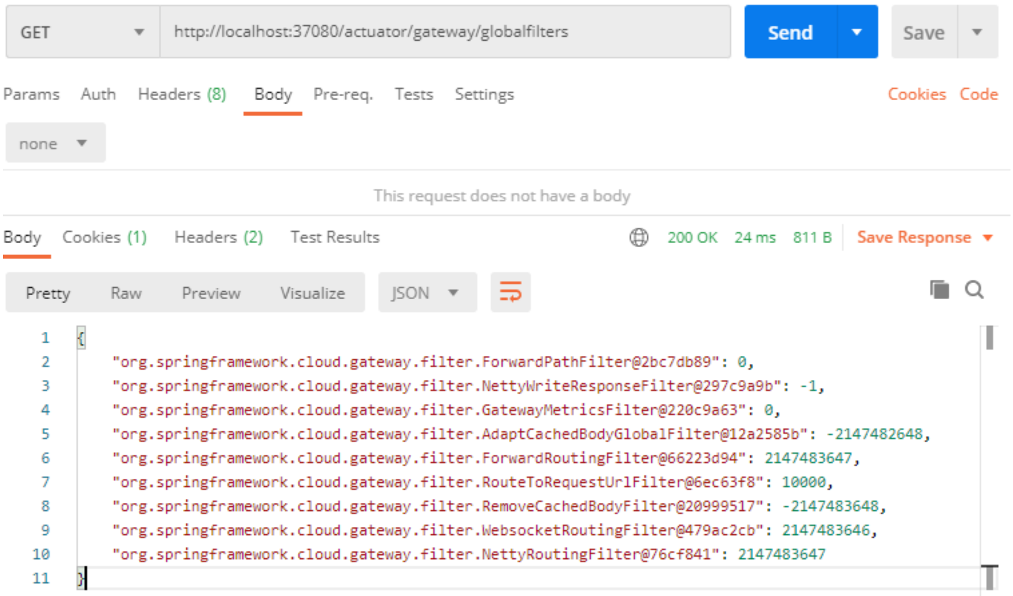
모든 route에 적용된 global filter를 조회하기 위해 /actuator/gateway/globalfilters 으로 GET 요청

request

GET http://localhost:37080/actuator/gateway/globalfilters

response

```
{
    "org.springframework.cloud.gateway.filter.ForwardPathFilter@2bc7db89": 0,
    "org.springframework.cloud.gateway.filter.NettyWriteResponseFilter@297c9a9b": -1,
    "org.springframework.cloud.gateway.filter.GatewayMetricsFilter@220c9a63": 0,
    "org.springframework.cloud.gateway.filter.AdaptCachedBodyGlobalFilter@12a2585b": -2147482648,
    "org.springframework.cloud.gateway.filter.ForwardRoutingFilter@66223d94": 2147483647,
    "org.springframework.cloud.gateway.filter.RouteToRequestUrlFilter@6ec63f8": 10000,
    "org.springframework.cloud.gateway.filter.RemoveCachedBodyFilter@20999517": -2147483648,
    "org.springframework.cloud.gateway.filter.WebsocketRoutingFilter@479ac2cb": 2147483646,
    "org.springframework.cloud.gateway.filter.NettyRoutingFilter@76cf841": 2147483647
}
```



GET http://localhost:37080/actuator/gateway/globalfilters

Body: none

This request does not have a body

200 OK 24 ms 811 B

```
{
    "org.springframework.cloud.gateway.filter.ForwardPathFilter@2bc7db89": 0,
    "org.springframework.cloud.gateway.filter.NettyWriteResponseFilter@297c9a9b": -1,
    "org.springframework.cloud.gateway.filter.GatewayMetricsFilter@220c9a63": 0,
    "org.springframework.cloud.gateway.filter.AdaptCachedBodyGlobalFilter@12a2585b": -2147482648,
    "org.springframework.cloud.gateway.filter.ForwardRoutingFilter@66223d94": 2147483647,
    "org.springframework.cloud.gateway.filter.RouteToRequestUrlFilter@6ec63f8": 10000,
    "org.springframework.cloud.gateway.filter.RemoveCachedBodyFilter@20999517": -2147483648,
    "org.springframework.cloud.gateway.filter.WebsocketRoutingFilter@479ac2cb": 2147483646,
    "org.springframework.cloud.gateway.filter.NettyRoutingFilter@76cf841": 2147483647
}
```

- Route Filters

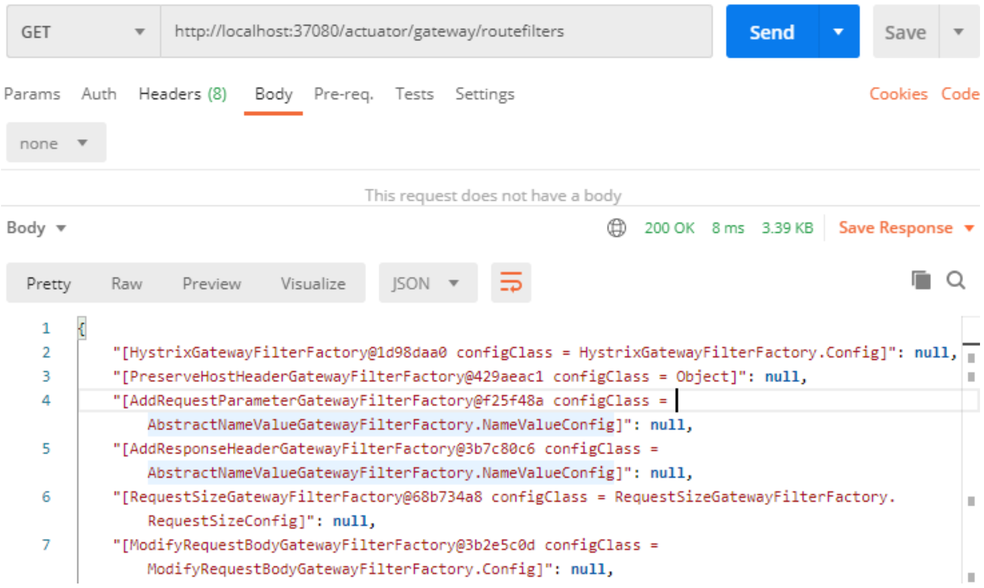
route에 적용된 GatewayFilter factories 를 조회하기 위해 /actuator/gateway/routefilters 에 GET 요청

request

GET http://localhost:37080/actuator/gateway/routefilters

response

```
{
    "[HystrixGatewayFilterFactory@1d98daa0 configClass = HystrixGatewayFilterFactory.Config]": null,
    "[PreserveHostHeaderGatewayFilterFactory@429aeac1 configClass = Object]": null,
    "[AddRequestParameterGatewayFilterFactory@f25f48a configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": nu
    "[AddResponseHeaderGatewayFilterFactory@3b7c80c6 configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": nul
    "[RequestSizeGatewayFilterFactory@68b734a8 configClass = RequestSizeGatewayFilterFactory.RequestSizeConfig]": null,
    "[ModifyRequestBodyGatewayFilterFactory@3b2e5c0d configClass = ModifyRequestBodyGatewayFilterFactory.Config]": null,
    "[RemoveResponseHeaderGatewayFilterFactory@1de4285e configClass = AbstractGatewayFilterFactory.NameConfig]": null,
    "[SetStatusGatewayFilterFactory@1dad01fe configClass = SetStatusGatewayFilterFactory.Config]": null,
    "[RetryGatewayFilterFactory@634ff56 configClass = RetryGatewayFilterFactory.RetryConfig]": null,
    "[RewriteResponseHeaderGatewayFilterFactory@4099209b configClass = RewriteResponseHeaderGatewayFilterFactory.Config]": null
    "[SetRequestHeaderGatewayFilterFactory@e91b4f4 configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": null,
    "[PrefixPathGatewayFilterFactory@b9d018b configClass = PrefixPathGatewayFilterFactory.Config]": null,
    "[RewriteLocationResponseHeaderGatewayFilterFactory@43ac0a68 configClass = RewriteLocationResponseHeaderGatewayFilterFactor
    "[StripPrefixGatewayFilterFactory@3e3cd6fe configClass = StripPrefixGatewayFilterFactory.Config]": null,
    "[SetResponseHeaderGatewayFilterFactory@58ae402b configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": nul
    "[SetPathGatewayFilterFactory@66522ead configClass = SetPathGatewayFilterFactory.Config]": null,
    "[RequestHeaderToRequestUriGatewayFilterFactory@1de0641b configClass = AbstractGatewayFilterFactory.NameConfig]": null,
    "[DedupeResponseHeaderGatewayFilterFactory@7499eac7 configClass = DedupeResponseHeaderGatewayFilterFactory.Config]": null,
    "[RemoveRequestParameterGatewayFilterFactory@8bd076a configClass = AbstractGatewayFilterFactory.NameConfig]": null,
    "[MapRequestHeaderGatewayFilterFactory@54336976 configClass = MapRequestHeaderGatewayFilterFactory.Config]": null,
    "[RemoveRequestHeaderGatewayFilterFactory@6b4fc2d1 configClass = AbstractGatewayFilterFactory.NameConfig]": null,
    "[SecureHeadersGatewayFilterFactory@5a484ce1 configClass = Object]": null,
    "[ModifyResponseBodyGatewayFilterFactory@5438c17a configClass = ModifyResponseBodyGatewayFilterFactory.Config]": null,
    "[RewritePathGatewayFilterFactory@1378eea2 configClass = RewritePathGatewayFilterFactory.Config]": null,
    "[SetRequestHostHeaderGatewayFilterFactory@2ffe243f configClass = SetRequestHostHeaderGatewayFilterFactory.Config]": null,
    "[RequestHeaderSizeGatewayFilterFactory@1a464fa3 configClass = RequestHeaderSizeGatewayFilterFactory.Config]": null,
    "[SaveSessionGatewayFilterFactory@3728a578 configClass = Object]": null,
    "[SpringCloudCircuitBreakerHystrixFilterFactory@4215e133 configClass = SpringCloudCircuitBreakerFilterFactory.Config]": nul
    "[FallbackHeadersGatewayFilterFactory@70a898b0 configClass = FallbackHeadersGatewayFilterFactory.Config]": null,
    "[AddRequestHeaderGatewayFilterFactory@7bdf61ad configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": null
    "[RedirectToGatewayFilterFactory@79eeff87 configClass = RedirectToGatewayFilterFactory.Config]": null
}
```



GET http://localhost:37080/actuator/gateway/routefilters

This request does not have a body

200 OK 8 ms 3.39 KB

```
{
    "[HystrixGatewayFilterFactory@1d98daa0 configClass = HystrixGatewayFilterFactory.Config]": null,
    "[PreserveHostHeaderGatewayFilterFactory@429aeac1 configClass = Object]": null,
    "[AddRequestParameterGatewayFilterFactory@f25f48a configClass = 
        AbstractNameValueGatewayFilterFactory.NameValueConfig]": null,
    "[AddResponseHeaderGatewayFilterFactory@3b7c80c6 configClass =
        AbstractNameValueGatewayFilterFactory.NameValueConfig]": null,
    "[RequestSizeGatewayFilterFactory@68b734a8 configClass = RequestSizeGatewayFilterFactory.
        RequestSizeConfig]": null,
    "[ModifyRequestBodyGatewayFilterFactory@3b2e5c0d configClass =
        ModifyRequestBodyGatewayFilterFactory.Config]": null,
```

2) route cache 갱신

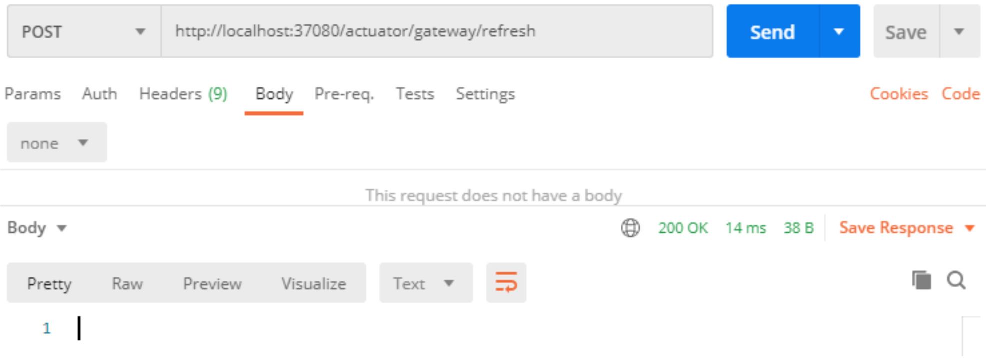
route cache 를 clear 하기 위해 /actuator/gateway/refresh 에 GET 요청

request

POST http://localhost:37080/actuator/gateway/refresh

response

200ok



POST http://localhost:37080/actuator/gateway/refresh

This request does not have a body

200 OK 14 ms 38 B

3)gateway에 정의된 route 조회

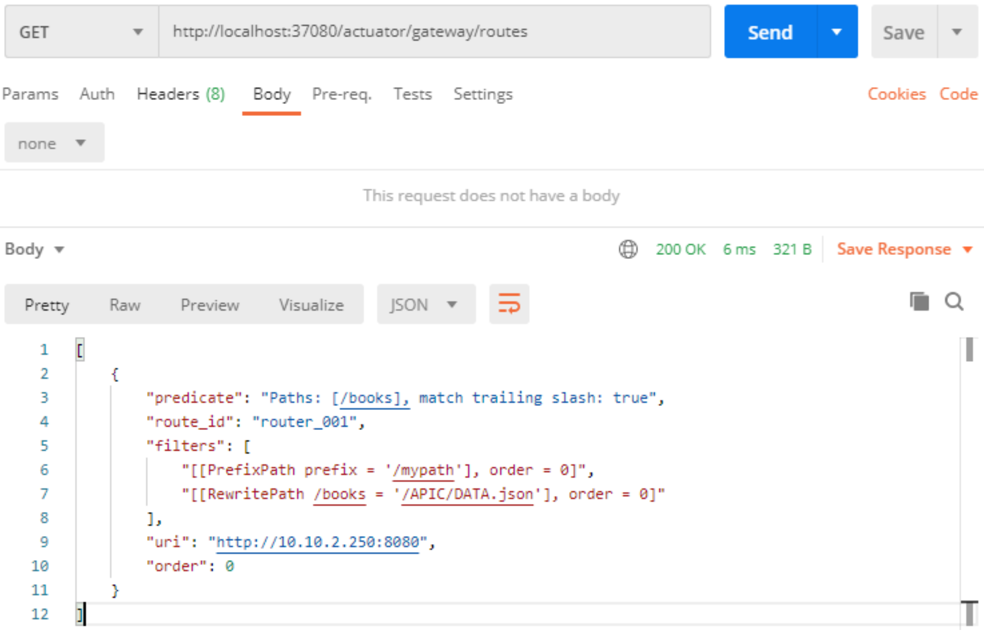
/actuator/gateway/routes 에 GET 요청

request

GET http://localhost:37080/actuator/gateway/routes

response

```
[
    {
        "predicate": "Paths: [/books], match trailing slash: true",
        "route_id": "router_001",
        "filters": [
            "[[PrefixPath prefix = '/mypath'], order = 0]",
            "[[RewritePath /books = '/APIC/DATA.json'], order = 0]"
        ],
        "uri": "http://10.10.2.250:8080",
        "order": 0
    }
]
```



GET http://localhost:37080/actuator/gateway/routes

This request does not have a body

200 OK 6 ms 321 B

```
[
    {
        "predicate": "Paths: [/books], match trailing slash: true",
        "route_id": "router_001",
        "filters": [
            "[[PrefixPath prefix = '/mypath'], order = 0]",
            "[[RewritePath /books = '/APIC/DATA.json'], order = 0]"
        ],
        "uri": "http://10.10.2.250:8080",
        "order": 0
    }
]
```

4) 특정 route에 대한 정보 조회

한 개의 route에 대한 정보를 조회하기 위해 /actuator/gateway/routes/{id} 에 GET 요청

request

GET http://localhost:37080/actuator/gateway/routes/first_route

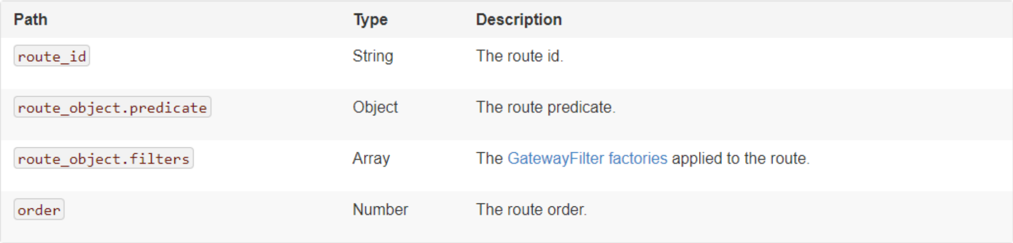
resoponse

```
{
  "id": "first_route",
  "predicates": [{
    "name": "Path",
    "args": {"_genkey_0":"/first"}
  }],
  "filters": [],
  "uri": "https://www.uri-destination.org",
  "order": 0
}]
```

5) 특정 route 생성

1. route의 필드 설정에 대한 JSON BODY와 함께 POST 요청

- /gateway/routes/{id_route_to_create} 에 POST 요청
- {id_route_to_create} 에는 새로 생성할 route의 id 명 입력
- body에는 JSON 형식으로 아래와 같은 필드를 포함



Path	Type	Description
`route_id`	String	The route id.
`route_object.predicate`	Object	The route predicate.
`route_object.filters`	Array	The GatewayFilter factories applied to the route.
`order`	Number	The route order.

request

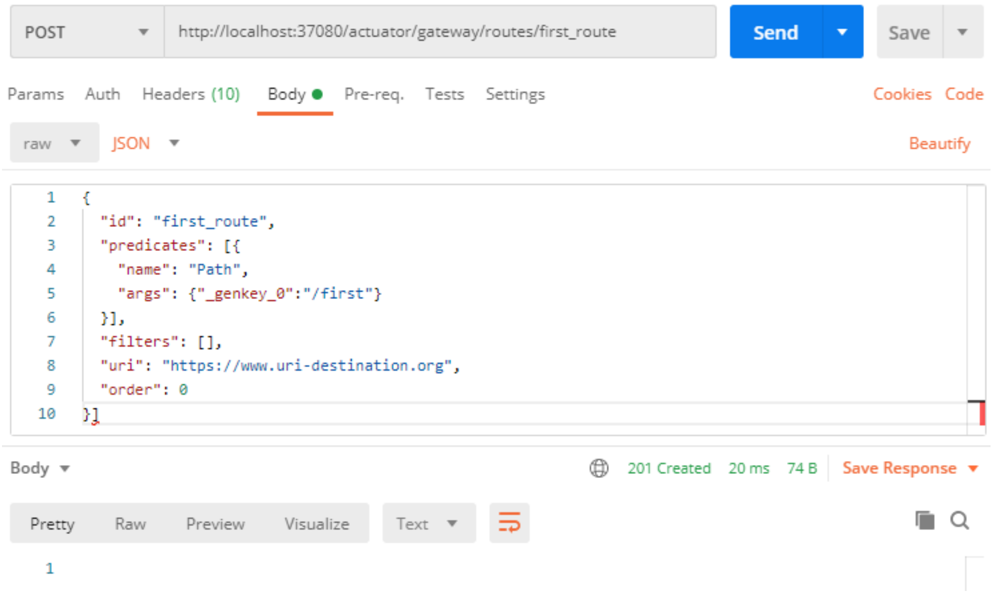
POST http://localhost:37080/actuator/gateway/routes/first_route

Body

```
{
  "id": "first_route",
  "predicates": [{
    "name": "Path",
    "args": {"_genkey_0":"/first"}
  }],
  "filters": [],
  "uri": "https://www.uri-destination.org",
  "order": 0
}]
```

response

201 created



```
{
  "id": "first_route",
  "predicates": [{
    "name": "Path",
    "args": {"_genkey_0":"/first"}
  }],
  "filters": [],
  "uri": "https://www.uri-destination.org",
  "order": 0
}]
```

2. route cache 갱신

- 새로 생성한 route 정보를 업데이트 시키기 위해 /actuator/gateway/refresh 에 POST 요청

request

POST http://localhost:37080/actuator/gateway/refresh

response

200 ok

3. 생성되었는지 route 목록 조회

- route 정보 생성을 확인하기 위해 actuator/gateway/routes 에 GET 요청

request

GET http://localhost:37080/actuator/gateway/routes

response

```
[
    {
        "predicate": "Paths: [/books], match trailing slash: true",
        "route_id": "router_001",
        "filters": [
            "[[PrefixPath prefix = '/mypath'], order = 0]",
            "[[RewritePath /books = '/APIC/DATA.json'], order = 0]"
        ],
        "uri": "http://10.10.2.250:8080",
        "order": 0
    },
    {
        "predicate": "Paths: [/first], match trailing slash: true",
        "route_id": "first_route",
        "filters": [],
        "uri": "https://www.uri-destination.org:443",
        "order": 0
    }
]
```

6) 특정 route 삭제

1. route id 로 gateway에 정의된 route 삭제

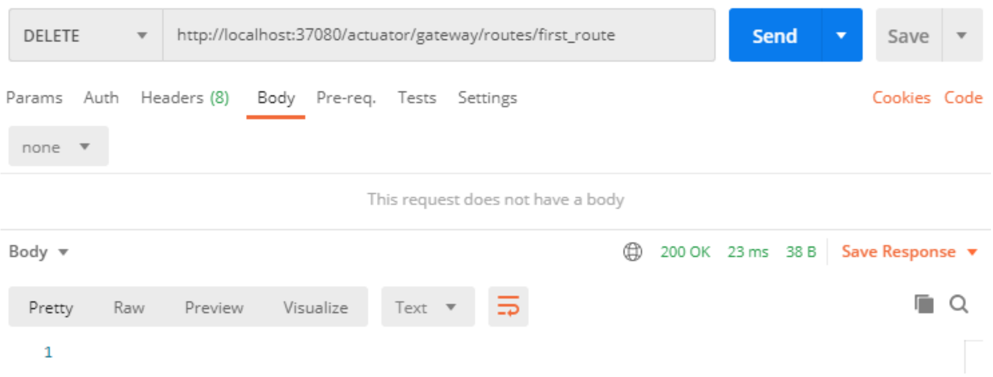
- /gateway/routes/{id_route_to_create} 에 DELETE요청
- {id_route_to_create} 에는 새로 삭제할 route의 id 명 입력
- body에는 JSON 형식으로 아래와 같은 필드를 포함

request

DELETE http://localhost:37080/actuator/gateway/routes/first_route

response

200 ok



DELETE http://localhost:37080/actuator/gateway/routes/first_route

Body: none

This request does not have a body

Body 200 OK 23 ms 38 B

```
1
```

2. route cache 갱신

- 삭제한 route 정보를 업데이트 시키기 위해 /actuator/gateway/refresh 에 POST 요청

request

POST http://localhost:37080/actuator/gateway/refresh

response

200 ok

3. 생성되었는지 route 목록 조회

- route 삭제가 됐는지 확인하기 위해 actuator/gateway/routes 에 GET 요청

request

GET http://localhost:37080/actuator/gateway/routes

response

```
[
    {
        "predicate": "Paths: [/books], match trailing slash: true",
        "route_id": "router_001",
        "filters": [
            "[[PrefixPath prefix = '/mypath'], order = 0]",
            "[[RewritePath /books = '/APIC/DATA.json'], order = 0]"
        ],
        "uri": "http://10.10.2.250:8080",
        "order": 0
    }
]
```

- 참고

11. Actuator API

application.properties. management.endpoint.gateway.enabled=true management.endpoints.web.exposure.include=gateway To retrieve the GatewayFilter factories applied to routes, make a GET request to /actuator/gateway/routefilters. The resulting response is similar to the following: { "[[email protected] configClass = AbstractNameValueGatewayFilterFactory.NameValueConfig]": null, "[[email protected] configClass = Object]": null, "[[email protected]

https://cloud.spring.io/spring-cloud-gateway/multi/multi__actuator_api.html