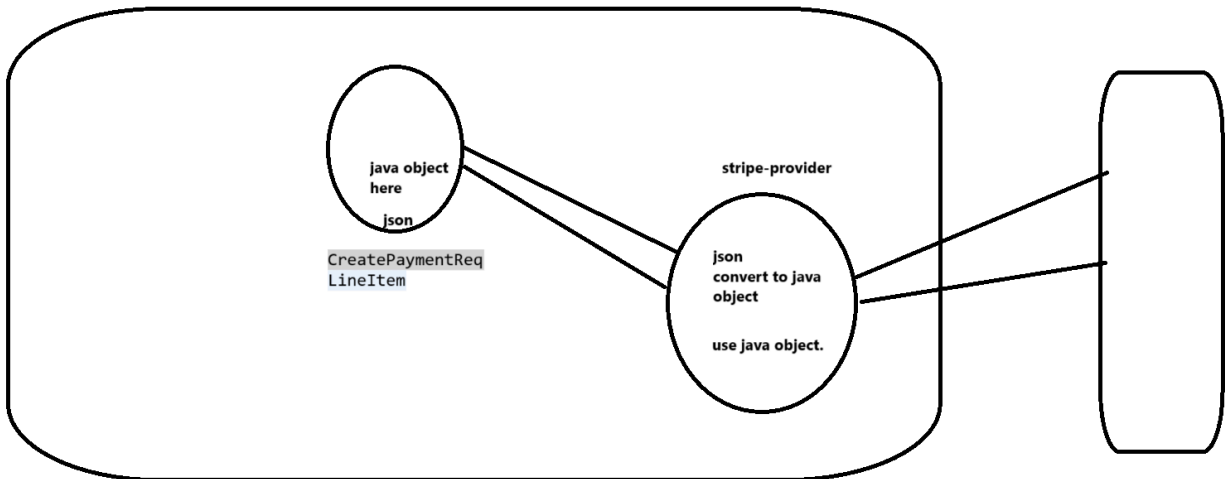


DIAGRAMS

Processing service request structure while making call to stripe-provider service



LIVE NOTES

W5D4:

Continue building Payment Status Management system.

For performing create operation, do you need any additional data??

UUID

A UUID (Universally Unique Identifier) is a 128-bit value used to generate unique references across systems. It ensures uniqueness without relying on centralized databases. UUIDs are commonly used for database keys, transaction IDs, and distributed systems.

```
UUID uuid = UUID.randomUUID();
```

Let's use Enum.

CREATED

no hardcoding
define it as constant.

CREATED - 1
INITIATED - 2
PENDING - 3
SUCCESS - 4
FAILED - 5

We have TransactionDTO ready to be inserted into DB.

=====

```
public interface TxnStatusHandler {  
    public TransactionDTO processStatus(TransactionDTO txnDto);  
}
```

& create 5 Status Handlers
with TODO empty b.logic.

PaymentStatusService
using the current txnStatus, get the handler from factory & call process method.

Creating a Factory
switching based on Enum & returning handler.

load the handler from Spring Container using ApplicationContext. (Dependency lookup)

@RequiredArgsConstructor
& final fields, then you don't have to explicitly define constructor.

==

we have status as String, but we need to get corresponding Enum back.

===

To Make API call to stripe-provider-service

.http
copies 2 files from stripe-provider-service

HttpRequest
HttpServiceEngine

stripe-provider-service APIs, we are not adding a security layer.
because these apis will be consumed by internal service like processing.
at validation-service security logic implemented, since its exposed to client app.

processing-service needs to call stripe-provider-service

```
{  
  "successUrl": "https://example.com/success",  
  "cancelUrl": "https://example.com/cancel",  
  "lineItems": [  
    {  
      "quantity": 5,  
      "currency": "EUR",  
      "productName": "Mechanical Keyboard",  
      "unitAmount": 1000  
    }  
  ]  
}
```

you should create Java objects which will represent this request structure. Then populate data to that & convert to json & pass it further.

```
.pojo  
.dto  
.<service-name>
```

```
.stripeprovider  
    StripeProviderCreatePaymentReq  
    StripeProviderLineItem
```