http://localhost:8072/actuator/gateway/routes

it is getway server to check what are the sevservices running over it.

[{"predicate":"Paths: [/gatewayserver/**], match trailing slash:
true","metadata":{"jmx.port":"52498","management.port":"8072"},"route_id":"ReactiveCompositeDiscoveryClient
_GATEWAYSERVER","filters":["[[RewritePath /gatewayserver/?(?<remaining>.*) = '/${remaining}'], order =
1]"],"uri":"lb://GATEWAYSERVER","order":0},{"predicate":"Paths: [/accounts/**], match trailing slash:
true","metadata":{"jmx.port":"52483","management.port":"8081"},"route_id":"ReactiveCompositeDiscoveryClient
_ACCOUNTS","filters":["[[RewritePath /accounts/?(?<remaining>.*) = '/${remaining}'], order =
1]"],"uri":"lb://ACCOUNTS","order":0}]

Using routing server we can call account application

POSTMAN:
http://localhost:8072/accounts/myaccount

Request:
```
{
   "customerId": 1
}
```

Response:
```
{
   "customerId": 1,
   "accountNumber": 100928281,
   "accountType": "saving",
   "branchAddress": "LVS",
   "createDt": "2021-05-16"
}
```

Implementing custom routing using spring cloud getway

```
@Bean
public RouteLocator myRoutes(RouteLocatorBuilder builder) {
return builder.routes()
.route(p -> p
.path("/bankAPI/accounts/**")
.filters(f -> f.rewritePath("/bankAPI/accounts/(?<segment>.*)","/${segment}")
.addResponseHeader("X-Response-Time",new Date().toString()))
.uri("lb://ACCOUNTS")).
route(p -> p
.path("/bankAPI/loans/**")
.filters(f -> f.rewritePath("/bankAPI/loans/(?<segment>.*)","/${segment}")
.addResponseHeader("X-Response-Time",new Date().toString()))
.uri("lb://LOANS")).
route(p -> p
.path("/bankAPI/cards/**")
.filters(f -> f.rewritePath("/bankAPI/cards/(?<segment>.*)","/${segment}")
.addResponseHeader("X-Response-Time",new Date().toString()))
.uri("lb://CARDS")).build();
}
```

POSTMAN

http://localhost:8072/bankAPI/accounts/myaccount


```
{
   "customerId": 1
}

{
   "customerId": 1,
   "accountNumber": 100928281,
   "accountType": "saving",
   "branchAddress": "LVS",
   "createDt": "2021-05-16"
}
```


http://localhost:8072/bankAPI/accounts/myCustomerDetails


```
{
   "customerId": 1
}

{
   "accounts": {
      "customerId": 1,
      "accountNumber": 100928281,
      "accountType": "saving",
      "branchAddress": "LVS",
      "createDt": "2021-05-16"
   },
   "loans": [
      {
         "loanNumber": 3,
         "customerId": 1,
         "startDt": "2021-02-14",
         "loanType": "Home",
         "totalLoan": 50000,
         "amountPaid": 10000,
         "outstandingAmount": 40000,
         "createDt": "2018-02-14"
      },
   ],
   "cards": [
      {
         "cardId": 1,
         "customerId": 1,
         "cardNumber": "4565XXXX4656",
         "cardType": "Credit",
         "totalLimit": 10000,
         "amountUsed": 500,
         "availableAmount": 9500,
         "createDt": "2022-07-03"
      },
      {
         "cardId": 2,
```

```
        "customerId": 1,
        "cardNumber": "3455XXXX8673",
        "cardType": "Credit",
        "totalLimit": 7500,
        "amountUsed": 600,
        "availableAmount": 6900,
        "createDt": "2022-07-03"
    },
    {
        "cardId": 3,
        "customerId": 1,
        "cardNumber": "2359XXXX9346",
        "cardType": "Credit",
        "totalLimit": 20000,
        "amountUsed": 4000,
        "availableAmount": 16000,
        "createDt": "2022-07-03"
    }
  ]
}
```


http://localhost:8072/bankAPI/loans/myLoans


```
{
  "customerId": 1
}
[
  {
    "loanNumber": 3,
    "customerId": 1,
    "startDt": "2021-02-14",
    "loanType": "Home",
    "totalLoan": 50000,
    "amountPaid": 10000,
    "outstandingAmount": 40000,
    "createDt": "2018-02-14"
  },
  {
        ----
  }
 ]
```


http://localhost:8072/bankAPI/cards/myCards


```
[
  {
    "cardId": 1,
    "customerId": 1,
    "cardNumber": "4565XXXX4656",
    "cardType": "Credit",
    "totalLimit": 10000,
    "amountUsed": 500,
    "availableAmount": 9500,
```

```
      "createDt": "2022-07-03"
   },
  {
          -------
  }
]
```


Adding @RequestHeader("eazybank-correlation-id")

update all the important classes like AccountsController.java to accept the @RequestHeader("eazybank-correlation-id") String correlationid as input inside the method parameters.

http://localhost:8072/bankAPI/accounts/myCustomerDetails


```
{
   "customerId": 1
}
```


Response:


```
{
   "accounts": {
      "customerId": 1,
      "accountNumber": 100928281,
      "accountType": "saving",
      "branchAddress": "LVS",
      "createDt": "2021-05-16"
   },

   ---
   ---
}
```


Console debug: eazybank-correlation-id getting from request header


```
------------------ff5c3077-cf5c-4056-8c5f-ad6d9280b967
EazyBank-correlation-id generated in tracing filter: 3a0745ec-6d6c-453a-b128-18c90f7b8d93.
```