

Ozan Exchange

Table of Contents

Exchange Api	1
ExchangeApi Rate - GET	1
ExchangeApi Rate - Error Handling	3
ExchangeApi Conversion - GET	4
Exchange Conversion - Error Handling	6
ExchangeApi Conversion - POST	7
ExchangeApi Conversion ErrorHandler	10
Exchange Conversion Api	11
Conversion By Transaction	11
Conversion By Transaction - Error Handling	12
Conversion By Date	13
Conversion By Date - Error Handling	15
Conversion By Transaction And Date	17
Conversion By Transaction And Date - Error Handling	18
System Design	20
Service Providers	20
Cache	23
Testing	24
Tech Stacks	25

This application is created for demo purpose.

maintainer : mithatkonuk@gmail.com

Exchange Api

ExchangeApi service provide exchange rate between currencies and support also conversion

ExchangeApi Rate - GET

Provide rate of between currencies

GET /exchangeApi/exchange

Query parameters

Parameter	Type	Optional	Description
currency_pair	String	false	Represent of Currencies,format[EUR, TRY]

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of rate
error	Object	true	Represent of Response Error
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code
error.errorCode.description	String	true	Represent of ErrorCode description

Example request

```
$ curl 'http://localhost:8080/exchangeApi/exchange?currency_pair=EUR%2CTRY' -i -X GET \
-H 'Accept: application/json;charset=UTF-8'
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 223

{
  "data" : {
    "base" : "EUR",
    "symbol" : "TRY",
    "date" : "2020-12-17T11:08:23.040+00:00",
    "rate" : 9.5219
  },
  "error" : {
    "errorCode" : {
      "errorCode" : 0,
      "description" : null
    }
  }
}
```

ExchangeApi Rate - Error Handling

ExchangeApi GET ErrorHandler with generic defined Response object.

GET /exchangeApi/exchange

Query parameters

Parameter	Type	Optional	Description
currency_pair	String	false	Represent of current pair , format : EUR,TRY

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of exchange rate
error	Object	true	Represent of response error
error.errorCode	Object	true	Represent of response error code
error.errorCode.code	Integer	true	Represent of errorcode as code
error.errorCode.description	String	true	Represent of errorcode description

Example request

Request with invalid parameter as seen below

```
$ curl 'http://localhost:8080/exchangeApi/exchange?currency_pair=unknow_pair' -i -X GET \
-H 'Accept: application/json;charset=UTF-8'
```

Example response

Response will be with error , and error description

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Content-Length: 128

{
  "error" : {
    "errorCode" : {
      "errorCode" : 4,
      "description" : "Given currency pair is not acceptable for exchange rate"
    }
  }
}
```

ExchangeApi Conversion - GET

Provide conversion from base to symbol currency via given amount with rate which is defined in system.

GET [/exchangeApi/exchange/conversion](#)

Query parameters

Parameter	Type	Optional	Description
base	String	false	Represent of source currency
symbol	String	false	Represent of target currency
amount	Decimal	false	Amount should be positive or zero.
detail	Boolean	true	Default value: 'false'.

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of data
error	Object	true	Represent of Response Error
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of Error Codes
error.errorCode.description	String	true	Represent of Error Code Description

Example request

```
$ curl
'http://localhost:8080/exchangeApi/exchange/conversion?base=EUR&symbol=TRY&amount=10.0
&detail=true' -i -X GET -H 'Accept: application/json;charset=UTF-8'
```

without detail

```
$ curl
'http://localhost:8080/exchangeApi/exchange/conversion?base=EUR&symbol=TRY&amount=10.0
' -i -X GET -H 'Accept: application/json;charset=UTF-8'
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 342

{
  "data" : {
    "base" : "EUR",
    "symbol" : "TRY",
    "amount" : 10.0,
    "conversion" : 95.21900000000001,
    "date" : "2020-12-17T11:08:22.923+00:00",
    "rate" : 9.5219,
    "transaction" : "6c8f65dd-ce0c-4bbe-9531-eec50c379653"
  },
  "error" : {
    "errorCode" : {
      "errorCode" : 0,
      "description" : null
    }
  }
}
```

without detail

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 100

{
  "data" : {
    "conversion" : 95.21900000000001,
    "transaction" : "6c8f65dd-ce0c-4bbe-9531-eec50c379653"
  },
  "error" : {
    "errorCode" : {
      "errorCode" : 0,
      "description" : null
    }
  }
}
```

Exchange Conversion - Error Handling

GET /exchangeApi/exchange/conversion

Provide generic response error handling

Query parameters

Parameter	Type	Optional	Description
base	String	false	Represent of source currency
symbol	String	false	Represent of target currency
amount	Decimal	false	Amount should be positive or zero.
detail	Boolean	true	Default value: 'false'.

Response fields

Path	Type	Optional	Description
data	Object	true	Represent Empty data
error	Object	true	Represent of Response Error
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code

Path	Type	Optional	Description
error.errorCode.description	String	true	Represent of ErrorCode description

Example request

```
$ curl
'http://localhost:8080/exchangeApi/exchange/conversion?base=test&symbol=test&amount=10.0&detail=true' -i -X GET
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 159

{
  "error" : {
    "errorCode" : {
      "errorCode" : 1,
      "description" : "Given Exchange source / Given Exchange target is not supported"
    }
  }
}
```

ExchangeApi Conversion - POST

Exchange Rate Api Conversion service provide conversion amount of currencies between given base and symbol.

POST /exchangeApi/exchange/conversion

Request fields

Request object defined as Conversion.java

```

public class Conversion
{
    @NotNull( message = "base must be represent" )
    private String base;

    @NotNull( message = "symbol must be represent" )
    private String symbol;

    @PositiveOrZero( message = "The amount of currency should be positive" )
    private Double amount;

    private boolean detail;
}

```

Path	Type	Optional	Description
base	String	false	Source currency
symbol	String	false	Target currency
amount	Decimal	true	The amount of base currency should be positive
detail	Boolean	true	Represent of detail will include or not

Response fields

Path	Type	Optional	Description
data	Object (Generic)	true	Represent of data
error	Object	true	ResponseError
error.errorCode	Object	true	ErrorCode
error.errorCode.code	Integer	true	value of error code
error.errorCode.description	String	true	ErrorCode description

Example request

In this example requesting EUR to TRY to get rate with detail.

```

$ curl 'http://localhost:8080/exchangeApi/exchange/conversion' -i -X POST \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json' \
  -d '{"base":"EUR","symbol":"TRY","amount":10.0,"detail":true}'

```


Example response

Simple Response as seen below. Requesting date, currencies, rate etc. according to setting detail.

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 342

{
  "data" : {
    "base" : "EUR",
    "symbol" : "TRY",
    "amount" : 10.0,
    "conversion" : 95.21900000000001,
    "date" : "2020-12-17T11:08:22.641+00:00",
    "rate" : 9.5219,
    "transaction" : "aeb59b83-96d8-401a-a66a-dcfb5dbbcbd3"
  },
  "error" : {
    "errorCode" : {
      "errorCode" : 0,
      "description" : null
    }
  }
}
```

Detail is not setted

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 342

{
  "data" : {
    "conversion" : 95.21900000000001,
    "transaction" : "aeb59b83-96d8-401a-a66a-dcfb5dbbcbd3"
  },
  "error" : {
    "errorCode" : {
      "errorCode" : 0,
      "description" : null
    }
  }
}
```

ExchangeApi Conversion ErrorHandling

POST /exchangeApi/exchange/conversion

Request fields

Path	Type	Optional	Description
base	String	false	Represent of base currency
symbol	String	false	Represent of target currency
amount	Decimal	true	The amount of base currency should be positive.
detail	Boolean	true	more information

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of data
error	Object	true	Represent of Respons Error
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Value of ErrorCode
error.errorCode.description	String	true	Value of Error Code description

Example request

```
$ curl 'http://localhost:8080/exchangeApi/exchange/conversion' -i -X POST \  
-H 'Content-Type: application/json' \  
-d '{"base":"test","symbol":"TRY","amount":10.0,"detail":true}'
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 159
```

```
{
  "error" : {
    "errorCode" : {
      "errorCode" : 1,
      "description" : "Given Exchange source / Given Exchange target is not supported"
    }
  }
}
```

Exchange Conversion Api

Conversion By Transaction

Conversion History provided by transaction,simple client can search with transaction id.

GET /conversionApi/conversion/transaction

Query parameters

Parameter	Type	Optional	Description
transaction_id	String	true	

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of data
error	Object	true	Represent of ResponseError
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code
error.errorCode.description	String	true	Represent of ErrorCode Description

Example request

```
$ curl
'http://localhost:8080/conversionApi/conversion/transaction?transaction_id=ab80de0c-
3b54-40b1-a5e7-3d0dcb47f606' -i -X GET \
-H 'Content-Type: application/json' \
-H 'Accept: application/json'
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 318

{
  "data" : {
    "base" : "EUR",
    "symbol" : "TRY",
    "amount" : 10.0,
    "conversion" : 95.21900000000001,
    "date" : "2020-12-17",
    "rate" : 9.5219,
    "transaction" : "ab80de0c-3b54-40b1-a5e7-3d0dcb47f606"
  },
  "error" : {
    "errorCode" : {
      "code" : 0,
      "description" : null
    }
  }
}
```

Conversion By Transaction - Error Handling

Transaction id is not acceptable , so it will return response as error, transaction is defined as UUID in the ozan exchange system.

GET /conversionApi/conversion/transaction

Query parameters

Parameter	Type	Optional	Description
transaction_id	String	true	Represent of transaction for each call exchange rate, (string type as UUID)

Response fields

Path	Type	Optional	Description
data	Object	true	
error	Object	true	
error.errorCode	Object	true	
error.errorCode.code	Integer	true	
error.errorCode.description	String	true	

Example request

```
$ curl
'http://localhost:8080/conversionApi/conversion/transaction?transaction_id=transaction
' -i -X GET \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json'
```

Example response

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 123

{
  "error" : {
    "errorCode" : {
      "code" : 4,
      "description" : "Given Parameter is not acceptable"
    }
  }
}
```

Conversion By Date

Conversion provide by date , client can search by date and response will be paging

GET /conversionApi/conversion/date

Query parameters

Parameter	Type	Optional	Description
created	String	true	Represent of date ,format (YYYY-MM-dd)

Parameter	Type	Optional	Description
offset	Integer	true	offset should be positive or zero.Default value: '0'.
pageSize	Integer	true	page size should be positive or zero.Default value: '10'.

Response fields

Path	Type	Optional	Description
data	Object	true	
error	Object	true	
error.errorCode	Object	true	
error.errorCode.code	Integer	true	
error.errorCode.description	String	true	

Example request

```
$ curl 'http://localhost:8080/conversionApi/conversion/date?created=2020-12-17' -i -X GET \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json'
```

Example response

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 908

```
{
  "data" : {
    "data" : [ {
      "base" : "EUR",
      "symbol" : "TRY",
      "amount" : 10.0,
      "conversion" : 95.21900000000001,
      "date" : "2020-12-17",
      "rate" : 9.5219,
      "transaction" : "ab80de0c-3b54-40b1-a5e7-3d0dcb47f606"
    }, {
      "base" : "EUR",
      "symbol" : "TRY",
      "amount" : 10.0,
      "conversion" : 95.21900000000001,
      "date" : "2020-12-17",
      "rate" : 9.5219,
      "transaction" : "267a8a0e-4e78-4826-a2e8-2bf361ee0ade"
    }, {
      "base" : "EUR",
      "symbol" : "TRY",
      "amount" : 10.0,
      "conversion" : 95.21900000000001,
      "date" : "2020-12-17",
      "rate" : 9.5219,
      "transaction" : "948f6c60-37f2-420e-8cad-3c1e6fd5fdda"
    } ],
    "totalSize" : 4,
    "pageSize" : 10,
    "pageNumber" : 0,
    "totalPageSize" : 1
  },
  "error" : {
    "errorCode" : {
      "code" : 0,
      "description" : null
    }
  }
}
```

Conversion By Date - Error Handling

GET /conversionApi/conversion/date

Query parameters

Parameter	Type	Optional	Description
created	String	true	Represent of Date , format [yyyy-MM-dd]
offset	Integer	true	offset should be positive or zero.Default value: '0'.
pageSize	Integer	true	page size should be positive or zero.Default value: '10'.

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of data
error	Object	true	Represent of ResponseError
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code
error.errorCode.description	String	true	Represent of ErrorCode description

Example request

```
$ curl 'http://localhost:8080/conversionApi/conversion/date?created=03-09-2020' -i -X GET \
-H 'Content-Type: application/json' \
-H 'Accept: application/json'
```

Example response


```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 151
```

```
{
  "error" : {
    "errorCode" : {
      "code" : 5,
      "description" : "Given Parameter format problem exist,date format [YYYY-MM-DD]"
    }
  }
}
```

Conversion By Transaction And Date

Provide Simple service to get conversion history with transaction , date .Transaction and date is optional

GET /conversionApi/conversion

Query parameters

Parameter	Type	Optional	Description
created	String	true	Represent of Date
transaction_id	String	true	Represent of Transaction
offset	Integer	true	offset size should be positive or zero.Default value: '0'.
pageSize	Integer	true	page size should be positive or zero.Default value: '10'.

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of data
error	Object	true	Represent of ResponseError
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code

Path	Type	Optional	Description
error.errorCode.description	String	true	Represent of ErrorCode description

Example request

```
$ curl 'http://localhost:8080/conversionApi/conversion?created=2020-12-17&transaction_id=267a8a0e-4e78-4826-a2e8-2bf361ee0ade' -i -X GET \
-H 'Content-Type: application/json' \
-H 'Accept: application/json'
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 446

{
  "data" : {
    "data" : [ {
      "base" : "EUR",
      "symbol" : "TRY",
      "amount" : 10.0,
      "conversion" : 95.21900000000001,
      "date" : "2020-12-17",
      "rate" : 9.5219,
      "transaction" : "267a8a0e-4e78-4826-a2e8-2bf361ee0ade"
    } ],
    "totalSize" : 1,
    "pageSize" : 10,
    "pageNumber" : 0,
    "totalPageSize" : 1
  },
  "error" : {
    "errorCode" : {
      "code" : 0,
      "description" : null
    }
  }
}
```

Conversion By Transaction And Date - Error Handling

GET /conversionApi/conversion

Query parameters

Parameter	Type	Optional	Description
created	String	true	Date
transaction_id	String	true	Transaction Id
offset	Integer	true	offset size should be positive or zero.Default value: '0'.
pageSize	Integer	true	page size should be positive or zero.Default value: '10'.

Response fields

Path	Type	Optional	Description
data	Object	true	Represent of Data
error	Object	true	Represent of Response Error
error.errorCode	Object	true	Represent of ErrorCode
error.errorCode.code	Integer	true	Represent of ErrorCode code
error.errorCode.description	String	true	Represent of ErrorCode description

Example request

```
$ curl 'http://localhost:8080/conversionApi/conversion?created=03-09-1971&transaction_id=948f6c60-37f2-420e-8cad-3c1e6fd5fdda' -i -X GET \
-H 'Content-Type: application/json' \
-H 'Accept: application/json'
```

Example response

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 151
```

```
{
  "error" : {
    "errorCode" : {
      "code" : 5,
      "description" : "Format problem exist on given Parameter,date format [YYYY-MM-DD]"
    }
  }
}
```

System Design

Service Providers

ForgienExchangeProvider

Simple Exchange Provider Interface defined to give functionality on providers.Extendable interface provide 'getExchange' method to handle external service calls.

ForgienExchangeProvider.java

```
public interface ForgienExchangeProvider
{
    @RequestMapping( method = RequestMethod.GET,
        value = "${forgien_exchange_providers.feign.exchange_path}" )
    ExternalExchange getExchange( @RequestParam( "base" ) String base,
        @RequestParam( "symbols" ) String symbols );
}
```

OzanExchange support two forgienExchangeProviders and configurations defined as property file

RateApiExternalExchangeProvider

Simple ForgienExchangeProvider implementation which wrapper 'RestTemplate' to handle external service calls. This Provider can be disabled by application.properties file.

```

public class RateApiExternalExchangeProvider implements ForgienExchangeProvider
{
    private static final Logger logger =
        LoggerFactory.getLogger(RateApiExternalExchangeProvider.class);

    private final RestTemplate restTemplate;
    private final OzanRateProviderRestConfiguration configuration;
    private String URI;

    public RateApiExternalExchangeProvider( RestTemplate restTemplate,
        OzanRateProviderRestConfiguration configuration )
    {
        this.restTemplate = restTemplate;
        this.configuration = configuration;
        this.URI = configuration.getUrl() + this.configuration.getExchangePath();
    }

    @Cacheable( value = "${forgien.service.provider.request.cache.name}",
        keyGenerator = "forgienCacheKeyGenerator" )
    @Override
    public ExternalExchange getExchange( String base, String symbols )
    {
        this.URI = this.URI.replace("{base}", base).replace("{symbols}", symbols);
        logger.
            info("[external-service ( " + configuration.getName() + ")]" +
                " - Generated url : " + this.URI);

        ExternalExchange externalExchange =
            restTemplate.getForObject(this.URI, ExternalExchange.class);

        logger.info("[external-service-response] - " + externalExchange.toString());

        return externalExchange;
    }

    @CacheEvict( allEntries = true,
        cacheNames = { "${forgien.service.provider.request.cache.name}" } )
    @Scheduled( fixedDelayString = "${forgien.service.provider.request.cache.ttl}" )
    public void cacheEvict()
    {
        logger.info("Cahce is refreshed [cahce-evict-RateApiExternalExchangeProvider]
");
    }
}

```

RateExchangeApiProvider

Simple ForgienExchangeProvider implementation which wrapper 'FeignClient' to handle external service.

RateExchangeApiProvider.java

```
@Component( "rateApiProvider" )
public class RateExchangeApiProvider implements ForgienExchangeProvider
{

    private static final Logger logger = LoggerFactory.getLogger
(RateExchangeApiProvider.class);

    private ForgienExchangeProvider forgienExchangeProvider;

    @Value( "${forgien_exchange_providers.feign.name}" )
    private String providerName;

    @Autowired
    public RateExchangeApiProvider( @Qualifier(
"${forgien_exchange_providers.feign.name}" )
        ForgienExchangeProvider forgienExchangeProvider )
    {
        this.forgienExchangeProvider = forgienExchangeProvider;
    }

    @Cacheable( value = "${forgien.service.provider.request.cache.name}",
        keyGenerator = "forgienCacheKeyGenerator" )
    @Override
    public ExternalExchange getExchange( String base, String symbols )
    {
        try
        {
            ExternalExchange externalExchange = forgienExchangeProvider.getExchange
(base, symbols);

            logger.info("[external-service (" + providerName + ")] : " +
                externalExchange.toString());

            return externalExchange;
        }
        catch( FeignException e )
        {
            throw
                new ExternalServiceException(ErrorCode
                    .EXTERNAL_SERVICE_PROVIDER.EXTERNAL_RESOURCE_EXCHANGE_NOT_FOUND,
                    "Currently External service is not supporting given exchange");
        }
    }
}
```

```

@CacheEvict( allEntries = true,
              cacheNames = { "${forgien.service.provider.request.cache.name}" } )
@Scheduled( fixedDelayString = "${forgien.service.provider.request.cache.ttl}" )
public void cacheEvict()
{
    logger.info("Cahce is refreshed [cahce-evict-RateExchangeApiProvider]");
}
}

```

Cache

OzanExchange Application has ability to cache external service rates and expire with specific time-to-live configuration which is defined on application.properties.

OzanExchangeCacheConfiguration.java

```

@Configuration
public class OzanExchangeCacheConfiguration
{

    @Value( "${forgien.service.provider.request.cache.name}" )
    private String serviceProviderCacheName;

    @Bean
    public CacheManager cacheManager()
    {
        SimpleCacheManager cacheManager = new SimpleCacheManager();
        Cache ServiceProviderRequestCache = new ConcurrentMapCache
(serviceProviderCacheName);
        cacheManager.setCaches(Arrays.asList(ServiceProviderRequestCache));
        return cacheManager;
    }

    @Bean( "forgienCacheKeyGenerator" )
    public KeyGenerator keyGenerator()
    {
        return new ForgienServiceProviderCacheKeyGenerator();
    }
}

```

Cache Key Generator policy

```
public class ForgienServiceProviderCacheKeyGenerator implements KeyGenerator
{
    @Override
    public Object generate( Object o, Method method, Object... objects )
    {
        String[] params = (String[])objects;

        return params[0] + "-" + params[1];
    }
}
```

Testing

Test implementations located , link: /Users/mithat.konuk/Desktop/Exchange/target/generated-snippets/../../src/test/java/com/ozan/exchange/resource/

Unit Testing

- Each functionality Test created on '/test' folder

Integration Testing

- Integration Test include external call with MockMvc on random port to process each scenario

Swagger Document

OzanExchange Application also support swagger , here is demo link: <http://localhost:8080/swagger-ui/index.html#/>

Central Logging

OzanExchange support two logging profile , dev and prod

dev Mode :

dev profile support logging to local directory '/tmp/ozan_exchange/logs'

prod Mode :


```
prod profile support logging to central
loggly central logging
email:galami2745@febeks.com -- temporary mail taken
account name : ozanExchange
password : Password_123456
URL : https://ozanexchange.loggly.com/dashboards/dashboard/2846642b-b55a-4bfa-9a0e-1979962e2fc8/view
```

Tech Stacks

- Spring boot ref : <https://spring.io/projects/spring-boot>
- H2 database (schema.sql)
- Spring AOP ref: <https://docs.spring.io/spring-framework/docs/2.5.x/reference/aop.html>
- Spring Cache ref : <https://spring.io/guides/gs/caching/>
- Spring JPA ref : <https://spring.io/projects/spring-data-jpa>
- Lombok ref: <https://projectlombok.org/>
- asciidoctor ref: <https://asciidoctor.org/>
- LogBack ref: <http://logback.qos.ch/>