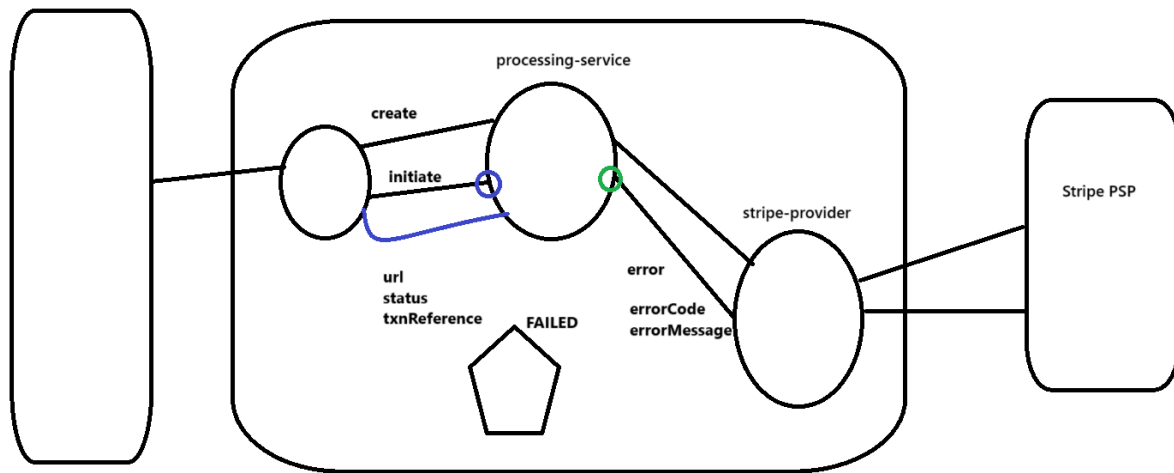


DIAGRAM:



LIVE NOTES:

Week6 Day1:

End-to-end features / testing - Payment Status Management System.

DB update.

Insert record in DB using NamedParameterJdbcTemplate

1. createTxn in DB.

2. getTxnByReference

How to pass parameters to query

`new BeanPropertySqlParameterSource(txnEntity)`

field names & query param names should be same.

more named parameters are needed.

```
MapSqlParameterSource params = new MapSqlParameterSource();
params.addValue("txnReference", txnReference);
```

update - insert - delete
nPJT.update

select
- queryForObject
- how to handle response: BeanPropertyRowMapper (TEntity.class)

Entity => DTO
Write new converters & add to AppConfig ModelMapper configuration.

```
modelMapper.addMappings(new PropertyMap<TransactionEntity, TransactionDTO>() {
    @Override
    protected void configure() {
        using(new PaymentMethodEnumIdToStringConverter()
            ).map(source.getPaymentMethodId(), destination.getPaymentMethod());
        using(new PaymentTypeEnumIdToStringConverter()
            ).map(source.getPaymentTypeId(), destination.getPaymentType());
        using(new ProviderEnumIdToStringConverter()
            ).map(source.getProviderId(), destination.getProvider());
        using(new TransactionStatusEnumIdToStringConverter()
            ).map(source.getTxnStatusId(), destination.getTxnStatus());
    }
});
```

=====

1. createTxn - insert --
CREATED

---update DB - update DB query

PaymentServiceImpl

2. INITIATED

update query
INITIATED

Whatever data needed, update it in TransactionDTO

INITIATED

paymentStatusServiceImpl.processStatus()

factory dp, based on status take handler

handler.processStatus

3. PENDING

when we get success response from provider with url

PENDING

providerReference

3. FAILED

when API call to stripe-provider fails

FAILED

errorCode

errorMessage

1 method with 1 sql query

update status, providerreference, errocode, errorMessage

transactionDTO

How do you call updateTxnDetails method.

- PaymentServiceImpl

set relavant fields to TransactionDTO object

statusService.process

All casese, INITIATED, PENDING, FAILED

- No change in PaymentStatusServiceImpl

- IN all the status handlers

TransactionDao
InitiatedStatusHandler
PendingStatusHandler
FailedStatusHandler

Test INITIATED Status update (with breakpoints)

=====

Making Http Call

with valid request data.

API Request structure of the 2 endpoints

createPayment

- have the request structure needed to successfully save txn object into DB.

initiatePayment

- in request, capture all the data needed for making call to stripe-provider service
- What data stripe-provider-service needs?
- does processing-service already have that data.
- If you dont have the data, then can validationservice provide you that info?
- if its relevant for validation-service to provide that data, then we define it as a

request structure for initiatePayment

1. save txn in DB
2. Make successful API call to stripe-provider => stripePSP

cart page:

PaymentController

CreateTxn:

amount: 2.8k
currency: INR

5 products

1k INR
1.5k INR
300 INR

Stripe-provider (StripePSP)

```
public class CreatePaymentReq {
```

```
    private String successUrl;  
    private String cancelUrl;
```

```
    private List<LineItem> lineItems;
```

```
}
```

```
public class LineItem {
```

```
    private int quantity;  
    private String currency;  
    private String productName;  
    private int unitAmount;
```

```
}
```

in processing-service, initiatePayment

request structure should have everything that stripe-provider needs

.pojo

InitiatePaymentReq

Item

.dto

InitiatePaymentDTO

Making Http Call (with valid request data)

HttpServiceEngine (RestClient)

processing-service got the response.
process the response.

HttpServiceEngine:

- if we get valid(success/failure) response from external service then return the response object back to the service layer.
- service layer process the response.
- if we dont get valid response, then we simply throw exception -
UNABLE_TO_CONNECT

How the response processing in PaymentServiceImpl should be:

1. success response
 2. error response
 3. exception can be thrown.
- Custom Exception handling is needed.
 - create custom Exception having errorCode, message, http status
 - when identified error situation, then throw this custom exception with proper values.
 - GlobalExceptionHandler handles this custom exception.
 - prepares the ErrorResponse structure & returns back.
 - Utilize ErrorCodeEnum
 - Also handle generic Exception.

stripe-provider service

returned success response.
stripe PSP ID (update in DB)
redirectionUrl (return back to the invoker)
PENDING (update in DB)

When stripe-provider is returning response, its sending as payment_status. in processing-service, we should have as paymentStatus. However gson will not be able to map, if string name & field names are different. so it give @SerializedName to handle correct mapping.

response string => java object

what is success for us?

get 201 with url
everything else is failed.

Failure scenarios...

when processing-service => stripe-provider > StripePSP
if error happens, we expect error response with valid
errorCode, errorMessage & httpStatus

1. update txn in DB as FAILED status with valid errorCode & errorMessage
2. what error response will you return to invoker (validation)

the error which we receive from StripPSP, we need to save in our DB, but not pass as
error response to the API.

Other than this, any other error which we receive from Strip-provider, needs to be saved
in DB & returned back in API response.

whatever error comes from stripe-provider, we save in DB.

UC1: StripePSP has error

we will not pass quantity in processing-service API
quantity will go as null to stripe-provider
quantity will go as null to stripe-psp
StripePSP will return 4xx error
save the reason in DB, but override the response that we return.

PaymentServiceImpl:

HttpServiceEngine
Success
Failure
Exception

=====

When we encounter failure:

1. Save in DB
status FAILED
errorCode & errorMessage

2. return the error details to invoker.

1. Whatever error stripe-provider returns, that has to be save in DB.
with FAILED status & received errorCode & errorMessage

2. If the error is coming form StripePSP, then override with static errorCode & errorMessage in processing-service.

dont give StripePSP errors to invoker (validation => ClientApp). Client APp is not aware of internal technical implementation of stripe system.

How to test failure:

Dont pass quantity in input request.

UC2: trigger problem at stripe-provider (Not from stripePSP)

Expectation:

The errorcode & errorMessage returned by stripe-provider, should be save in DB & return in api response of processing-service

Actual:

same errorCode & errorMessage as saved in DB & returned in response.

success

failed

exception

in HttpServiceEngine, if we fail to connect to stripe-psp, then exception would be thrown.

Handle it & update txn in DB with FAILED status & valid errorcode & errorMessage

