

Realization Development

INTERNSHIP TELENET

MATTHIAS HEYLEN

Bachelor of Applied Computer Science

Option: Application Development

Academic Year 2024-2025
Campus Geel, Kleinhoefstraat 4, 2440 Geel

Contents

Introduction	2
Development Story	3
Manual testing development story	6
Sprint board	7
Pipelines.....	8
[DEV] [Achieve SonarQube coverage for different modules]	9
Cnp-service (coverage)	10
Task Request Handler (coverage)	52
CNP-Inbound-Message-Handler (coverage).....	56
Bff-portingrequest-datafeed (coverage).....	68
CRDC-Outbound-message-handler (coverage).....	69
NCBSS-Outbound-message-handler (coverage)	70
CRDC-Inbound-message-handler (Coverage)	74
Complex-message-handler (coverage)	75
Message transformers	76
NCOSS inbound message transformer	76
NCBSS inbound message transformer	86
CRDC inbound message transformer	96
Order Seach Export button	114
Summary	125

Introduction

During my internship at Telenet, I had the opportunity to engage in various aspects of software development within the Number Porting Tribe, specifically the TOF squad. My work contributed to the modernization of Telenet's number porting systems for both mobile and fixed numbers. These systems, previously reliant on outdated infrastructure, were rebuilt using modern technologies like AWS to enhance reliability, scalability, and maintainability.

As part of the development process, I worked on backend services and frontend features, gaining hands-on experience with frameworks like Spring Boot for backend APIs and Vue.js and React for frontend interfaces. This work often involved resolving bugs, implementing new functionalities, and improving the overall code quality of the system. By collaborating closely with developers, testers, analysts, and the product owner, I was able to contribute effectively to the project's progress while adhering to agile methodologies like sprints, daily stand-ups, and code reviews.

One of the key areas of focus was the fixed number porting system, where I worked on specific development stories. These tasks included creating message transformers to convert XML to JSON, building serverless functions with AWS Lambda, and refining infrastructure through Terraform. Additionally, I contributed to frontend enhancements by developing features like a CSV export functionality, which streamlined certain workflows.

Through my work at Telenet, I gained practical experience in applying modern development tools and techniques to real-world projects. This experience not only strengthened my technical skills but also provided valuable insights into teamwork, agile practices, and delivering high-quality software solutions in a professional environment.

Development Story

For my first development story I had change the description during a scenario when Artilium sends reject code 199. The issue was that description currently was an open text. I managed to solve this scenario by changing one line of code so the description returns “Other” correctly.

During development however, I saw that the code coverage for this component was quite low. I asked the team if it was okay to write unit tests to help expand coverage, after creating a couple of unit tests I managed to cover nearly every line of code in this component.

The screenshot shows a Jira issue page for TOFTM-6503. The left sidebar is visible with various project management links like TOF Sprint Board, User Story Map, Backlog, Active sprints, Releases, Reports, and Issues (which is selected). The main content area shows the issue details:

- Details:** Type: Story, Priority: Minor, Affects Version/s: None, Component/s: None, Resolution: Unresolved, Fix Version/s: None, Security Level: Confidential - Internal (for internal distribution only).
- General:** Original Key: TOFTM-6503, Epic Link: CNP Mobile leftovers, Story Points: 3, Sprint: TOF S24.22(23/10-5/11), Requirement Status: UNCOVERED.
- Description:** With this story we aim to handle NACK cases for reject code 199. Currently when Artilium sends reject code 199, in description they are sending open text, on the other hand CRDC is only expecting a description as "OTHER", if its not like that then we receive a NACK. To solve this whenever from Artilium, CNP receive code 199 then to CRDC it will send description as Other.

Below the description is a code editor showing Java code:

```
private void setRejectDetails(
    PortingRequest request, ArrayList<Npreject.RejectReason> rejectReasons, Reason reason) {
    Npreject.RejectReason rejectReason = new Npreject.RejectReason();
    rejectReason.setNumber(request.getPortingDetails().getMsisdn().get(0));
    rejectReason.setRejectcode(reason.getCode());
    rejectReason.setRejectstring(reason.getDescription());
    rejectReasons.add(rejectReason);
}
```

```
15      ▲ mheylen1 +1
16  @ExtendWith(MockitoExtension.class)
17  class NrejectMessageFactoryTest {
18
19      2 usages
20      @InjectMocks
21      NrejectMessageFactory nrejectMessageFactory;
22
23      ▲ mheylen1 +1
24      @Test
25      void testCreateMessageNC() throws DatatypeConfigurationException {
26          Reason mockReason1 = Reason.builder().description("NOT_ACTIVE").build();
27
28          Step mockStep = Step.builder() Step.StepBuilder<capture of ?, capture of ?>
29              .source(Systems.NC_BSS) capture of ?
30              .stepName("SOME_STEP")
31              .reason(Collections.singletonList(mockReason1))
32              .build();
33
34          PortingRequest mockPortingRequest = PortingRequest.builder() PortingRequest.PortingRequestBuilder<...>
35              .portingDetails(PortingDetails.builder()
36                  .nprNumber("123")
37                  .msisdns(Collections.singletonList("123456789"))
38                  .build()) capture of ?
39              .build();
40
41          Nreject nreject = (Nreject) nrejectMessageFactory.createMessage(mockStep, mockPortingRequest);
42
43          assertEquals(VERSION, nreject.getVersion());
44          assertEquals(expected: "123", nreject.getNprnumber());
45
46          Nreject.Rejectreason rejectreason = nreject.getRejectreason().get(0);
47
48          assertEquals(expected: "123456789", rejectreason.getNumber());
49          assertEquals(expected: "100", rejectreason.getRejectcode());
50          assertEquals(expected: "Not active", rejectreason.getRejectstring());
51
52      }
53
54      ▲ mheylen1 +1
```

```

± mheylen1 +1
@Test
void testCreateMessageArtillum() throws DatatypeConfigurationException {
    Reason mockReason1 =
        Reason.builder().code("100").description("Not Active").build();

    Reason mockReason2 = Reason.builder().code("199").description("OTHER").build();

    ArrayList<Reason> reasons = new ArrayList<>();
    reasons.add(mockReason1);
    reasons.add(mockReason2);

    Step mockStep = Step.builder() Step.StepBuilder<capture of ?, capture of ?>
        .source(Systems.ARTILIUM) capture of ?
        .stepName("SOME_STEP")
        .reason(reasons)
        .build();

    PortingRequest mockPortingRequest = PortingRequest.builder() PortingRequest.PortingRequestBuilder<...>
        .portingDetails(PortingDetails.builder()
            .nprNumber("123")
            .msisdns(Collections.singletonList("123456789"))
            .build()) capture of ?
        .build();

    Nreject npject = (Nreject) npjectMessageFactory.createMessage(mockStep, mockPortingRequest);

    assertEquals(VERSION, npject.getVersion());
    assertEquals(expected: "123", npject.getNprnumber());

    Nreject.Rejectreason rejectreason1 = npject.getRejectreason().get(0);
    Nreject.Rejectreason rejectreason2 = npject.getRejectreason().get(1);

    assertEquals(expected: "123456789", rejectreason1.getNumber());
    assertEquals(expected: "100", rejectreason1.getRejectcode());
    assertEquals(expected: "Not Active", rejectreason1.getRejectstring());

    assertEquals(expected: "123456789", rejectreason1.getNumber());
    assertEquals(expected: "199", rejectreason2.getRejectcode());
    assertEquals(expected: "Other", rejectreason2.getRejectstring());
}

```

Element ▲	Class, %	Method, %	Line, %
be.telenet.cnp.crdc.outbound.message.fac	20% (1/5)	33% (5/15)	36% (27/75)
● NpacceptMessageFactory	0% (0/1)	0% (0/2)	0% (0/6)
● NpdoneMessageFactory	0% (0/1)	0% (0/3)	0% (0/14)
● NpexecrejectMessageFactory	0% (0/1)	0% (0/3)	0% (0/17)
● NpreadyMessageFactory	0% (0/1)	0% (0/2)	0% (0/10)
● NprejectMessageFactory	100% (1/1)	100% (5/5)	96% (27/28)

Manual testing development story

Beside unit tests we also need to manually test the story to see if the code is functional. This is mostly done via Postman and the UI.

The image displays three separate screenshots of the Postman application interface, illustrating the manual testing process for different API stories:

- Screenshot 1: Initiate Number**
This screenshot shows a single POST request to `http://dinoman-mock.mnp.dev.numberportability-dev.awsnprd.internal.telenet.be/admin/initiate`. The request body is defined as follows:

```
1 {
2     "phoneNumber": "0462010901",
3     "currentProviderCode": "BASE",
4     "initialProviderCode": "MOBM",
5     "responseProviderDescription": "Response",
6     "bssSystem": "BASE LEGACY"
7 }
```

- Screenshot 2: sync-commands - nprequest - prepaid**
This screenshot shows a collection named "Sync-commands" containing several POST requests under the "Admin" category. One specific request is highlighted with its body defined as:

```
1 {
2     "serviceclass": "PREPAID",
3     "operatorid": "TEMO",
4     "number": "0462010901",
5     "simcardnumber": "893207001010673510",
6     "nprnnumber": "0462010901"
7 }
```

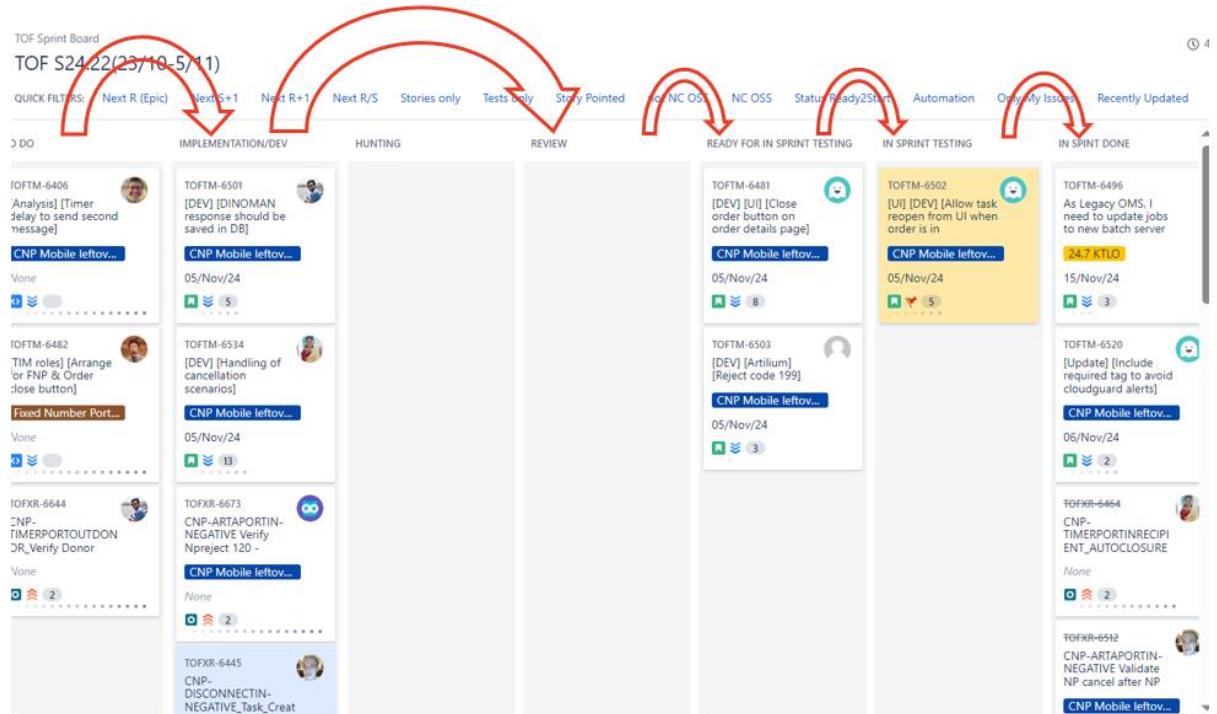
- Screenshot 3: NpRejectParameters**
This screenshot shows a collection named "WisailMock" containing various POST requests. One request is highlighted with its body defined as:

```
1 {
2     "nprnnumber": "0462010901",
3     "number": "0462010901",
4     "rejectcode": "199",
5     "rejectstring": "random"
6 }
```

Creation Date	Step Name	Message Type	Status	Source	Destination	Message
October 29, 2024 03:44:37.0436 PM	Order Closed	CNP_INTERNAL	OK	CNP	CNP	View
October 29, 2024 03:44:36.0498 PM	Reject Feedback Response From Artilium	NP_REJECT_SNT_FEEDBACK_RESPONSE	OK	CNP	CNP	View
October 29, 2024 03:44:36.0481 PM			OK	CNP	ARTILIUM	View
October 29, 2024 03:44:36.0029 PM			OK	CRDC_BASE	ARTILIUM	View
October 29, 2024 03:44:35.0993 PM			OK	ARTILIUM	CRDC_BASE	View
October 29, 2024 03:44:31.0128 PM			OK	ARTILIUM	CRDC_BASE	View
October 29, 2024 03:43:35.0926 PM			OK	CNP	CNP	View
October 29, 2024 03:43:35.0911 PM			OK	CNP	ARTILIUM	View
October 29, 2024 03:43:35.0248 PM	Duplicate Checked	CNP_INTERNAL	OK	CNP	CNP	View
October 29, 2024 03:43:32.0010 PM	Request Enriched	CNP_INTERNAL	OK	CNP	CNP	View
October 29, 2024 03:43:27.0281 PM	Msisdn Request Mapping	CNP_INTERNAL	OK	CNP	CNP	View
October 29, 2024 03:43:24.0232 PM	Request Received From Crdc	NP_REQUEST	OK	CRDC_BASE	CNP	View
1 row selected						
Rows per page: 25 ▾ 1–12 of 12 < >						

Sprint board

The following image display the workflow for development stories. The stories of the current sprint will be displayed in the TO DO column. When you start a story, you put it into IMPLEMENTATION/DEV. After your story is complete and manually tested with an open Merge Request, the story will need to be reviewed by another team member. After the review is successful, the story will need to be tested in sprint. This means one of the tester members in the team will take up your story for integration testing. After this is finally complete and successful, the story will be considered done.



Pipelines

Our code needs to go through the automated workflow where it is used to build, test and deploy the code. The dev environment is used for depoying and testing code changes in a non-production environment. E2ETest stands for End-to-End testing where it runs the automated tests to verify the entire application workflow, ensuring all components work together as expected. Finally we come to the E2ESTAB stage or End-to-End Stable where final validation happens before actual deployment to production.

The screenshot shows a GitLab pipeline interface with the following details:

- Project:** TIF / Number Management / testing / e2e-flow / Pipelines
- Branch name:** matthias/TOFXR
- Status:** Blocked (for three pipelines)
- Pipeline Details:**
 - #101673: Update version to based on Merge branch 'dev/TOFTM-6503' into 'develop'. Status: Passed (00:02:54, 20 hours ago). Stages: DEV-validate: passed, DEV-apply: passed, E2TEST-validate: passed, E2ESTAB-deploy: passed.
 - #1016569: Update version to based on Merge branch 'dev/TOFTM-6503' into 'develop'. Status: Passed (00:22:44, 20 hours ago). Stages: DEV-validate: passed, DEV-apply: passed, E2TEST-validate: passed, E2ESTAB-deploy: passed.
 - #1016569: Update version to based on Merge branch 'dev/TOFTM-6503' into 'develop'. Status: Passed (00:22:44, 20 hours ago). Stages: DEV-validate: passed, DEV-apply: passed, E2TEST-validate: passed, E2ESTAB-deploy: passed.
- Actions:** Clear runner caches, Run pipeline

[DEV] [Achieve SonarQube coverage for different modules]

After my first development story, the developers noted there are other components that do not reach the expected code coverage, which is 80% minimum. This was a longer story where I had to not only expand code coverage for many components, but also remove the bugs and code smells. This meant writing a lot of unit tests for these components and refactoring some parts of the code to increase the quality of the code.

The team uses SonarQube to check the code coverage, code smells and bugs of the components.

Most of the following code will be from GitLab to help clear confusion about which parts I worked on and which parts of the code are written by someone else entirely.

The screenshot shows a GitLab issue page for a story titled "[DEV] [Achieve SonarQube coverage for different modules]". The story is assigned to Heylen Matthias and Tiwari Vikas. It was created on 04/Nov/24 at 9:46 AM, updated on 20/Nov/24 at 10:39 AM, and resolved on 20/Nov/24 at 10:39 AM. The sprint is TOF S24.23(06/11-19/11). The story has a resolution of "Done", a fix version of "None", and a security level of "Confidential - Internal (for internal distribution only)". The description section details the goal of achieving 80% test coverage for various components, listing numerous code smells and bugs found in components like cnp-service, task-request-handler, etc. The test coverage section indicates no tests were found. The activity tab shows the story was pinned to highlight important information. The page includes sections for attachments, comments, work log, history, activity, transitions, source, and smart checklist history.

Cnp-service (coverage)

sonarqube Projects Portfolios Issues Rules Quality Profiles Quality Gates M

Last analysis of this Branch had 1 warning December 3, 2024 at 2:48 PM Version 24.18.1102628

Overview Issues Security Hotspots Security Reports Measures Code Activity Project Information Project PDF report

To benefit from more of SonarQube's features, ask a project administrator to set up DevOps platform integration.

QUALITY GATE STATUS

Passed
All conditions passed.

You added extra conditions to the "Clean as You Code" quality gate, which is something we do not recommend. Review your [quality gate](#).

[Learn why](#)

MEASURES

New Code	Overall Code
Since 24.18.1094629 Started 7 days ago	
0 Bugs	Reliability A
0 Vulnerabilities	Security A
0 Security Hotspots	Reviewed
10 Code Smells	Security Review A
3h 5min Debt	Maintainability A
68.8% Coverage on 2K Lines to cover	Unit Tests 148
2.3% Duplications on 3.8k Lines	Duplicated Blocks 5

ACTIVITY

Choose graph type: Issues

Bugs: 0, Code Smells: 10, Vulnerabilities: 0

new code

December 3, 2024 at 2:48 PM
24.18.1102628
November 29, 2024 at 9:47 AM
24.18.1094629
November 18, 2024 at 10:13 AM
24.18.102146
Quality Gate: Passed
November 14, 2024 at 3:26 PM
24.18.1055670
Quality Gate: Failed
November 14, 2024 at 1:54 PM
24.18.1055237

Activity

sonarqube Projects Portfolios Issues Rules Quality Profiles Quality Gates M

Last analysis of this Branch had 1 warning November 19, 2024 at 3:38 PM Version 24.18.1067701

Overview Issues Security Hotspots Security Reports Measures Code Activity Project Information Project PDF report

To benefit from more of SonarQube's features, ask a project administrator to set up DevOps platform integration.

QUALITY GATE STATUS

Passed
All conditions passed.

You added extra conditions to the "Clean as You Code" quality gate, which is something we do not recommend. Review your [quality gate](#).

[Learn why](#)

MEASURES

New Code	Overall Code
Since 24.18.1067504 Started 17 days ago	
0 Bugs	Reliability A
0 Vulnerabilities	Security A
0 Security Hotspots	Reviewed
0 Debt	Code Smells 0
85.7% Coverage on 1.7k Lines to cover	Unit Tests 175
0.0% Duplications on 3.4k Lines	Duplicated Blocks 0

ACTIVITY

Choose graph type: Issues

Bugs: 0, Code Smells: 8, Vulnerabilities: 0

new code

November 19, 2024 at 3:38 PM
24.18.1067701
November 19, 2024 at 3:21 PM
24.18.1067504
November 19, 2024 at 3:01 PM
24.18.1067341
November 19, 2024 at 2:43 PM
24.18.1067252
November 19, 2024 at 12:38 PM
24.18.1066884

Activity

Resolve TOFTM-6606 "Dev/"

Open Heylen Matthias requested to merge dev/TOFTM-6606 into develop 2 weeks ago

Overview 0 Commits 11 Pipelines 1 Changes 31 Add a to do

Compare develop and latest version

Files 31 31 files +1018 -388 Viewed

Search (e.g. *.vue) (Ctrl+P)

src

- main/java/be...enet/cnpservice
 - exception
 - TaskQu...tion.java +8 -0
 - TaskSe...tion.java +0 -3
 - handler
 - MetricH...lder.java +5 -0
 - mapper
 - Portin...oper.java +6 -111
 - repository
 - Paginat...tory.java +4 -3

src/main/java/be/telenet/cnpservice/exception/TaskQueryException.java 0 → +8 -0 Viewed

```
1 + package be.telenet.cnpservice.exception;
2 +
3 + public class TaskQueryException extends RuntimeException {
4 +
5 +     public TaskQueryException(String message, Throwable cause) {
6 +         super(message, cause);
7 +     }
8 + }
```

src/main/java/be/telenet/cnpservice/exception/TaskSearchException.java +0 -3 Viewed

```
1 1 package be.telenet.cnpservice.exception;
2 2
3 3 public class TaskSearchException extends RuntimeException {
4 -
5 -     public TaskSearchException(String message) {
6 -         super(message);
7 -     }
8 - }
```

src/main/java/be/telenet/cnpservice/handler/MetricHandler.java +5 -0 Viewed

```
↑ @@ -21,6 +21,11 @@ public class MetricHandler {
21   21     this.cloudWatchClient = CloudWatchClient.create();
22   22 }
23   23
24 + public MetricHandler(CloudWatchClient cloudWatchClient, String namespace) {
25 +     this.cloudWatchClient = cloudWatchClient;
26 +     this.namespace = namespace;
27 + }
28 +
29 public void publishMetric(String metricName, String dimension, String dimensionValue)
30     throws SOAPException, IOException, ParserConfigurationException, SAXException {
31     if (null == metricName) {
```

src/main/java/be/telenet/cnpservice/mapper/PortingMessageDTOMapper.java +6 -111 Viewed

```

    @@ -4,6 +4,8 @@ import static be.telenet.cnp.data.model.Operator.BASE_RETAIL;
4   4     import static be.telenet.cnp.data.model.Operator.BASE_WHOLESALE;
5   5     import static be.telenet.cnp.data.model.Operator.TELENET;
6   6     import static be.telenet.cnp.data.model.Systems.*;
7   7 + import static be.telenet.cnpservice.util.PortingMapperUtil.*;
8   8 + import static be.telenet.cnpservice.util.TimeProvider.currentTime;
9   9
8 10     import be.telenet.cnp.data.model.BffPortingRequestOrder;
9 11     import be.telenet.cnp.data.model.Customer;
12
13     @@ -37,7 +39,6 @@ import be.telenet.cnpservice.repository.model.TaskBean;
37 39     import com.fasterxml.jackson.core.JsonProcessingException;
38 40     import com.fasterxml.jackson.core.type.TypeReference;
39 41     import com.fasterxml.jackson.databind.ObjectMapper;
40
41 42     - import java.time.ZoneId;
42 43     import java.time.ZonedDateTime;
43 44     import java.util.ArrayList;
44     import java.util.HashMap;
14
59 60     @@ -59,13 +60,6 @@ import software.amazon.awssdk.services.dynamodb.model.AttributeValue;
60 61     @RequiredArgsConstructor
61 62     public class PortingMessageDTOMapper {
62
63     -     private static final String STEP = "STEP#";
64     -     private static final String ZONE = "Europe/Brussels";
64     -     private static final String UNKNOWN = "UNKNOWN";
65
65     -     private static final String ORDER_CLOSED = "ORDER_CLOSED";
66     -     private static final String TYPE_STATUS_PATTERN = "%s#%s";
67     -     private static final String RESUBMIT = "RESUBMIT";
68
69 63     public BffPortingRequestOrderBean buildRequest(BffPortingRequestOrder bffPortingRequestOrder,
70 64     Operator operator) {
71 64         return BffPortingRequestOrderBean.builder()
71 65             .npRequestId(bffPortingRequestOrder.getNpRequestId())
15
84 78     @@ -84,33 +78,14 @@ public class PortingMessageDTOMapper {
85 79         }
86 80
87     -     private String orderTypePortingStatus(BffPortingRequestOrder bffPortingRequestOrder) {
88     -         return TYPE_STATUS_PATTERN.formatted(
89     -             bffPortingRequestOrder.getOrderType(), bffPortingRequestOrder.getPortingStatus());
90     -     }
91
92     -     public static ZonedDateTime currentTime() {
93     -         return ZonedDateTime.now().withZoneSameInstant(ZoneId.of(ZONE));
94     -     }
95
96 81     public List<Operator> findOperator(BffPortingRequestOrder bffPortingRequestOrder) {
97 82         List<Operator> operators = new ArrayList<>();
98 83
99 84         switch (bffPortingRequestOrder.getOrderType()) {
100    -         case PORT_IN:

```

```

101      -             handlePortIn(bffPortingRequestOrder, operators);
102      -             break;
103      -         case PORT_OUT:
104      -             handlePortOut(bffPortingRequestOrder, operators);
105      -             break;
106      -         case DISCONNECT_IN, PORT_ACROSS:
107      -             handleDisconnectIn(bffPortingRequestOrder, operators);
108      -             break;
109      -         case DISCONNECT_OUT:
110      -             handleDisconnectOut(bffPortingRequestOrder, operators);
111      -             break;
112      -         default:
113      -             break;
85      +     case PORT_IN -> handlePortIn(bffPortingRequestOrder, operators);
86      +     case PORT_OUT -> handlePortOut(bffPortingRequestOrder, operators);
87      +     case DISCONNECT_IN, PORT_ACROSS -> handleDisconnectIn(bffPortingRequestOrder,
88      +                                         operators);
89      +     case DISCONNECT_OUT -> handleDisconnectOut(bffPortingRequestOrder, operators);
114      89      }
115      90      return operators;
116      91  }

↓
↑ @@ -251,14 +226,6 @@ public class PortingMessageDTOMapper {
251  226      return existingPortInInfo;
252  227  }
253  228

254      -     private String modificationUser(PortingRequest request) {
255      -         return null != request.getModificationUser() ? request.getModificationUser() :
256      -             Systems.CNP.name();
257      -
258      -     private Object updateCheck(Object newValue, Object oldValue) {
259      -         return null != newValue ? newValue : oldValue;
260      -     }
261      -
262  229      -     private void mapPortingDetailsForUpdate(PortingRequest portingRequest, PortingRequestBean
263  230      existingPortInInfo) {
264  231      PortingDetails portingDetails = portingRequest.getPortingDetails();
265      if (null != portingDetails) {
266      }

↓
↑ @@ -368,10 +335,6 @@ public class PortingMessageDTOMapper {
368  335      }
369  336      }
370  337      }

371      -     private String nonEmptyString(String value) {
372      -         return StringUtils.hasLength(value) ? value : null;
373      -     }
374      -

375  338      public PortingRequestBean buildPortInStepFromPortingRequest(
376  339          PortingRequest portingRequest, String generatedPortingRequestId) {
377  340          String npRequestIdFromRequest = portingRequest.getNpRequestId();

↓

```

```

↑                                     @@ -501,10 +504,6 @@ public class PortingMessageDTOMapper {
401 364         return builder.build();
402 365     }
403 366 }
404 -     private String creationUser(Step step) {
405 -         return null != step.getCreationUser() ? step.getCreationUser() : Systems.CNP.name();
406 -     }
407 -
408 367     private void setStepReason(PortingRequestBean.PortingRequestBeanBuilder builder, Step step) {
409 368         if (null != step.getReason() && !step.getReason().isEmpty()) {
410 369             try {
↓                                     @@ -572,27 +531,6 @@ public class PortingMessageDTOMapper {
572 531                 .build();
573 532             }
574 533 }
575 -     private String buildPartitionKey(OrderType orderType, MessageType messageType, String
576 -                                     portingRequestId) {
577 -         return orderType
578 -             .name()
579 -             .concat("#")
580 -             .concat(messageType.getMessage())
581 -             .concat("#")
582 -             .concat(portingRequestId);
583 -     }
584 -     private String buildSortKeyForStep(String stepName, String stepId, String npRequestId) {
585 -         return stepName.equals(ORDER_CLOSED) ? stepSortKey(stepName, npRequestId) :
586 -             stepSortKey(stepName, stepId);
587 -     }
588 -     private String stepSortKey(String stepName, String id) {
589 -         return STEP.concat(stepName).concat("#").concat(id);
590 -     }
591 -
592 -     private String buildSortKeyForInfo(String portingRequestId) {
593 -         return "INFO#".concat(portingRequestId);
594 -     }
595 -
596 534     public BffPortingRequestOrder buildOrderFromDynamodbEntity(BffPortingRequestOrderBean entity)
597 535     {
598 536         return BffPortingRequestOrder.builder()
↓                                     @@ -647,16 +585,6 @@ public class PortingMessageDTOMapper {
647 585             .npRequestId(entity.getNpRequestId())
648 586         }
649 587 }
650 -     private String customerNumber(Task task) {
651 -         return StringUtils.hasLength(task.getCustomerNumber()) ? task.getCustomerNumber() : null;

```

```

652      }
653
654      private String taskTypeStatusScenario(Task task) {
655          return null != task.getTaskTypeStatusScenario()
656              ? task.getTaskTypeStatusScenario()
657              : TYPE_STATUS_PATTERN.formatted(task.getType(), task.getStatus());
658      }
659
660      private void setReason(Task task, TaskBean.TaskBeanBuilder builder) throws
661          JsonProcessingException {
662          if (null != task.getReason() && !task.getReason().isEmpty()) {
663              var reasons = task.getReason().stream()
664                  @@ -672,11 +600,6 @@ public class PortingMessageDTOMapper {
665                      }
666
667      }
668
669      private String generateTaskPartitionKey(TaskType taskType, OrderType orderType, String
670          taskId) {
671          String orderTypeValue = null != orderType ? orderType.name() : UNKNOWN;
672          return taskType.name().concat("#").concat(orderTypeValue).concat("#").concat(taskId);
673      }
674
675
676
677
678
679
680      public Task buildTaskFromTaskBean(TaskBean taskBean) {
681          return Task.builder()
682              .id(taskBean.getTaskId())
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2
```

793	700	
834	741	@@ -834,18 +741,6 @@ public class PortingMessageDTOMapper {
835	742	. build();
836	743	}
837	-	private static String getStepId(String sortKey) {
838	-	String input = sortKey;
839	-	-
840	-	String[] parts = input.split("#");
841	-	if (parts.length > 2) {
842	-	return parts[2];
843	-	} else {
844	-	log.info("UUID not found.");
845	-	}
846	-	return null;
847	-	}
848	-	-
849	744	public MsgOutToCrdrBean buildMsgOutToCrdrBean(MsgOutToCrdr msgOutToCrdr) {
850	745	ZonedDateTime currentTime = currentTime();
851	746	String partitionKey = UUID.randomUUID().toString();

src/main/java/be/telenet/cnpservice/repository/PaginatedTaskRepository.java		
1	1	package be.telenet.cnpservice.repository;
2	2	
3	+ 3	import be.telenet.cnpservice.exception.TaskQueryException;
4	4	import be.telenet.cnpservice.repository.model.MsisdnTaskMappingBean;
5	5	import be.telenet.cnpservice.repository.model.TaskBean;
6	6	import io.awspring.cloud.dynamodb.DynamoDbOperations;
27	28	@@ -27,7 +28,7 @@ public class PaginatedTaskRepository {
28	29	return result;
29	30	} catch (Exception e) {
		log.info("Error executing paginated query for tasks with index: {}. Error: {}",
		index, e.getMessage(), e);
30	-	throw new RuntimeException("Failed to execute paginated query for tasks", e);
31	+ 31	throw new TaskQueryException("Failed to execute paginated query for tasks", e);
32	32	}
33	33	}
41	42	@@ -41,7 +42,7 @@ public class PaginatedTaskRepository {
42	43	return result;
43	44	} catch (Exception e) {
		log.info("Error executing paginated scan for tasks. Error: {}", e.getMessage(), e);
44	-	throw new RuntimeException("Failed to execute paginated scan for tasks", e);
45	+ 45	throw new TaskQueryException("Failed to execute paginated scan for tasks", e);
46	46	}
47	47	}
56	57	@@ -56,7 +57,7 @@ public class PaginatedTaskRepository {
57	58	return result;
58	59	} catch (Exception e) {
		log.info("Error executing paginated query for MSISDN-Task ID mapping. Error: {}",
		e.getMessage(), e);
59	-	throw new RuntimeException("Failed to execute paginated query for MSISDN-Task ID mapping", e);
60	+ 60	throw new TaskQueryException("Failed to execute paginated query for MSISDN-Task ID mapping", e);
61	61	}
62	62	}

src/main/java/be/telenet/cnpservice/repository/TaskSearchRepository.java

```

@@ -138,12 +138,10 @@ public class TaskSearchRepository {
    tasks.addAll(pageTasks);
} else {
    tasks.addAll(pageTasks.subList(0, remainingSpace));
}
hasMoreResults = true;
}
if (page.lastEvaluatedKey() != null && !page.lastEvaluatedKey().isEmpty()) {
    lastEvaluatedKey = LastEvaluatedKeySerializer.serialize(page.lastEvaluatedKey());
}
hasMoreResults = true;
}
totalCount += page.items().size();

```

src/main/java/be/telenet/cnpservice/service/MsgOutToCrdcService.java

```

@@ -2,9 +2,9 @@ package be.telenet.cnpservice.service;
import static be.telenet.cnp.data.model.MessageType.NP_EXEC;
import static be.telenet.cnp.data.model.ServiceClass.POST_PAID_COMPLEX;
-import static be.telenet.cnpservice.mapper.PortingMessageDTOMapper.currentTime;
import static be.telenet.cnpservice.util.CnpServiceUtil.DOWN;
import static be.telenet.cnpservice.util.CnpServiceUtil.ON_HOLD;
+ import static be.telenet.cnpservice.util.TimeProvider.currentTime;
import be.telenet.cnp.data.model.MsgOutToCrdc;
import be.telenet.cnp.data.model.PortingRequest;
@@ -119,13 +119,11 @@ public class MsgOutToCrdcService {
private List<String> getOperatorsWithStatus(String status) {
    List<OperatorStatus> operatorStatusList = operatorService.getDownOperatorStatus();
}
List<String> operatorDownList = new ArrayList<>();
var downOperators = operatorStatusList.stream()
    .filter(operatorStatus -> status.equals(operatorStatus.getStatus()))
    .map(OperatorStatus::getOperator)
    .toList();
operatorDownList.addAll(downOperators);
return operatorDownList;
}
+ return new ArrayList<>(downOperators);
}

```

src/main/java/be/telenet/cnpservice/service/NprNumberGeneratorService.java

```

package be.telenet.cnpservice.service;
-import static be.telenet.cnpservice.mapper.PortingMessageDTOMapper.currentTime;
+ import static be.telenet.cnpservice.util.TimeProvider.currentTime;
import be.telenet.cnp.data.model.OperatorCode;
import be.telenet.cnpservice.repository.NprCounterRepository;
@@ -34,7 +34,7 @@ public class NprNumberGeneratorService {
    log.info("NPR number key generated: {}", key);
    var counterValue = getCounter(key);
    log.info("NPR number count generated: {}", counterValue);
}
if (Integer.valueOf(counterValue) >= 100) {
}
if (Integer.parseInt(counterValue) >= 100) {
    log.error("NPR number counter exceeded the limit");
    Thread.sleep(1000);
    key = getNprKey(recipientCode);
}

```

```

src/main/java/be/telenet/cnpservice/service/OperatorService.java
+1 -1 Viewed

1 1 package be.telenet.cnpservice.service;
2 2
3 3 import static be.telenet.cnp.data.model.Systems.CNP;
4 4 - import static be.telenet.cnpservice.mapper.PortingMessageDTOMapper.currentTime;
5 5 import static be.telenet.cnpservice.util.CnpServiceUtil.DOWN;
6 6 import static be.telenet.cnpservice.util.CnpServiceUtil.ON_HOLD;
7 7 + import static be.telenet.cnpservice.util.TimeProvider.currentTime;
8 8 import be.telenet.cnpservice.mapper.PortingMessageDTOMapper;
9 9 import be.telenet.cnpservice.repository.OperatorRepository;

↓

```



```

src/main/java/be/telenet/cnpservice/service/OrderService.java
+9 -67 Viewed

1 1 package be.telenet.cnpservice.service;
2 2
3 3 + import static be.telenet.cnpservice.util.FilterUtil.*;
4 4 +
5 5 import be.telenet.cnp.data.model.BffPortingRequestOrder;
6 6 import be.telenet.cnp.data.model.Donor;
7 7 import be.telenet.cnp.data.model.Operator;

↓
↑ @@ -42,7 +44,6 @@ public class OrderService {
42 44     private final BffPortingRequestOrderRepository repository;
43 45     private final MsisdnMappingRepository msisdnMappingRepository;
44 46     private final PortingMessageDTOMapper converter;
45 47 - private static final String TASK_ID_GSI = "task-id-gsi";
46 48     private static final String TYPE_CREATION_DATE_GSI = "type-creation-date-gsi";
47 49     private final RequestService requestService;

↓
↑ @@ -116,8 +117,7 @@ public class OrderService {
116 117         .withZoneSameInstant(ZoneId.of("Europe/Brussels"))
117 118         .toString();
118 119         return partitionValues(orderType, orderStatus).stream()
119 120             .map(partitionValue ->
-                 getOrdersWithToDate(orderSearchCriteria, partitionValue, date,
TYPE_CREATION_DATE_GSI))

```

```

120 +             .map(partitionValue -> getOrdersWithToDate(orderSearchCriteria,
partitionValue, date))
121 121             .flatMap(List::stream)
122 122             .toList();
123 123         } else {
↓
↑
193 193     @@@ -193,8 +193,7 @@ public class OrderService {
194 194         private List<BffPortingRequestOrder> streamOrdersWithFromDate(
195 195             List<String> partitionValues, OrderSearchCriteria orderSearchCriteria, String
fromDate) {
196             return partitionValues.stream()
197             -                 .map(partitionValue ->
198                 -                     getOrdersWithFromDate(orderSearchCriteria, partitionValue, fromDate,
199                 TYPE_CREATION_DATE_GSI))
200                 +                         .map(partitionValue -> getOrdersWithFromDate(orderSearchCriteria, partitionValue,
fromDate))
201                 .flatMap(List::stream)
202                 .toList();
203             }
↓
↑
204 204     @@@ -202,8 +201,7 @@ public class OrderService {
205 205         private List<BffPortingRequestOrder> streamOrdersWithToDate(
206 206             List<String> partitionValues, OrderSearchCriteria orderSearchCriteria, String toDate)
207             {
208                 return partitionValues.stream()
209                 -                     .map(partitionValue ->
210                     -                         getOrdersWithToDate(orderSearchCriteria, partitionValue, toDate,
211                     TYPE_CREATION_DATE_GSI))
212                     .map(partitionValue -> getOrdersWithToDate(orderSearchCriteria, partitionValue,
toDate))
213                     .flatMap(List::stream)
214                     .toList();
215             }
↓
↑
263 261     @@@ -263,7 +261,7 @@ public class OrderService {
264 262         }
265 263     private List<BffPortingRequestOrder> getOrdersWithFromDate(
266             -                 OrderSearchCriteria orderSearchCriteria, String partitionValue, String fromDate,
267             String index) {
268                 +                     OrderSearchCriteria orderSearchCriteria, String partitionValue, String fromDate) {
269                     Expression filterExpression = filterExpression(orderSearchCriteria, false, true);
270                     QueryConditional queryConditional = QueryConditional.sortGreaterThanOrEqualTo(
271                         Key.builder().partitionValue(partitionValue).sortValue(fromDate).build());
↓
↑
275 273     @@@ -275,13 +273,13 @@ public class OrderService {
276 274         : QueryEnhancedRequest.builder()
277 275             .queryConditional(queryConditional)
278             .build();
279             -             return repository.getOrders(request, index).stream()
280             +             return repository.getOrders(request, OrderService.TYPE_CREATION_DATE_GSI).stream()
281                 .map(converter::buildOrderFromDynamodbEntity)
282                 .toList();
283             }
284     private List<BffPortingRequestOrder> getOrdersWithToDate(

```

```

284      -     OrderSearchCriteria orderSearchCriteria, String partitionValue, String toDate, String
285      282      index) {
286      283      +     OrderSearchCriteria orderSearchCriteria, String partitionValue, String toDate) {
287      284          Expression filterExpression = filterExpression(orderSearchCriteria, false, true);
288      285          QueryConditional queryConditional = QueryConditional.sortLessThanOrEqualTo(
289              Key.builder().partitionValue(partitionValue).sortValue(toDate).build());
290      291      @@ -293,7 +291,7 @@ public class OrderService {
291          : QueryEnhancedRequest.builder()
292              .queryConditional(queryConditional)
293              .build();
294      296      -     return repository.getOrders(request, index).stream()
295      297      +     return repository.getOrders(request, OrderService.TYPE_CREATION_DATE_GSI).stream()
296          .map(converter::buildOrderFromDynamodbEntity)
297          .toList();
297      }
298
299
300      @@ -333,60 +331,4 @@ public class OrderService {
301          .expressionValues(expressionValues)
302          .build();
303      }
304
305      -     private Expression filterForOperator(
306          -         OrderSearchCriteria orderSearchCriteria, String expression, Map<String,
307              AttributeValue> expressionValues) {
308          -             if (!expression.isEmpty()) {
309          -                 expression = expression.concat(" AND ");
310
311          -             }
312          -             expression = expression.concat("#op = :operator");
313          -             expressionValues.put(
314          -                 ":operator",
315          -                 AttributeValue.builder()
316          -                     .s(orderSearchCriteria.getOperator().name())
317          -                     .build());
318          -             return Expression.builder()
319          -                 .expression(expression)
320          -                 .expressionValues(expressionValues)
321          -                 .expressionNames(Map.of("#op", "operator"))
322          -                 .build();
323      }
324
325
326      -     private String filterExpressionForDate(
327          -         String fromDate, String toDate, Map<String, AttributeValue> expressionValues) {
328          -             if (fromDate != null && toDate != null) {
329          -                 expressionValues.put(
330          -                     ":fromDate", AttributeValue.builder().s(fromDate).build());
331          -                 expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
332          -                 return "creationDate BETWEEN :fromDate AND :toDate";
333          -             } else if (fromDate != null) {
334          -                 expressionValues.put(
335          -                     ":fromDate", AttributeValue.builder().s(fromDate).build());
336          -                 return "creationDate >= :fromDate";
337          -             } else if (toDate != null) {
338          -                 expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
339          -                 return "creationDate <= :toDate";
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368

```

```

369         }
370         return "";
371     }
372
373     private QueryEnhancedRequest requestWithFilterExpression(
374         String partitionValue, String sortValue, Expression filterExpression) {
375         return QueryEnhancedRequest.builder()
376             .queryConditional(QueryConditional.keyEqualTo(Key.builder()
377                 .partitionValue(partitionValue)
378                 .sortValue(sortValue)
379                 .build()))
380             .filterExpression(filterExpression)
381             .build();
382     }
383
384     private QueryEnhancedRequest requestWithoutFilterExpression(String partitionValue, String
385         sortValue) {
386         return QueryEnhancedRequest.builder()
387             .queryConditional(QueryConditional.keyEqualTo(Key.builder()
388                 .partitionValue(partitionValue)
389                 .sortValue(sortValue)
390                 .build()))
391             .build();
392     }
393     334 }
```

src/main/java/be/telenet/cnpservice/service/OrderServiceV2.java

+120 -53 Viewed

↑	@@ -162,64 +162,95 @@ public class OrderServiceV2 {
162	}
163	163
164	164 private OrderSearchResponseV2 getOrdersByTypeAndStatus(OrderSearchCriteriaV2 criteria) {
165	+ logSearchStart(criteria);
166	+
167	+ List<String> partitionValues = partitionValues(criteria.getOrderType(),
	criteria.getOrderStatus());
168	+ Map<String, AttributeValue> lastEvaluatedKey = getLastEvaluatedKey(criteria);
169	+ String currentPartition = determineCurrentPartition(partitionValues, lastEvaluatedKey);
170	+ int startIndex = getstartIndex(partitionValues, currentPartition);
171	+
172	+ List<BffPortingRequestOrder> allOrders = new ArrayList<>();
173	+ Map<String, AttributeValue> finalLastEvaluatedKey = null;
174	+ boolean hasMoreResults = false;
175	+
176	+ for (int i = startIndex; i < partitionValues.size(); i++) {
177	+ QueryResult queryResult = processPartition(
178	+ partitionValues.get(i), criteria, (i == startIndex) ? lastEvaluatedKey :
	null, allOrders);
179	+
180	+ allOrders.addAll(queryResult.orders);
181	+ finalLastEvaluatedKey = queryResult.lastEvaluatedKey;
182	+
183	+ if (allOrders.size() >= criteria.getPageSize()) {
184	+ hasMoreResults = finalLastEvaluatedKey != null;
185	+ break;

```

186     }
187     }
188     +
189     +     return buildResponse(allOrders, criteria.getPageSize(), hasMoreResults,
190     +         finalLastEvaluatedKey);
191     +
192     +     private void logSearchStart(OrderSearchCriteriaV2 criteria) {
193         log.info(
194             "Searching for orders with status: {} and type: {}",
195             criteria.getOrderStatus(),
196             criteria.getOrderType());
197         -     List<String> partitionValues = partitionValues(criteria.getOrderType(),
198             criteria.getOrderStatus());
199         -     log.info("getOrdersByTypeAndStatus --> Partition values : {}", partitionValues);
200         +
201         -     List<BffPortingRequestOrder> allOrders = new ArrayList<>();
202         -     Map<String, AttributeValue> lastEvaluatedKey = criteria.getLastEvaluatedKey() != null
203         +     private Map<String, AttributeValue> getLastEvaluatedKey(OrderSearchCriteriaV2 criteria) {
204         +         return criteria.getLastEvaluatedKey() != null
205             ? LastEvaluatedKeySerializer.deserialize(criteria.getLastEvaluatedKey())
206             : null;
207         +
208         -     String currentPartition = lastEvaluatedKey != null
209         +     private String determineCurrentPartition(
210             List<String> partitionValues, Map<String, AttributeValue> lastEvaluatedKey) {
211             return lastEvaluatedKey != null
212                 ? lastEvaluatedKey.get(ORDER_TYPE_PORTING_STATUS).s()
213                 : partitionValues.getFirst();
214         +
215         -     private int getstartIndex(List<String> partitionValues, String currentPartition) {
216             int startIndex = partitionValues.indexOf(currentPartition);
217             if (startIndex == -1) startIndex = 0;
218             -
219             Map<String, AttributeValue> finalLastEvaluatedKey = null;
220             boolean hasMoreResults = false;
221             -
222             for (int i = startIndex; i < partitionValues.size(); i++) {
223                 String partitionValue = partitionValues.get(i);
224                 Map<String, AttributeValue> partitionLastEvaluatedKey = (i == startIndex) ?
225                     lastEvaluatedKey : null;
226                 if (partitionValue.equals(partitionLastEvaluatedKey.s()))
227                     return (startIndex == -1) ? 0 : startIndex;
228             }
229         +
230             while (true) {
231                 QueryEnhancedRequest request = buildQueryRequest(
232                     criteria, partitionValue, partitionLastEvaluatedKey,
233                     criteria.getPageSize() - allOrders.size());
234                 log.info("Querying DynamoDB with request: {}", request);
235             }
236         +
237         -     private QueryResult processPartition(
238             String partitionValue,
239             OrderSearchCriteriaV2 criteria,

```

```

220 +         Map<String, AttributeValue> partitionLastEvaluatedKey,
221 +         List<BffPortingRequestOrder> allOrders) {
222 +     Map<String, AttributeValue> lastEvaluatedKey = partitionLastEvaluatedKey;
223 +     List<BffPortingRequestOrder> partitionOrders = new ArrayList<>();
195 224
196 -         Page<BffPortingRequestOrderBean> page = repository.getOrdersPage(request,
197 -             TYPE_CREATION_DATE_GSI);
197 -         log.info("Raw results from DynamoDB: {}", page.items());
225 +     do {
226 +         QueryEnhancedRequest request = buildQueryRequest(
227 +             criteria, partitionValue, lastEvaluatedKey, criteria.getPageSize() -
228 +             allOrders.size());
228 +         log.info("Querying DynamoDB with request: {}", request);
198 229
199 -         List<BffPortingRequestOrder> orders = page.items().stream()
200 -             .map(converter::buildOrderFromDynamodbEntity)
201 -             .filter(order -> matchesCriteria(order, criteria))
202 -             .toList();
230 +     Page<BffPortingRequestOrderBean> page = repository.getOrdersPage(request,
231 -         TYPE_CREATION_DATE_GSI);
231 +     log.info("Raw results from DynamoDB: {}", page.items());
203 232
204 -         log.info("Filtered orders for partition {}: {}", partitionValue, orders);
233 +     List<BffPortingRequestOrder> orders = page.items().stream()
234 -         .map(converter::buildOrderFromDynamodbEntity)
235 -         .filter(order -> matchesCriteria(order, criteria))
236 -         .toList();
205 237
206 -         allOrders.addAll(orders);
207 -         partitionLastEvaluatedKey = page.lastEvaluatedKey();
208 -         finalLastEvaluatedKey = partitionLastEvaluatedKey;
238 +     log.info("Filtered orders for partition {}: {}", partitionValue, orders);
239 +     partitionOrders.addAll(orders);
240 +     lastEvaluatedKey = page.lastEvaluatedKey();
209 241
210 -         if (allOrders.size() >= criteria.getPageSize() || partitionLastEvaluatedKey ==
211 -             null) {
211 -             break;
212 -         }
213 -     }
242 +     } while (partitionOrders.size() < criteria.getPageSize() && lastEvaluatedKey != null);
214 243
215 -         if (allOrders.size() >= criteria.getPageSize()) {
216 -             hasMoreResults = finalLastEvaluatedKey != null;
217 -             break;
218 -         }
219 -     }
244 +     return new QueryResult(partitionOrders, lastEvaluatedKey);
245 +   }
220 246
221 -         if (allOrders.size() > criteria.getPageSize()) {
222 -             allOrders = allOrders.subList(0, criteria.getPageSize());
247 +     private OrderSearchResponseV2 buildResponse(
248 +         List<BffPortingRequestOrder> allOrders,
249 +         int pageSize,
250 -             boolean hasMoreResults

```

```

250 +         boolean hasMoreResults,
251 +         Map<String, AttributeValue> finalLastEvaluatedKey) {
252 +     if (allOrders.size() > pageSize) {
253 +         allOrders = allOrders.subList(0, pageSize);
254 +         hasMoreResults = true;
255 +     }
256 +
257 +     @@ -234,6 +265,16 @@ public class OrderServiceV2 {
258 +         return new OrderSearchResponseV2(allOrders, lastEvaluatedKeyString);
259 +     }
260 +
261 +     private static class QueryResult {
262 +         final List<BffPortingRequestOrder> orders;
263 +         final Map<String, AttributeValue> lastEvaluatedKey;
264 +
265 +         QueryResult(List<BffPortingRequestOrder> orders, Map<String, AttributeValue>
266 +         lastEvaluatedKey) {
267 +             this.orders = orders;
268 +             this.lastEvaluatedKey = lastEvaluatedKey;
269 +         }
270 +     }
271 +
272 +     private boolean matchesCriteria(BffPortingRequestOrder order, OrderSearchCriteriaV2 criteria)
273 +     {
274 +         return (criteria.getOperator() == null
275 +                 || criteria.getOperator()
276 +
277 +     @@ -377,28 +418,21 @@ public class OrderServiceV2 {
278 +         List<BffPortingRequestOrder> allOrders = new ArrayList<>();
279 +         String lastEvaluatedKeyString = null;
280 +
281 +         // If we have an exclusiveStartKey, find the corresponding partition
282 +         String currentPartition = null;
283 +         if (exclusiveStartKey != null &&
284 +             exclusiveStartKey.containsKey(ORDER_TYPE_PORTING_STATUS)) {
285 +             currentPartition = exclusiveStartKey.get(ORDER_TYPE_PORTING_STATUS).s();
286 +             int startIndex = partitionValues.indexOf(currentPartition);
287 +             if (startIndex != -1) {
288 +                 partitionValues = partitionValues.subList(startIndex, partitionValues.size());
289 +             }
290 +
291 +         String currentPartition = getCurrentPartition(exclusiveStartKey, partitionValues);
292 +
293 +         for (String partitionValue : partitionValues) {
294 +             Map<String, AttributeValue> partitionExclusiveStartKey =
295 +                 (currentPartition != null && currentPartition.equals(partitionValue)) ?
296 +                     exclusiveStartKey : null;
297 +             getPartitionExclusiveStartKey(currentPartition, partitionValue,
298 +                 exclusiveStartKey);
299 +
300 +             while (true) {
301 +                 OrderSearchResponseV2 partitionResult = getOrdersForPartition(
302 +                     OrderSearchResponseV2 partitionResult = fetchOrdersForPartition(
303 +                         criteria,
304 +                         partitionValue,
305 +                         fromDate.

```

```

399 432           toDate,
400 433           partitionExclusiveStartKey,
401 -           criteria.getPageSize() - allOrders.size());
402 +           allOrders.size(),
403 +           criteria.getPageSize());
404 436           allOrders.addAll(partitionResult.getOrders());
405 437           lastEvaluatedKeyString = partitionResult.getLastEvaluatedKey();
406 440           @@ -406,7 +440,6 @@ public class OrderServiceV2 {
407 441               if (lastEvaluatedKeyString == null || allOrders.size() >= criteria.getPageSize())
408 442               {
409 -               break;
410 443               partitionExclusiveStartKey =
411 444                   LastEvaluatedKeySerializer.deserialize(lastEvaluatedKeyString);
412 445           }
413 446           @@ -415,10 +448,44 @@ public class OrderServiceV2 {
414 447               }
415 448           }
416 449           }
417 450           }
418 -           if (allOrders.size() > criteria.getPageSize()) {
419 -               allOrders = allOrders.sublist(0, criteria.getPageSize());
420 +           return getPagedResponse(allOrders, criteria.getPageSize(), lastEvaluatedKeyString);
421 +       }
422 +   }
423 +
424 +   private String getCurrentPartition(Map<String, AttributeValue> exclusiveStartKey,
425 +           List<String> partitionValues) {
426 +       if (exclusiveStartKey != null &&
427 +           exclusiveStartKey.containsKey(ORDER_TYPE_PORTING_STATUS)) {
428 +           String currentPartition =
429 +               exclusiveStartKey.get(ORDER_TYPE_PORTING_STATUS).s();
430 +           int startIndex = partitionValues.indexOf(currentPartition);
431 +           if (startIndex != -1) {
432 +               return currentPartition;
433 +           }
434 +       }
435 +       return null;
436 +   }
437 +
438 +   private Map<String, AttributeValue> getPartitionExclusiveStartKey(
439 +           String currentPartition, String partitionValue, Map<String, AttributeValue>
440 +           exclusiveStartKey) {
441 +       return (currentPartition != null && currentPartition.equals(partitionValue)) ?
442 +           exclusiveStartKey : null;
443 +   }
444 +
445 +   private OrderSearchResponseV2 fetchOrdersForPartition(
446 +           OrderSearchCriteriaV2 criteria,
447 +           String partitionValue,
448 +           String fromDate,
449 +           String toDate,
450 +           Map<String, AttributeValue> exclusiveStartKey,
451 +           int currentPageSize,

```

```

478 +         int pageSize) {
479 +
480 +     return getOrdersForPartition(
481 +         criteria, partitionValue, fromDate, toDate, exclusiveStartKey, pageSize -
482 +         currentPageSize);
483 +
484 +     private OrderSearchResponseV2 getPagedResponse(
485 +         List<BffPortingRequestOrder> allOrders, int pageSize, String lastEvaluatedKeyString) {
486 +         if (allOrders.size() > pageSize) {
487 +             allOrders = allOrders.subList(0, pageSize);
488 +         }
489 +
490 +         return new OrderSearchResponseV2(allOrders, lastEvaluatedKeyString);
491     }

```

↓

src/main/java/be/telenet/cnpservice/service/PaginatedTaskService.java				+8 -44	<input type="checkbox"/> Viewed		
1	1	package be.telenet.cnpservice.service;					
2	2						
3	3	- import be.telenet.cnpservice.model.Donor;					
4		- import be.telenet.cnpservice.model.PortingDetails;					
5		- import be.telenet.cnpservice.model.PortingRequest;					
6		- import be.telenet.cnpservice.model.Recipient;					
7	3	import be.telenet.cnpservice.model.Task;					
8		- import be.telenet.cnpservice.mapper.PortingMessageDTOMapper;					
9	4	import be.telenet.cnpservice.model.PaginatedTaskSearchCriteria;					
10	5	import be.telenet.cnpservice.model.PaginatedTaskSearchResponse;					
11	6	import be.telenet.cnpservice.repository.PaginatedTaskRepository;					
12	15	@@ -20,8 +15,6 @@ import java.util.HashMap;					
20	16	import java.util.Iterator;					
21	17	import java.util.List;					
22		import java.util.Map;					
23	18	- import java.util.function.Consumer;					
24		- import java.util.function.Function;					
25	19	import lombok.RequiredArgsConstructor;					
26	20	import lombok.extern.slf4j.Slf4j;					
27		import org.springframework.stereotype.Service;					
28	33	@@ -40,8 +33,6 @@ import software.amazon.awssdk.services.dynamodb.model.AttributeValue;					
40	34	public class PaginatedTaskService {					
41	35	private final PaginatedTaskRepository paginatedRepository;					

```

43      -     private final PortingMessageDTOMapper converter;
44      -     private final RequestService requestService;
45  36      private static final String TYPE_CREATION_DATE_GSI = "type-creation-date-gsi";
46  37      private static final String TASK_ID_GSI = "task-id-gsi";
47  38      private static final String FROM_DATE = ":fromDate";
    ↓
    ↑ @@ -272,16 +263,14 @@ public class PaginatedTaskService {
272 263          String expression;
273 264
274 265          if (fromDate != null && toDate != null) {
275      -             expressionValues.put(
276      -                 ":fromDate", AttributeValue.builder().s(fromDate).build());
277      -             expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
278  266      +             expressionValues.put(FROM_DATE, AttributeValue.builder().s(fromDate).build());
279  267      +             expressionValues.put(TO_DATE, AttributeValue.builder().s(toDate).build());
280      -             expression = "creationTime BETWEEN :fromDate AND :toDate";
281      -             expressionValues.put(
282      -                 ":fromDate", AttributeValue.builder().s(fromDate).build());
283  270      +             expressionValues.put(FROM_DATE, AttributeValue.builder().s(fromDate).build());
284      -             expression = "creationTime >= :fromDate";
285  273      +             expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
286  274      +             expression = "creationTime <= :toDate";
287  275      } else {
288      -             return null;
289
290      -             .build();
291
292      }
293  282
294  283
295  284
296      -     private Task enrichTask(Task task) {
297      -         log.info("Enriching task: {}", task);
298      -         if (task.getNpRequestId() != null) {
299      -             PortingRequest portingRequest =
300      -                 requestService.getPortingRequestById(task.getNpRequestId());
301      -             if (portingRequest != null) {
302      -                 PortingDetails portingDetails = portingRequest.getPortingDetails();
303      -                 if (portingDetails != null) {
304      -                     setCodeIfPresent(portingDetails.getDonor(), Donor::getCode, task::setDonor);
305      -                     setCodeIfPresent(portingDetails.getRecipient(), Recipient::getCode,
306      -                         task::setRecipient);
307      -                     }
308      -                     task.setOrderStatus(portingRequest.getPortingStatus());
309      -                 }
310      -             return task;
311      -         }
312
313      -     private <T> void setCodeIfPresent(T entity, Function<T, String> getCode, Consumer<String>
314      -                                         setCode) {
315      -         if (entity != null && getCode != null && setCode != null) {
316      -             setCode.accept(getCode.apply(entity));
317      -         }

```

```

317 318      }
319 285      private Expression filterExpression(
320 286          PaginatedTaskSearchCriteria criteria, boolean shouldFilterByDate, boolean
321 287          shouldFilterByOperator) {
322
323      Map<String, AttributeValue> expressionValues = new HashMap<>();
324
325      @@ -371,16 +337,14 @@ public class PaginatedTaskService {
326
327      private String filterExpressionForDate(
328          String fromDate, String toDate, Map<String, AttributeValue> expressionValues) {
329          if (fromDate != null && toDate != null) {
330              expressionValues.put(
331                  ":fromDate", AttributeValue.builder().s(fromDate).build());
332              expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
333
334          } else if (fromDate != null) {
335              expressionValues.put(
336                  ":fromDate", AttributeValue.builder().s(fromDate).build());
337
338          } else if (toDate != null) {
339              expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
340
341          return "creationTime BETWEEN :fromDate AND :toDate";
342
343      } else if (fromDate != null) {
344          expressionValues.put(
345              ":fromDate", AttributeValue.builder().s(fromDate).build());
346
347      } else if (toDate != null) {
348          expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
349
350      return "";
351
352  }
353
354  }
355
356  }
357
358  }
359
360  }
361
362  }
363
364  }
365
366  }
367
368  }
369
370  }
371
372  }
373
374  }
375
376  }
377
378  }
379
380  }
381
382  }
383
384  }
385
386  }

```

src/main/java/be/telenet/cnpservice/service/RequestService.java

↑	@@ -2,7 +2,10 @@ package be.telenet.cnpservice.service;	+5 -8	<input type="checkbox"/> Viewed		
2	import static be.telenet.cnp.data.model.PortingStatus.CLOSED;				
3					
4					
5	- import be.telenet.cnp.data.model.*;				
6	+ import be.telenet.cnp.data.model.PortingRequest;				
7	+ import be.telenet.cnp.data.model.PortingStatus;				
8	+ import be.telenet.cnp.data.model.Request;				
9	+ import be.telenet.cnp.data.model.Step;				
10	import be.telenet.cnpservice.exception.PortingRequestNotFoundException;				
11	import be.telenet.cnpservice.mapper.PortingMessageDTOMapper;				
12	import be.telenet.cnpservice.repository.MsisdnMappingRepository;				
13	@@ -10,18 +13,14 @@ import be.telenet.cnpservice.repository.RequestRepository;				
14	- import be.telenet.cnpservice.repository.StepRepository;				
15	import be.telenet.cnpservice.repository.model.MsisdnNpRequestIdMappingBean;				
16	import be.telenet.cnpservice.repository.model.PortingRequestBean;				
17	- import jakarta.xml.soap.SOAPException;				
18	- import java.io.IOException;				
19	import java.util.AbstractMap;				
20	import java.util.ArrayList;				
	import java.util.Collections;				
	import java.util.Comparator;				
	import java.util.List;				

```

20   - import javax.xml.parsers.ParserConfigurationException;
21   21   import lombok.extern.slf4j.Slf4j;
22   22   import org.springframework.context.annotation.Lazy;
23   23   import org.springframework.stereotype.Service;
24   - import org.xml.sax.SAXException;
25   24   import software.amazon.awssdk.enhanced.dynamodb.model.QueryEnhancedRequest;
26   25
27   26   @Service
28   ⇩ @@ -29,7 +28,6 @@ import software.amazon.awssdk.enhanced.dynamodb.model.QueryEnhancedRequest;
29   28   public class RequestService {
30   29       private static final String STEP = "STEP";
31   30       private static final String INFO = "INFO";
32   -
33   31       private static final String ID_GSI = "id-gsi";
34   32       private static final String NPR_NUMBER_GSI = "npr-number-gsi";
35   33       private static final String NP_REQUEST_GSI = "np-request-gsi";
36   ⇩ @@ -53,8 +51,7 @@ public class RequestService {
37   51           this.metricService = metricService;
38   52       }
39   -
40   51       public PortingRequest createPortingRequest(PortingRequest portingRequest)
41   52           throws SOAPException, IOException, ParserConfigurationException, SAXException {
42   53   +       public PortingRequest createPortingRequest(PortingRequest portingRequest) {
43   54           PortingRequestBean portInInfo =
44   55               converter.buildPortInInfoFromPortingRequest(portingRequest);
45   56           PortingRequestBean portInStep =
46   57               converter.buildPortInStepFromPortingRequest(portingRequest, portInInfo.getId());

```

src/main/java/be/telenet/cnpservice/service/TaskSearchService.java

```

↑ @@ -46,6 +46,7 @@ public class TaskSearchService {
46   46       private final RequestService requestService;
47   47       private static final String TASK_ID_GSI = "task-id-gsi";
48   48       private static final String TYPE_CREATION_DATE_GSI = "type-creation-date-gsi";
49   49   +       private static final String TASK_ID = "taskId";
50   50
51   51       public PaginatedTaskSearchResponse searchTasks(PaginatedTaskSearchCriteria criteria) {
52           List<Task> allTasks = new ArrayList<>();
53   ⇩
54   ⇩ @@ -433,8 +434,8 @@ public class TaskSearchService {
55   433   434           if (StringUtils.hasText(lastEvaluatedKey)) {
56   435               Map<String,AttributeValue> lastEvaluatedKeyMap =
57   436                   LastEvaluatedKeySerializer.deserialize(lastEvaluatedKey);
58   437   -               if (lastEvaluatedKeyMap.containsKey("taskId")) {
59   438   -                   builder.exclusiveStartKey(Map.of("taskId", lastEvaluatedKeyMap.get("taskId")));
60   439   +               if (lastEvaluatedKeyMap.containsKey(TASK_ID)) {
61   440   +                   builder.exclusiveStartKey(Map.of(TASK_ID, lastEvaluatedKeyMap.get(TASK_ID)));
62   441               }
63   442           }
64   443   ↓

```

src/main/java/be/telenet/cnpservice/service/TaskService.java

+11 -74 Viewed

```

@@ -2,7 +2,8 @@ package be.telenet.cnpservice.service;

2   2     import static be.telenet.cnpservice.model.TaskStatus.CLOSED;
3   3     import static be.telenet.cnpservice.model.TaskStatus.RESUBMIT;
4   4
5   5 - import static be.telenet.cnpservice.mapper.PortingMessageDTOMapper.currentTime;
6   6 + import static be.telenet.cnpservice.util.FilterUtil.*;
7   7 + import static be.telenet.cnpservice.util.TimeProvider.currentTimeMillis;
8   8
9   9     import be.telenet.cnpservice.model.Donor;
10 10    import be.telenet.cnpservice.model.Operator;

@@ -21,7 +22,6 @@ import be.telenet.cnpservice.repository.TaskRepository;
11 22    import be.telenet.cnpservice.repository.model.MsisdnTaskMappingBean;
12 23    import be.telenet.cnpservice.repository.model.TaskBean;
13 24    import be.telenet.cnpservice.repository.model.TaskSearchCriteria;
14 24 - import jakarta.xml.soap.SOAPException;
15 25    import java.io.IOException;
16 26    import java.time.ZoneId;
17 27    import java.time.ZonedDateTime;

@@ -34,12 +34,10 @@ import java.util.Map;
34 34    import java.util.Optional;
35 35    import java.util.function.Consumer;
36 36    import java.util.function.Function;
37 37 - import javax.xml.parsers.ParserConfigurationException;
38 37    import lombok.RequiredArgsConstructor;
39 38    import lombok.extern.slf4j.Slf4j;
40 39    import org.springframework.stereotype.Service;
41 40    import org.springframework.util.StringUtils;
42 42 - import org.xml.sax.SAXException;
43 41    import software.amazon.awssdk.enhanced.dynamodb.Expression;
44 42    import software.amazon.awssdk.enhanced.dynamodb.Key;
45 43    import software.amazon.awssdk.enhanced.dynamodb.model.QueryConditional;

@@ -62,8 +60,7 @@ public class TaskService {
62 60    private final RequestService requestService;
63 61    private final MetricHandler metricHandler;
64 62
65 65 -     public Task createPortingTask(Task input)
66 66 -         throws IOException, SOAPException, ParserConfigurationException, SAXException {
67 67 +     public Task createPortingTask(Task input) throws IOException {
68 68         TaskBean taskInfo = converter.buildTaskBeanFromTask(input);
69 69         TaskBean taskSubFlow = converter.buildTaskSubFlowBean(input, taskInfo.getTaskId());
log.debug("taskInfo" + taskInfo);

@@ -95,7 +92,7 @@ public class TaskService {
95 92     List<TaskBean> taskBeanList = getTaskItems(taskId, TASK_ID_GSI);
96 93     log.info("Attempting to retrieve task with ID: {}", taskBeanList);
97 94     Optional<Task> task = getInfoAndConvertTaskBean(taskBeanList, taskId);
98 98 -     if (!task.isEmpty() && null != task.get().getNpRequestId()) {

```

```

      95 +     if (task.isPresent() && null != task.get().getNpRequestId()) {
      96         PortingRequest portingRequest =
      97             requestService.getPortingRequestById(task.get().getNpRequestId());
      98             task.get().setOrderStatus(portingRequest.getPortingStatus());
      ↓
      ↑ @@ -165,8 +162,7 @@ public class TaskService {
  165 162             .withZoneSameInstant(ZoneId.of("Europe/Brussels"))
  166 163                 .toString();
  167 164             return partitionValues(taskType, taskStatus).stream()
  168 -                 .map(partitionValue ->
  169 -                     getTasksWithToDate(taskSearchCriteria, partitionValue, date,
  TYPE_CREATION_DATE_GSI))
  165 +                 .map(partitionValue -> getTasksWithToDate(taskSearchCriteria, partitionValue,
  date))
  170 166             .flatMap(List::stream)
  171 167             .toList();
  172 168         } else {
      ↓
      ↑ @@ -237,8 +233,7 @@ public class TaskService {
  237 233             private List<Task> streamTasksWithFromDate(
  238 234                 List<String> partitionValues, TaskSearchCriteria taskSearchCriteria, String fromDate)
  239 235             {
  240                 return partitionValues.stream()
  241 -                     .map(partitionValue ->
  242 -                         getTasksWithFromDate(taskSearchCriteria, partitionValue, fromDate,
  TYPE_CREATION_DATE_GSI))
  236 +                     .map(partitionValue -> getTasksWithFromDate(taskSearchCriteria, partitionValue,
  242 237                         .flatMap(List::stream)
  243 238                         .toList());
  244 239         }
      ⇤ @@ -246,8 +241,7 @@ public class TaskService {
  246 241             private List<Task> streamTasksWithToDate(
  247 242                 List<String> partitionValues, TaskSearchCriteria taskSearchCriteria, String toDate) {
  248 243                 return partitionValues.stream()
  249 -                     .map(partitionValue ->
  250 -                         getTasksWithToDate(taskSearchCriteria, partitionValue, toDate,
  TYPE_CREATION_DATE_GSI))
  244 +                     .map(partitionValue -> getTasksWithToDate(taskSearchCriteria, partitionValue,
  toDate))
  251 245             .flatMap(List::stream)
  252 246             .toList();
  253 247         }
      ↓
      ↑ @@ -304,7 +298,7 @@ public class TaskService {
  304 298         }
  305 299
  306 300             private List<Task> getTasksWithFromDate(
  307 -                 TaskSearchCriteria taskSearchCriteria, String partitionValue, String fromDate, String
  index) {
  301 +                 TaskSearchCriteria taskSearchCriteria, String partitionValue, String fromDate) {
  308 302                 Expression filterExpression = filterExpression(taskSearchCriteria, false, true);
  309 303                 QueryConditional queryConditional = QueryConditional.sortGreaterThanOrEqualTo(
  310 304                     Key.builder().partitionValue(partitionValue).sortValue(fromDate).build());

```

```

316 318 : QueryEnhancedRequest.builder()
317 311 .queryConditional(queryConditional)
318 312 .build();
319 -     return repository.getTasks(request, index).stream()
320 313 +     return repository.getTasks(request, TaskService.TYPE_CREATION_DATE_GSI).stream()
321 314 .map(converter::buildTaskFromTaskBean)
322 316 .toList();
323 317 }
324 -     private List<Task> getTasksWithToDate(
325 -             TaskSearchCriteria taskSearchCriteria, String partitionValue, String toDate, String
326 318 +             private List<Task> getTasksWithToDate(TaskSearchCriteria taskSearchCriteria, String
327 319 partitionValue, String toDate) {
328 320         Expression filterExpression = filterExpression(taskSearchCriteria, false, true);
329 321         QueryConditional queryConditional = QueryConditional.sortLessThanOrEqualTo(
330 322             Key.builder().partitionValue(partitionValue).sortValue(toDate).build());
331 323     @@ -334,7 +327,7 @@ public class TaskService {
332 324         : QueryEnhancedRequest.builder()
333 325             .queryConditional(queryConditional)
334 327             .build();
335 328 }
336 329 -     return repository.getTasks(request, index).stream()
337 330 +     return repository.getTasks(request, TaskService.TYPE_CREATION_DATE_GSI).stream()
338 331 .map(converter::buildTaskFromTaskBean)
339 332 .toList();
340 333 }

369 369     .build();
370 370 }
371 371 }
372 372 }
373 373 }
374 374 }
375 375 }
376 376     private Expression filterForOperator(
377 377     TaskSearchCriteria taskSearchCriteria, String expression, Map<String, AttributeValue>
378 378 expressionValues) {
379 379     if (!expression.isEmpty()) {
380 380         expression = expression.concat(" AND ");
381 381     }
382 382     expression = expression.concat("#op = :operator");
383 383     expressionValues.put(
384 384         ":operator",
385 385         AttributeValue.builder()
386 386             .s(taskSearchCriteria.getOperator().name())
387 387             .build());
388 388     return Expression.builder()
389 389         .expression(expression)
390 390         .expressionValues(expressionValues)
391 391         .expressionNames(Map.of("#op", "operator"))
392 392         .build();
393 393     }
394 394 }
395 395 }
396 396 }
397 397     private String filterExpressionForDate(
398 398     String fromDate, String toDate, Map<String, AttributeValue> expressionValues) {
399 399     if (fromDate != null && toDate != null) {
400 400         expressionValues.put(
401 401             ":fromDate", AttributeValue.builder().s(fromDate).build());

```

```

402      -         expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
403      -         return "creationTime BETWEEN :fromDate AND :toDate";
404      -     } else if (fromDate != null) {
405      -         expressionValues.put(
406      -             ":fromDate", AttributeValue.builder().s(fromDate).build());
407      -         return "creationTime >= :fromDate";
408      -     } else if (toDate != null) {
409      -         expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
410      -         return "creationTime <= :toDate";
411      -     }
412      -     return "";
413  }
414
415  private QueryEnhancedRequest requestWithFilterExpression(
416      String partitionValue, String sortValue, Expression filterExpression) {
417      return QueryEnhancedRequest.builder()
418          .queryConditional(QueryConditional.keyEqualTo(Key.builder()
419              .partitionValue(partitionValue)
420              .sortValue(sortValue)
421              .build()))
422          .filterExpression(filterExpression)
423          .build();
424  }
425
426  private QueryEnhancedRequest requestWithoutFilterExpression(String partitionValue, String
427      sortValue) {
428      return QueryEnhancedRequest.builder()
429          .queryConditional(QueryConditional.keyEqualTo(Key.builder()
430              .partitionValue(partitionValue)
431              .sortValue(sortValue)
432              .build()))
433          .build();
434  }
435  372  private Optional<Task> getInfoAndConvertTaskBean(List<TaskBean> taskBeanList, String taskId) {
436  373      return taskBeanList.stream()
437  374          .filter(taskBean ->

```

↓

src/main/java/be/telenet/cnpservice/util/EnhancedLastEvaluatedKeySerializer.java		+2 -1	<input type="checkbox"/> Viewed		
↑	@@ -2,6 +2,7 @@ package be.telenet.cnpservice.util;				
2 2	import java.io.UncheckedIOException;				
3 3	import java.util.Base64;				
4 4	+ import java.util.Collections;				
5 6	import java.util.Map;				
6 7	import software.amazon.awssdk.enhanced.dynamodb.document EnhancedDocument;				
7 8	import software.amazon.awssdk.services.dynamodb.model.AttributeValue;				
↓	@@ -31,7 +32,7 @@ public class EnhancedLastEvaluatedKeySerializer {				
31 32	*/				
32 33	public static Map<String, AttributeValue> deserialize(final String paginationToken) throws				
33 34	UncheckedIOException {				
34 -	if (paginationToken == null paginationToken.isEmpty()) {				
35 +	return null;				
35 36	return Collections.emptyMap();				
36 37	}				
37 38	String json = new String(Base64.getDecoder().decode(paginationToken));				
↓	return EnhancedDocument.fromJson(json).toMap();				

src/main/java/be/telenet/cnpservice/util/FilterUtil.java 0 → 100644 +83 -0 Viewed

```

1+ package be.telenet.cnpservice.util;
2+
3+ import be.telenet.cnpservice.repository.model.OrderSearchCriteria;
4+ import be.telenet.cnpservice.repository.model.TaskSearchCriteria;
5+ import java.util.Map;
6+ import software.amazon.awssdk.enhanced.dynamodb.Expression;
7+ import software.amazon.awssdk.enhanced.dynamodb.Key;
8+ import software.amazon.awssdk.enhanced.dynamodb.model.QueryConditional;
9+ import software.amazon.awssdk.enhanced.dynamodb.model.QueryEnhancedRequest;
10+ import software.amazon.awssdk.services.dynamodb.model.AttributeValue;
11+
12+ public class FilterUtil {
13+
14+     private FilterUtil() {
15+         throw new UnsupportedOperationException("ServiceUtil is a utility class and cannot be instantiated.");
16+     }
17+
18+     public static Expression filterForOperator(
19+         Object searchCriteria, String expression, Map<String, AttributeValue>
20+         expressionValues) {
21+
22+         String operator = null;
23+
24+         if (searchCriteria instanceof TaskSearchCriteria taskSearchCriteria) {
25+             operator = taskSearchCriteria.getOperator().name();
26+
27+         } else if (searchCriteria instanceof OrderSearchCriteria orderSearchCriteria) {
28+             operator = orderSearchCriteria.getOperator().name();
29+
30+             if (operator == null) {
31+                 throw new IllegalArgumentException("Unsupported search criteria type or operator not found.");
32+             }
33+
34+             if (!expression.isEmpty()) {
35+                 expression = expression.concat(" AND ");
36+             }
37+             expression = expression.concat("#op = :operator");
38+             expressionValues.put(":operator", AttributeValue.builder().s(operator).build());
39+
40+             return Expression.builder()
41+                 .expression(expression)
42+                 .expressionValues(expressionValues)
43+                 .expressionNames(Map.of("#op", "operator"))
44+                 .build();
45+
46+         public static String filterExpressionForDate(
47+             String fromDate, String toDate, Map<String, AttributeValue> expressionValues) {
48+             if (fromDate != null && toDate != null) {
49+                 expressionValues.put(
50+                     ":fromDate", AttributeValue.builder().s(fromDate).build());
51+             }
52+         }
53+     }
54+
55+     public static void main(String[] args) {
56+         System.out.println(filterForOperator(OrderSearchCriteria.builder().build(), "id", Map.of("id", AttributeValue.builder().s("1").build())));
57+     }
58+ }

```

```

51 +         expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
52 +         return "creationTime BETWEEN :fromDate AND :toDate";
53 +     } else if (fromDate != null) {
54 +         expressionValues.put(
55 +             ":fromDate", AttributeValue.builder().s(fromDate).build());
56 +         return "creationTime >= :fromDate";
57 +     } else if (toDate != null) {
58 +         expressionValues.put(":toDate", AttributeValue.builder().s(toDate).build());
59 +         return "creationTime <= :toDate";
60 +     }
61 +     return "";
62 +
63 +
64 +    public static QueryEnhancedRequest requestWithFilterExpression(
65 +        String partitionValue, String sortValue, Expression filterExpression) {
66 +        return QueryEnhancedRequest.builder()
67 +            .queryConditional(QueryConditional.keyEqualTo(Key.builder()
68 +                .partitionValue(partitionValue)
69 +                .sortValue(sortValue)
70 +                .build()))
71 +            .filterExpression(filterExpression)
72 +            .build();
73 +
74 +
75 +    public static QueryEnhancedRequest requestWithoutFilterExpression(String partitionValue,
76 +        String sortValue) {
77 +        return QueryEnhancedRequest.builder()
78 +            .queryConditional(QueryConditional.keyEqualTo(Key.builder()
79 +                .partitionValue(partitionValue)
80 +                .sortValue(sortValue)
81 +                .build()))
82 +            .build();
83 +    }

```

src/main/java/be/telenet/cnpservice/util/PortingMapperUtil.java 0 → 100644 +111 -0 Viewed

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import be.telenet.cnp.data.model.BffPortingRequestOrder;
4 + import be.telenet.cnp.data.model.MessageType;
5 + import be.telenet.cnp.data.model.OrderType;
6 + import be.telenet.cnp.data.model.PortingRequest;
7 + import be.telenet.cnp.data.model.Step;
8 + import be.telenet.cnp.data.model.Systems;
9 + import be.telenet.cnp.data.model.Task;
10 + import be.telenet.cnp.data.model.TaskStatus;
11 + import be.telenet.cnp.data.model.TaskType;
12 + import java.util.UUID;
13 + import lombok.extern.slf4j.Slf4j;
14 + import org.springframework.util.StringUtils;
15 +
16 + @Slf4j
17 + public class PortingMapperUtil {
18 +     private static final String TYPE_STATUS_PATTERN = "%s#%s";
19 +     private static final String STEP = "STEP#";
20 +     private static final String ORDER_CLOSED = "ORDER_CLOSED";
21 +     private static final String UNKNOWN = "UNKNOWN";
22 +     private static final String RESUBMIT = "RESUBMIT";
23 +

```

```

24 +     private PortingMapperUtil() {
25 +         throw new UnsupportedOperationException("MapperUtil is a utility class and cannot be
instantiated.");
26 +
27 +
28 +     public static String nonEmptyString(String value) {
29 +         return StringUtils.hasLength(value) ? value : null;
30 +     }
31 +
32 +     public static String buildPartitionKey(OrderType orderType, MessageType messageType, String
portingRequestId) {
33 +         return orderType
34 +             .name()
35 +             .concat("#")
36 +             .concat(messageType.getMessage())
37 +             .concat("#")
38 +             .concat(portingRequestId);
39 +
40 +
41 +     public static String buildSortKeyForInfo(String portingRequestId) {
42 +         return "INFO#".concat(portingRequestId);
43 +
44 +
45 +     public static String modificationUser(PortingRequest request) {
46 +         return null != request.getModificationUser() ? request.getModificationUser() :
Systems.CNP.name();
47 +
48 +
49 +     public static String orderTypePortingStatus(BffPortingRequestOrder bffPortingRequestOrder) {
50 +         return TYPE_STATUS_PATTERN.formatted(
51 +             bffPortingRequestOrder.getOrderType(), bffPortingRequestOrder.getPortingStatus());
52 +
53 +
54 +     public static String taskTypeStatusScenario(Task task) {
55 +         return null != task.getTaskTypeStatusScenario()
56 +             ? task.getTaskTypeStatusScenario()
57 +             : TYPE_STATUS_PATTERN.formatted(task.getType(), task.getStatus());
58 +
59 +
60 +     public static Object updateCheck(Object newValue, Object oldValue) {
61 +         return null != newValue ? newValue : oldValue;
62 +
63 +
64 +     public static String creationUser(Step step) {
65 +         return null != step.getCreationUser() ? step.getCreationUser() : Systems.CNP.name();
66 +
67 +
68 +     public static String buildSortKeyForStep(String stepName, String stepId, String npRequestId) {
69 +         return stepName.equals(ORDER_CLOSED) ? stepSortKey(stepName, npRequestId) :
stepSortKey(stepName, stepId);
70 +
71 +
72 +     public static String customerNumber(Task task) {
73 +         return StringUtils.hasLength(task.getCustomerNumber()) ? task.getCustomerNumber() : null;
74 +

```

```

76 +     public static String generateTaskPartitionKey(TaskType taskType, OrderType orderType, String
77 +         taskId) {
78 +             String orderTypeValue = null != orderType ? orderType.name() : UNKNOWN;
79 +             return taskType.name().concat("#").concat(orderTypeValue).concat("#").concat(taskId);
80 +
81 +     public static String taskAction(TaskStatus status) {
82 +         return null != status ? status.name() : RESUBMIT;
83 +     }
84 +
85 +     public static String taskSubFlowId() {
86 +         return UUID.randomUUID().toString();
87 +     }
88 +
89 +     public static TaskStatus taskNewStatus(TaskStatus newStatus) {
90 +         return TaskStatus.RESUBMIT != newStatus ? newStatus : null;
91 +     }
92 +
93 +     public static TaskStatus taskOldStatus(TaskStatus oldStatus, TaskStatus newStatus) {
94 +         return TaskStatus.RESUBMIT != newStatus ? oldStatus : null;
95 +     }
96 +
97 +     public static String getStepId(String sortKey) {
98 +
99 +         String[] parts = sortKey.split("#");
100 +        if (parts.length > 2) {
101 +            return parts[2];
102 +        } else {
103 +            log.info("UUID not found.");
104 +        }
105 +        return null;
106 +    }
107 +
108 +    private static String stepSortKey(String stepName, String id) {
109 +        return STEP.concat(stepName).concat("#").concat(id);
110 +    }
111 +}

```

src/main/java/be/telenet/cnpservice/util/TimeProvider.java 0 → 100644 +17 -0 Viewed

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import java.time.ZoneId;
4 + import java.time.ZonedDateTime;
5 +
6 + public class TimeProvider {
7 +
8 +     private static final String ZONE = "Europe/Brussels";
9 +
10 +    private TimeProvider() {
11 +        throw new UnsupportedOperationException("TimeProvider is a utility class and cannot be
12 +        instantiated.");
13 +    }
14 +
15 +    public static ZonedDateTime currentTime() {
16 +        return ZonedDateTime.now().withZoneSameInstant(ZoneId.of(ZONE));
17 +    }

```

src/test/java/be/telenet/cnpservice/controller/OrderControllerTest.java 0 → +102 -0 Viewed

```

1 + package be.telenet.cnpservice.controller;
2 +
3 + import static org.mockito.Mockito.*;
4 + import static org.springframework.http.MediaType.APPLICATION_JSON;
5 + import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*;
6 + import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*;
7 +
8 + import be.telenet.cnpservice.data.model.BffPortingRequestOrder;
9 + import be.telenet.cnpservice.data.model.Operator;
10 + import be.telenet.cnpservice.data.model.OrderType;
11 + import be.telenet.cnpservice.data.model.PortingStatus;
12 + import be.telenet.cnpservice.repository.model.OrderSearchCriteria;
13 + import be.telenet.cnpservice.service.OrderService;
14 + import com.fasterxml.jackson.databind.ObjectMapper;
15 + import java.time.ZonedDateTime;
16 + import java.util.Collections;
17 + import java.util.List;
18 + import org.junit.jupiter.api.BeforeEach;
19 + import org.junit.jupiter.api.Test;
20 + import org.mockito.Mockito;
21 + import org.springframework.beans.factory.annotation.Autowired;
22 + import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
23 + import org.springframework.boot.test.mock.mockito.MockBean;
24 + import org.springframework.test.web.servlet.MockMvc;
25 +
26 + @WebMvcTest(OrderController.class)
27 + class OrderControllerTest {
28 +
29 +     @Autowired
30 +     private MockMvc mockMvc;
31 +
32 +     @MockBean
33 +     private OrderService orderService;
34 +
35 +     @Autowired
36 +     private ObjectMapper objectMapper;
37 +
38 +     private BffPortingRequestOrder sampleOrder;
39 +
40 +     @BeforeEach
41 +     void setUp() {
42 +         sampleOrder = new BffPortingRequestOrder();
43 +         sampleOrder.setPartitionKey("partitionKey123");
44 +         sampleOrder.setSortKey("sortKey123");
45 +         sampleOrder.setNpRequestId("npRequestId123");
46 +         sampleOrder.setCustomerNumber("customer123");
47 +         sampleOrder.setRecipientCode("recipient123");
48 +         sampleOrder.setDonorCode("donor123");
49 +         sampleOrder.setMsisdn(Collections.singletonList("0123456789"));
50 +         sampleOrder.setNprNumber("9876543210");
51 +     }
52 +
53 +     @Test
54 +     void testGetOrder() throws Exception {
55 +         // Arrange
56 +         // Act
57 +         // Assert
58 +     }
59 +
60 +     @Test
61 +     void testUpdateOrder() throws Exception {
62 +         // Arrange
63 +         // Act
64 +         // Assert
65 +     }
66 +
67 +     @Test
68 +     void testDeleteOrder() throws Exception {
69 +         // Arrange
70 +         // Act
71 +         // Assert
72 +     }
73 +
74 +     @Test
75 +     void testSearchOrders() throws Exception {
76 +         // Arrange
77 +         // Act
78 +         // Assert
79 +     }
80 +
81 +     @Test
82 +     void testCreateOrder() throws Exception {
83 +         // Arrange
84 +         // Act
85 +         // Assert
86 +     }
87 +
88 +     @Test
89 +     void testGetOrderStatus() throws Exception {
90 +         // Arrange
91 +         // Act
92 +         // Assert
93 +     }
94 +
95 +     @Test
96 +     void testUpdateOrderStatus() throws Exception {
97 +         // Arrange
98 +         // Act
99 +         // Assert
100 +    }
101 +
102 +    @Test
103 +    void testGetOrderHistory() throws Exception {
104 +        // Arrange
105 +        // Act
106 +        // Assert
107 +    }
108 +
109 +    @Test
110 +    void testUpdateOrderHistory() throws Exception {
111 +        // Arrange
112 +        // Act
113 +        // Assert
114 +    }
115 +
116 +    @Test
117 +    void testGetOrderDetails() throws Exception {
118 +        // Arrange
119 +        // Act
120 +        // Assert
121 +    }
122 +
123 +    @Test
124 +    void testUpdateOrderDetails() throws Exception {
125 +        // Arrange
126 +        // Act
127 +        // Assert
128 +    }
129 +
130 +    @Test
131 +    void testGetOrderLogs() throws Exception {
132 +        // Arrange
133 +        // Act
134 +        // Assert
135 +    }
136 +
137 +    @Test
138 +    void testUpdateOrderLogs() throws Exception {
139 +        // Arrange
140 +        // Act
141 +        // Assert
142 +    }
143 +
144 +    @Test
145 +    void testGetOrderMetrics() throws Exception {
146 +        // Arrange
147 +        // Act
148 +        // Assert
149 +    }
150 +
151 +    @Test
152 +    void testUpdateOrderMetrics() throws Exception {
153 +        // Arrange
154 +        // Act
155 +        // Assert
156 +    }
157 +
158 +    @Test
159 +    void testGetOrderAnalytics() throws Exception {
160 +        // Arrange
161 +        // Act
162 +        // Assert
163 +    }
164 +
165 +    @Test
166 +    void testUpdateOrderAnalytics() throws Exception {
167 +        // Arrange
168 +        // Act
169 +        // Assert
170 +    }
171 +
172 +    @Test
173 +    void testGetOrderReports() throws Exception {
174 +        // Arrange
175 +        // Act
176 +        // Assert
177 +    }
178 +
179 +    @Test
180 +    void testUpdateOrderReports() throws Exception {
181 +        // Arrange
182 +        // Act
183 +        // Assert
184 +    }
185 +
186 +    @Test
187 +    void testGetOrderStatistics() throws Exception {
188 +        // Arrange
189 +        // Act
190 +        // Assert
191 +    }
192 +
193 +    @Test
194 +    void testUpdateOrderStatistics() throws Exception {
195 +        // Arrange
196 +        // Act
197 +        // Assert
198 +    }
199 +
200 +    @Test
201 +    void testGetOrderAuditLogs() throws Exception {
202 +        // Arrange
203 +        // Act
204 +        // Assert
205 +    }
206 +
207 +    @Test
208 +    void testUpdateOrderAuditLogs() throws Exception {
209 +        // Arrange
210 +        // Act
211 +        // Assert
212 +    }
213 +
214 +    @Test
215 +    void testGetOrderComplianceLogs() throws Exception {
216 +        // Arrange
217 +        // Act
218 +        // Assert
219 +    }
220 +
221 +    @Test
222 +    void testUpdateOrderComplianceLogs() throws Exception {
223 +        // Arrange
224 +        // Act
225 +        // Assert
226 +    }
227 +
228 +    @Test
229 +    void testGetOrderRiskLogs() throws Exception {
230 +        // Arrange
231 +        // Act
232 +        // Assert
233 +    }
234 +
235 +    @Test
236 +    void testUpdateOrderRiskLogs() throws Exception {
237 +        // Arrange
238 +        // Act
239 +        // Assert
240 +    }
241 +
242 +    @Test
243 +    void testGetOrderFraudLogs() throws Exception {
244 +        // Arrange
245 +        // Act
246 +        // Assert
247 +    }
248 +
249 +    @Test
250 +    void testUpdateOrderFraudLogs() throws Exception {
251 +        // Arrange
252 +        // Act
253 +        // Assert
254 +    }
255 +
256 +    @Test
257 +    void testGetOrderComplianceReports() throws Exception {
258 +        // Arrange
259 +        // Act
260 +        // Assert
261 +    }
262 +
263 +    @Test
264 +    void testUpdateOrderComplianceReports() throws Exception {
265 +        // Arrange
266 +        // Act
267 +        // Assert
268 +    }
269 +
270 +    @Test
271 +    void testGetOrderRiskReports() throws Exception {
272 +        // Arrange
273 +        // Act
274 +        // Assert
275 +    }
276 +
277 +    @Test
278 +    void testUpdateOrderRiskReports() throws Exception {
279 +        // Arrange
280 +        // Act
281 +        // Assert
282 +    }
283 +
284 +    @Test
285 +    void testGetOrderFraudReports() throws Exception {
286 +        // Arrange
287 +        // Act
288 +        // Assert
289 +    }
290 +
291 +    @Test
292 +    void testUpdateOrderFraudReports() throws Exception {
293 +        // Arrange
294 +        // Act
295 +        // Assert
296 +    }
297 +
298 +    @Test
299 +    void testGetOrderComplianceMetrics() throws Exception {
300 +        // Arrange
301 +        // Act
302 +        // Assert
303 +    }
304 +
305 +    @Test
306 +    void testUpdateOrderComplianceMetrics() throws Exception {
307 +        // Arrange
308 +        // Act
309 +        // Assert
310 +    }
311 +
312 +    @Test
313 +    void testGetOrderRiskMetrics() throws Exception {
314 +        // Arrange
315 +        // Act
316 +        // Assert
317 +    }
318 +
319 +    @Test
320 +    void testUpdateOrderRiskMetrics() throws Exception {
321 +        // Arrange
322 +        // Act
323 +        // Assert
324 +    }
325 +
326 +    @Test
327 +    void testGetOrderFraudMetrics() throws Exception {
328 +        // Arrange
329 +        // Act
330 +        // Assert
331 +    }
332 +
333 +    @Test
334 +    void testUpdateOrderFraudMetrics() throws Exception {
335 +        // Arrange
336 +        // Act
337 +        // Assert
338 +    }
339 +
340 +    @Test
341 +    void testGetOrderComplianceAnalytics() throws Exception {
342 +        // Arrange
343 +        // Act
344 +        // Assert
345 +    }
346 +
347 +    @Test
348 +    void testUpdateOrderComplianceAnalytics() throws Exception {
349 +        // Arrange
350 +        // Act
351 +        // Assert
352 +    }
353 +
354 +    @Test
355 +    void testGetOrderRiskAnalytics() throws Exception {
356 +        // Arrange
357 +        // Act
358 +        // Assert
359 +    }
360 +
361 +    @Test
362 +    void testUpdateOrderRiskAnalytics() throws Exception {
363 +        // Arrange
364 +        // Act
365 +        // Assert
366 +    }
367 +
368 +    @Test
369 +    void testGetOrderFraudAnalytics() throws Exception {
370 +        // Arrange
371 +        // Act
372 +        // Assert
373 +    }
374 +
375 +    @Test
376 +    void testUpdateOrderFraudAnalytics() throws Exception {
377 +        // Arrange
378 +        // Act
379 +        // Assert
380 +    }
381 +
382 +    @Test
383 +    void testGetOrderComplianceReports() throws Exception {
384 +        // Arrange
385 +        // Act
386 +        // Assert
387 +    }
388 +
389 +    @Test
390 +    void testUpdateOrderComplianceReports() throws Exception {
391 +        // Arrange
392 +        // Act
393 +        // Assert
394 +    }
395 +
396 +    @Test
397 +    void testGetOrderRiskReports() throws Exception {
398 +        // Arrange
399 +        // Act
400 +        // Assert
401 +    }
402 +
403 +    @Test
404 +    void testUpdateOrderRiskReports() throws Exception {
405 +        // Arrange
406 +        // Act
407 +        // Assert
408 +    }
409 +
410 +    @Test
411 +    void testGetOrderFraudReports() throws Exception {
412 +        // Arrange
413 +        // Act
414 +        // Assert
415 +    }
416 +
417 +    @Test
418 +    void testUpdateOrderFraudReports() throws Exception {
419 +        // Arrange
420 +        // Act
421 +        // Assert
422 +    }
423 +
424 +    @Test
425 +    void testGetOrderComplianceMetrics() throws Exception {
426 +        // Arrange
427 +        // Act
428 +        // Assert
429 +    }
430 +
431 +    @Test
432 +    void testUpdateOrderComplianceMetrics() throws Exception {
433 +        // Arrange
434 +        // Act
435 +        // Assert
436 +    }
437 +
438 +    @Test
439 +    void testGetOrderRiskMetrics() throws Exception {
440 +        // Arrange
441 +        // Act
442 +        // Assert
443 +    }
444 +
445 +    @Test
446 +    void testUpdateOrderRiskMetrics() throws Exception {
447 +        // Arrange
448 +        // Act
449 +        // Assert
450 +    }
451 +
452 +    @Test
453 +    void testGetOrderFraudMetrics() throws Exception {
454 +        // Arrange
455 +        // Act
456 +        // Assert
457 +    }
458 +
459 +    @Test
460 +    void testUpdateOrderFraudMetrics() throws Exception {
461 +        // Arrange
462 +        // Act
463 +        // Assert
464 +    }
465 +
466 +    @Test
467 +    void testGetOrderComplianceAnalytics() throws Exception {
468 +        // Arrange
469 +        // Act
470 +        // Assert
471 +    }
472 +
473 +    @Test
474 +    void testUpdateOrderComplianceAnalytics() throws Exception {
475 +        // Arrange
476 +        // Act
477 +        // Assert
478 +    }
479 +
480 +    @Test
481 +    void testGetOrderRiskAnalytics() throws Exception {
482 +        // Arrange
483 +        // Act
484 +        // Assert
485 +    }
486 +
487 +    @Test
488 +    void testUpdateOrderRiskAnalytics() throws Exception {
489 +        // Arrange
490 +        // Act
491 +        // Assert
492 +    }
493 +
494 +    @Test
495 +    void testGetOrderFraudAnalytics() throws Exception {
496 +        // Arrange
497 +        // Act
498 +        // Assert
499 +    }
500 +
501 +    @Test
502 +    void testUpdateOrderFraudAnalytics() throws Exception {
503 +        // Arrange
504 +        // Act
505 +        // Assert
506 +    }
507 +
508 +    @Test
509 +    void testGetOrderComplianceReports() throws Exception {
510 +        // Arrange
511 +        // Act
512 +        // Assert
513 +    }
514 +
515 +    @Test
516 +    void testUpdateOrderComplianceReports() throws Exception {
517 +        // Arrange
518 +        // Act
519 +        // Assert
520 +    }
521 +
522 +    @Test
523 +    void testGetOrderRiskReports() throws Exception {
524 +        // Arrange
525 +        // Act
526 +        // Assert
527 +    }
528 +
529 +    @Test
530 +    void testUpdateOrderRiskReports() throws Exception {
531 +        // Arrange
532 +        // Act
533 +        // Assert
534 +    }
535 +
536 +    @Test
537 +    void testGetOrderFraudReports() throws Exception {
538 +        // Arrange
539 +        // Act
540 +        // Assert
541 +    }
542 +
543 +    @Test
544 +    void testUpdateOrderFraudReports() throws Exception {
545 +        // Arrange
546 +        // Act
547 +        // Assert
548 +    }
549 +
550 +    @Test
551 +    void testGetOrderComplianceMetrics() throws Exception {
552 +        // Arrange
553 +        // Act
554 +        // Assert
555 +    }
556 +
557 +    @Test
558 +    void testUpdateOrderComplianceMetrics() throws Exception {
559 +        // Arrange
560 +        // Act
561 +        // Assert
562 +    }
563 +
564 +    @Test
565 +    void testGetOrderRiskMetrics() throws Exception {
566 +        // Arrange
567 +        // Act
568 +        // Assert
569 +    }
570 +
571 +    @Test
572 +    void testUpdateOrderRiskMetrics() throws Exception {
573 +        // Arrange
574 +        // Act
575 +        // Assert
576 +    }
577 +
578 +    @Test
579 +    void testGetOrderFraudMetrics() throws Exception {
580 +        // Arrange
581 +        // Act
582 +        // Assert
583 +    }
584 +
585 +    @Test
586 +    void testUpdateOrderFraudMetrics() throws Exception {
587 +        // Arrange
588 +        // Act
589 +        // Assert
590 +    }
591 +
592 +    @Test
593 +    void testGetOrderComplianceAnalytics() throws Exception {
594 +        // Arrange
595 +        // Act
596 +        // Assert
597 +    }
598 +
599 +    @Test
600 +    void testUpdateOrderComplianceAnalytics() throws Exception {
601 +        // Arrange
602 +        // Act
603 +        // Assert
604 +    }
605 +
606 +    @Test
607 +    void testGetOrderRiskAnalytics() throws Exception {
608 +        // Arrange
609 +        // Act
610 +        // Assert
611 +    }
612 +
613 +    @Test
614 +    void testUpdateOrderRiskAnalytics() throws Exception {
615 +        // Arrange
616 +        // Act
617 +        // Assert
618 +    }
619 +
620 +    @Test
621 +    void testGetOrderFraudAnalytics() throws Exception {
622 +        // Arrange
623 +        // Act
624 +        // Assert
625 +    }
626 +
627 +    @Test
628 +    void testUpdateOrderFraudAnalytics() throws Exception {
629 +        // Arrange
630 +        // Act
631 +        // Assert
632 +    }
633 +
634 +    @Test
635 +    void testGetOrderComplianceReports() throws Exception {
636 +        // Arrange
637 +        // Act
638 +        // Assert
639 +    }
640 +
641 +    @Test
642 +    void testUpdateOrderComplianceReports() throws Exception {
643 +        // Arrange
644 +        // Act
645 +        // Assert
646 +    }
647 +
648 +    @Test
649 +    void testGetOrderRiskReports() throws Exception {
650 +        // Arrange
651 +        // Act
652 +        // Assert
653 +    }
654 +
655 +    @Test
656 +    void testUpdateOrderRiskReports() throws Exception {
657 +        // Arrange
658 +        // Act
659 +        // Assert
660 +    }
661 +
662 +    @Test
663 +    void testGetOrderFraudReports() throws Exception {
664 +        // Arrange
665 +        // Act
666 +        // Assert
667 +    }
668 +
669 +    @Test
670 +    void testUpdateOrderFraudReports() throws Exception {
671 +        // Arrange
672 +        // Act
673 +        // Assert
674 +    }
675 +
676 +    @Test
677 +    void testGetOrderComplianceMetrics() throws Exception {
678 +        // Arrange
679 +        // Act
680 +        // Assert
681 +    }
682 +
683 +    @Test
684 +    void testUpdateOrderComplianceMetrics() throws Exception {
685 +        // Arrange
686 +        // Act
687 +        // Assert
688 +    }
689 +
690 +    @Test
691 +    void testGetOrderRiskMetrics() throws Exception {
692 +        // Arrange
693 +        // Act
694 +        // Assert
695 +    }
696 +
697 +    @Test
698 +    void testUpdateOrderRiskMetrics() throws Exception {
699 +        // Arrange
700 +        // Act
701 +        // Assert
702 +    }
703 +
704 +    @Test
705 +    void testGetOrderFraudMetrics() throws Exception {
706 +        // Arrange
707 +        // Act
708 +        // Assert
709 +    }
710 +
711 +    @Test
712 +    void testUpdateOrderFraudMetrics() throws Exception {
713 +        // Arrange
714 +        // Act
715 +        // Assert
716 +    }
717 +
718 +    @Test
719 +    void testGetOrderComplianceAnalytics() throws Exception {
720 +        // Arrange
721 +        // Act
722 +        // Assert
723 +    }
724 +
725 +    @Test
726 +    void testUpdateOrderComplianceAnalytics() throws Exception {
727 +        // Arrange
728 +        // Act
729 +        // Assert
730 +    }
731 +
732 +    @Test
733 +    void testGetOrderRiskAnalytics() throws Exception {
734 +        // Arrange
735 +        // Act
736 +        // Assert
737 +    }
738 +
739 +    @Test
740 +    void testUpdateOrderRiskAnalytics() throws Exception {
741 +        // Arrange
742 +        // Act
743 +        // Assert
744 +    }
745 +
746 +    @Test
747 +    void testGetOrderFraudAnalytics() throws Exception {
748 +        // Arrange
749 +        // Act
750 +        // Assert
751 +    }
752 +
753 +    @Test
754 +    void testUpdateOrderFraudAnalytics() throws Exception {
755 +        // Arrange
756 +        // Act
757 +        // Assert
758 +    }
759 +
760 +    @Test
761 +    void testGetOrderComplianceReports() throws Exception {
762 +        // Arrange
763 +        // Act
764 +        // Assert
765 +    }
766 +
767 +    @Test
768 +    void testUpdateOrderComplianceReports() throws Exception {
769 +        // Arrange
770 +        // Act
771 +        // Assert
772 +    }
773 +
774 +    @Test
775 +    void testGetOrderRiskReports() throws Exception {
776 +        // Arrange
777 +        // Act
778 +        // Assert
779 +    }
780 +
781 +    @Test
782 +    void testUpdateOrderRiskReports() throws Exception {
783 +        // Arrange
784 +        // Act
785 +        // Assert
786 +    }
787 +
788 +    @Test
789 +    void testGetOrderFraudReports() throws Exception {
790 +        // Arrange
791 +        // Act
792 +        // Assert
793 +    }
794 +
795 +    @Test
796 +    void testUpdateOrderFraudReports() throws Exception {
797 +        // Arrange
798 +        // Act
799 +        // Assert
800 +    }
801 +
802 +    @Test
803 +    void testGetOrderComplianceMetrics() throws Exception {
804 +        // Arrange
805 +        // Act
806 +        // Assert
807 +    }
808 +
809 +    @Test
810 +    void testUpdateOrderComplianceMetrics() throws Exception {
811 +        // Arrange
812 +        // Act
813 +        // Assert
814 +    }
815 +
816 +    @Test
817 +    void testGetOrderRiskMetrics() throws Exception {
818 +        // Arrange
819 +        // Act
820 +        // Assert
821 +    }
822 +
823 +    @Test
824 +    void testUpdateOrderRiskMetrics() throws Exception {
825 +        // Arrange
826 +        // Act
827 +        // Assert
828 +    }
829 +
830 +    @Test
831 +    void testGetOrderFraudMetrics() throws Exception {
832 +        // Arrange
833 +        // Act
834 +        // Assert
835 +    }
836 +
837 +    @Test
838 +    void testUpdateOrderFraudMetrics() throws Exception {
839 +        // Arrange
840 +        // Act
841 +        // Assert
842 +    }
843 +
844 +    @Test
845 +    void testGetOrderComplianceAnalytics() throws Exception {
846 +        // Arrange
847 +        // Act
848 +        // Assert
849 +    }
850 +
851 +    @Test
852 +    void testUpdateOrderComplianceAnalytics() throws Exception {
853 +        // Arrange
854 +        // Act
855 +        // Assert
856 +    }
857 +
858 +    @Test
859 +    void testGetOrderRiskAnalytics() throws Exception {
860 +        // Arrange
861 +        // Act
862 +        // Assert
863 +    }
864 +
865 +    @Test
866 +    void testUpdateOrderRiskAnalytics() throws Exception {
867 +        // Arrange
868 +        // Act
869 +        // Assert
870 +    }
871 +
872 +    @Test
873 +    void testGetOrderFraudAnalytics() throws Exception {
874 +        // Arrange
875 +        // Act
876 +        // Assert
877 +    }
878 +
879 +    @Test
880 +    void testUpdateOrderFraudAnalytics() throws Exception {
881 +        // Arrange
882 +        // Act
883 +        // Assert
884 +    }
885 +
886 +    @Test
887 +    void testGetOrderComplianceReports() throws Exception {
888 +        // Arrange
889 +        // Act
890 +        // Assert
891 +    }
892 +
893 +    @Test
894 +    void testUpdateOrderComplianceReports() throws Exception {
895 +        // Arrange
896 +        // Act
897 +        // Assert
898 +    }
899 +
900 +    @Test
901 +    void testGetOrderRiskReports() throws Exception {
902 +        // Arrange
903 +        // Act
904 +        // Assert
905 +    }
906 +
907 +    @Test
908 +    void testUpdateOrderRiskReports() throws Exception {
909 +        // Arrange
910 +        // Act
911 +        // Assert
912 +    }
913 +
914 +    @Test
915 +    void testGetOrderFraudReports() throws Exception {
916 +        // Arrange
917 +        // Act
918 +        // Assert
919 +    }
920 +
921 +    @Test
922 +    void testUpdateOrderFraudReports() throws Exception {
923 +        // Arrange
924 +        // Act
925 +        // Assert
926 +    }
927 +
928 +    @Test
929 +    void testGetOrderComplianceMetrics() throws Exception {
930 +        // Arrange
931 +        // Act
932 +        // Assert
933 +    }
934 +
935 +    @Test
936 +    void testUpdateOrderComplianceMetrics() throws Exception {
937 +        // Arrange
938 +        // Act
939 +        // Assert
940 +    }
941 +
942 +    @Test
943 +    void testGetOrderRiskMetrics() throws Exception {
944 +        // Arrange
945 +        // Act
946 +        // Assert
947 +    }
948 +
949 +    @Test
950 +    void testUpdateOrderRiskMetrics() throws Exception {
951 +        // Arrange
952 +        // Act
953 +        // Assert
954 +    }
955 +
956 +    @Test
957 +    void testGetOrderFraudMetrics() throws Exception {
958 +        // Arrange
959 +        // Act
960 +        // Assert
961 +    }
962 +
963 +    @Test
964 +    void testUpdateOrderFraudMetrics() throws Exception {
965 +        // Arrange
966 +        // Act
967 +        // Assert
968 +    }
969 +
970 +    @Test
971 +    void testGetOrderComplianceAnalytics() throws Exception {
972 +        // Arrange
973 +        // Act
974 +        // Assert
975 +    }
976 +
977 +    @Test
978 +    void testUpdateOrderComplianceAnalytics() throws Exception {
979 +        // Arrange
980 +        // Act
981 +        // Assert
982 +    }
983 +
984 +    @Test
985 +    void testGetOrderRiskAnalytics() throws Exception {
986 +        // Arrange
987 +        // Act
988 +        // Assert
989 +    }
990 +
991 +    @Test
992 +    void testUpdateOrderRiskAnalytics() throws Exception {
993 +        // Arrange
994 +        // Act
995 +        // Assert
996 +    }
997 +
998 +    @Test
999 +    void testGetOrderFraudAnalytics() throws Exception {
1000 +        // Arrange
1001 +        // Act
1002 +        // Assert
1003 +    }
1004 +
1005 +    @Test
1006 +    void testUpdateOrderFraudAnalytics() throws Exception {
1007 +        // Arrange
1008 +        // Act
1009 +        // Assert
1010 +    }
1011 +
1012 +    @Test
1013 +    void testGetOrderComplianceReports() throws Exception {
1014 +        // Arrange
1015 +        // Act
1016 +        // Assert
1017 +    }
1018 +
1019 +    @Test
1020 +    void testUpdateOrderComplianceReports() throws Exception {
1021 +        // Arrange
1022 +        // Act
1023 +        // Assert
1024 +    }
1025 +
1026 +    @Test
1027 +    void testGetOrderRiskReports() throws Exception {
1028 +        // Arrange
1029 +        // Act
1030 +        // Assert
1031 +    }
1032 +
1033 +    @Test
1034 +    void testUpdateOrderRiskReports() throws Exception {
1035 +        // Arrange
1036 +        // Act
1037 +        // Assert
1038 +    }
1039 +
1040 +    @Test
1041 +    void testGetOrderFraudReports() throws Exception {
1042 +        // Arrange
1043 +        // Act
1044 +        // Assert
1045 +    }
1046 +
1047 +    @Test
1048 +    void testUpdateOrderFraudReports() throws Exception {
1049 +        // Arrange
1050 +        // Act
1051 +        // Assert
1052 +    }
1053 +
1054 +    @Test
1055 +    void testGetOrderComplianceMetrics() throws Exception {
1056 +        // Arrange
1057 +        // Act
1058 +        // Assert
1059 +    }
1060 +
1061 +    @Test
1062 +    void testUpdateOrderComplianceMetrics() throws Exception {
1063 +        // Arrange
1064 +        // Act
1065 +        // Assert
1066 +    }
1067 +
1068 +    @Test
1069 +    void testGetOrderRiskMetrics() throws Exception {
1070 +        // Arrange
1071 +        // Act
1072 +        // Assert
1073 +    }
1074 +
1075 +    @Test
1076 +    void testUpdateOrderRiskMetrics() throws Exception {
1077 +        // Arrange
1078 +        // Act
1079 +        // Assert
1080 +    }
1081 +
1082 +    @Test
1083 +    void testGetOrderFraudMetrics() throws Exception {
1084 +        // Arrange
1085 +        // Act
1086 +        // Assert
1087 +    }
1088 +
1089 +    @Test
1090 +    void testUpdateOrderFraudMetrics() throws Exception {
1091 +        // Arrange
1092 +        // Act
1093 +        // Assert
1094 +    }
1095 +
1096 +    @Test
1097 +    void testGetOrderComplianceAnalytics() throws Exception {
1098 +        // Arrange
1099 +        // Act
1100 +        // Assert
1101 +    }
1102 +
1103 +    @Test
1104 +    void testUpdateOrderComplianceAnalytics() throws Exception {
1105 +        // Arrange
1106 +        // Act
1107 +        // Assert
1108 +    }
1109 +
1110 +    @Test
1111 +    void testGetOrderRiskAnalytics() throws Exception {
1112 +        // Arrange
1113 +        // Act
1114 +        // Assert
1115 +    }
1116 +
1117 +    @Test
1118 +    void testUpdateOrderRiskAnalytics() throws Exception {
1119 +        // Arrange
1120 +        // Act
1121 +        // Assert
1122 +    }
1123 +
1124 +    @Test
1125 +    void testGetOrderFraudAnalytics() throws Exception {
1126 +        // Arrange
1127 +        // Act
1128 +        // Assert
1129 +    }
1130 +
1131 +    @Test
1132 +    void testUpdateOrderFraudAnalytics() throws Exception {
1133 +        // Arrange
1134 +        // Act
1135 +        // Assert
1136 +    }
1137 +
1138 +    @Test
1139 +    void testGetOrderComplianceReports() throws Exception {
1140 +        // Arrange
1141 +        // Act
1142 +        // Assert
1143 +    }
1144 +
1145 +    @Test
1146 +    void testUpdateOrderComplianceReports() throws Exception {
1147 +        // Arrange
1148 +        // Act
1149 +        // Assert
1150 +    }
1151 +
1152 +    @Test
1153 +    void testGetOrderRiskReports() throws Exception {
1154 +        // Arrange
1155 +        // Act
1156 +        // Assert
1157 +    }
1158 +
1159 +    @Test
1160 +    void testUpdateOrderRiskReports() throws Exception {
1161 +        // Arrange
1162 +        // Act
1163 +        // Assert
1164 +    }
1165 +
1166 +    @Test
1167 +    void testGetOrderFraudReports() throws Exception {
1168 +        // Arrange
1169 +        // Act
1170 +        // Assert
1171 +    }
1172 +
1173 +    @Test
1174 +    void testUpdateOrderFraudReports() throws Exception {
1175 +        // Arrange
1176 +        // Act
1177 +        // Assert
1178 +    }
1179 +
1180 +    @Test
1181 +    void testGetOrderComplianceMetrics() throws Exception {
1182 +        // Arrange
1183 +        // Act
1184 +        // Assert
1185 +    }
1186 +
1187 +    @Test
1188 +    void testUpdateOrderComplianceMetrics() throws Exception {
1189 +        // Arrange
1190 +        // Act
1191 +        // Assert
1192 +    }
1193 +
1194 +    @Test
1195 +    void testGetOrderRiskMetrics() throws Exception {
1196 +        // Arrange
1197 +        // Act
1198 +        // Assert
1199 +    }
1200 +
1201 +    @Test
1202 +    void testUpdateOrderRiskMetrics() throws Exception {
1203 +        // Arrange
1204 +        // Act
1205 +        // Assert
1206 +    }
1207 +
1208 +    @Test
1209 +    void testGetOrderFraudMetrics() throws Exception {
1210 +        // Arrange
1211 +        // Act
1212 +        // Assert
1213 +    }
1214 +
1215 +    @Test
1216 +    void testUpdateOrderFraudMetrics() throws Exception {
1217 +        // Arrange
1218 +        // Act
1219 +        // Assert
1220 +    }
1221 +
1222 +    @Test
1223 +    void testGetOrderComplianceAnalytics() throws Exception {
1224 +        // Arrange
1225 +        // Act
1226 +        // Assert
1227 +    }
1228 +
1229 +    @Test
1230 +    void testUpdateOrderComplianceAnalytics() throws Exception {
1231 +        // Arrange
1232 +        // Act
1233 +        // Assert
1234 +    }
1235 +
1236 +    @Test
1237 +    void testGetOrderRiskAnalytics() throws Exception {
1238 +        // Arrange
1239 +        // Act
1240 +        // Assert
1241 +    }
1242 +
1243 +    @Test
1244 +    void testUpdateOrderRiskAnalytics() throws Exception {
1245 +        // Arrange
1246 +        // Act
1247 +        // Assert
1248 +    }
1249 +
1250 +    @Test
1251 +    void testGetOrderFraudAnalytics() throws Exception {
1252 +        // Arrange
1253 +        // Act
1254 +        // Assert
1255 +    }
1256 +
1257 +    @Test
1258 +    void testUpdateOrderFraudAnalytics() throws Exception {
1259 +        // Arrange
1260 +        // Act
1261 +        // Assert
1262 +    }
1263 +
1264 +    @Test
1265 +    void testGetOrderComplianceReports() throws Exception {
1266 +        // Arrange
1267 +        // Act
1268 +        // Assert
1269 +    }
1270 +
1271 +    @Test
1272 +    void testUpdateOrderComplianceReports() throws Exception {
1273 +        // Arrange
1274 +        // Act
1275 +        // Assert
1276 +    }
1277 +
1278 +    @Test
1279 +    void testGetOrderRiskReports() throws Exception {
1280 +        // Arrange
1281 +        // Act
1282 +        // Assert
1283 +    }
1284 +
1285 +    @Test
1286 +    void testUpdateOrderRiskReports() throws Exception {
1287 +        // Arrange
1288 +        // Act
1289 +        // Assert
1290 +    }
1291 +
1292 +    @Test
1293 +    void testGetOrderFraudReports() throws Exception {
1294 +        // Arrange
1295 +        // Act
1296 +        // Assert
1297 +    }
1298 +
1299 +    @Test
1300 +    void testUpdateOrderFraudReports() throws Exception {
1301 +        // Arrange
1302 +        // Act
1303 +        // Assert
1304 +    }
1305 +
1306 +    @Test
1307 +    void testGetOrderComplianceMetrics() throws Exception
```

```

51 +         sampleOrder.setOperator(Operator.TELENET);
52 +         sampleOrder.setOrderType(OrderType.PORT_OUT);
53 +         sampleOrder.setPortingStatus(PortingStatus.CLOSED);
54 +         sampleOrder.setCreationDate(ZonedDateTime.now());
55 +         sampleOrder.setModificationDate(ZonedDateTime.now());
56     }
57 +
58     @Test
59     void testGetOrders() throws Exception {
60         String portingRequestId = "npRequestId123";
61         when(orderService.getOrders(portingRequestId)).thenReturn(sampleOrder);
62 +
63         mockMvc.perform(get("/v1/orders")
64             .param("portingRequestId", portingRequestId)
65             .contentType(APPLICATION_JSON))
66             .andExpect(status().isOk())
67             .andExpect(jsonPath("$.partitionKey").value("partitionKey123"))
68             .andExpect(jsonPath("$.sortKey").value("sortKey123"))
69             .andExpect(jsonPath("$.npRequestId").value("npRequestId123"))
70             .andExpect(jsonPath("$.customerNumber").value("customer123"))
71             .andExpect(jsonPath("$.recipientCode").value("recipient123"))
72             .andExpect(jsonPath("$.donorCode").value("donor123"));
73 +
74         verify(orderService, times(1)).getOrders(portingRequestId);
75     }
76
77     @Test
78     void testSearchOrders() throws Exception {
79         OrderSearchCriteria criteria = new OrderSearchCriteria();
80         criteria.setCustomerNumber("customer123");
81         criteria.setMsisdn("0123456789");
82         criteria.setNprNumber("9876543210");
83         criteria.setOperator(Operator.TELENET);
84         criteria.setOrderType(OrderType.PORT_OUT);
85         criteria.setOrderStatus(PortingStatus.CLOSED);
86 +
87         List<BffPortingRequestOrder> orders = Collections.singletonList(sampleOrder);
88 +
89         when(orderService.searchOrders(Mockito.any(OrderSearchCriteria.class))).thenReturn(orders);
90 +
91         mockMvc.perform(post("/v1/searchorders")
92             .contentType(APPLICATION_JSON)
93             .content(objectMapper.writeValueAsString(criteria)))
94             .andExpect(status().isOk())
95             .andExpect(jsonPath("[0].customerNumber").value("customer123"))
96             .andExpect(jsonPath("[0].msisdn").value("0123456789"))
97             .andExpect(jsonPath("[0].nprNumber").value("9876543210"))
98             .andExpect(jsonPath("[0].operator").value("TELENET"))
99             .andExpect(jsonPath("[0].orderType").value("PORT_OUT"));
100 +
101         verify(orderService, times(1)).searchOrders(Mockito.any(OrderSearchCriteria.class));
102     }

```

✓ src/test/java/be/telenet/cnpservice/handler/MetricHandlerTest.java 0 → +69 -0 Viewed ⚡

```

1  + package be.telenet.cnpservice.handler;
2  +
3  + import static org.junit.jupiter.api.Assertions.*;
4  + import static org.mockito.Mockito.*;
5  +
6  + import lombok.SneakyThrows;
7  + import org.junit.jupiter.api.BeforeEach;
8  + import org.junit.jupiter.api.Test;
9  + import org.mockito.ArgumentCaptor;
10 + import org.mockito.Captor;
11 + import org.mockito.Mock;
12 + import org.mockito.MockitoAnnotations;
13 + import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
14 + import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
15 + import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
16 +
17 + class MetricHandlerTest {
18 +
19 +     @Mock
20 +     private CloudWatchClient cloudWatchClient;
21 +
22 +     private MetricHandler metricHandler;
23 +
24 +     @Captor
25 +     private ArgumentCaptor<PutMetricDataRequest> requestCaptor;
26 +
27 +     @BeforeEach
28 +     void setUp() {
29 +         MockitoAnnotations.openMocks(this);
30 +         metricHandler = new MetricHandler(cloudWatchClient, "test-namespace");
31 +
32 +         when(cloudWatchClient.serviceName()).thenReturn("mockCloudWatch");
33 +     }
34 +
35 +     @Test
36 +     @SneakyThrows
37 +     void testPublishMetricValidInput() {
38 +         String metricName = "TestMetric";
39 +         String dimensionName = "DimensionName";
40 +         String dimensionValue = "DimensionValue";
41 +
42 +         metricHandler.publishMetric(metricName, dimensionName, dimensionValue);
43 +
44 +         verify(cloudWatchClient, times(1)).putMetricData(requestCaptor.capture());
45 +
46 +         PutMetricDataRequest request = requestCaptor.getValue();
47 +
48 +         assertEquals("test-namespace", request.namespace()); // Match the namespace from the
49 +         // constructor
50 +         assertEquals(1, request.metricData().size());
51 +         MetricDatum datum = request.metricData().get(0);

```

```

51 +         MetricDatum datum = request.metricData().get(0);
52 +         assertEquals("TestMetric", datum.metricName());
53 +         assertEquals(1.0, datum.value());
54 +         assertEquals("DimensionName", datum.dimensions().get(0).name());
55 +         assertEquals("DimensionValue", datum.dimensions().get(0).value());
56 +
57 +
58 +     @Test
59 +     @SneakyThrows
60 +     void testPublishMetricNullMetricName() {
61 +         String metricName = null;
62 +         String dimensionName = "DimensionName";
63 +         String dimensionValue = "DimensionValue";
64 +
65 +         metricHandler.publishMetric(metricName, dimensionName, dimensionValue);
66 +
67 +         verify(cloudWatchClient, times(0)).putMetricData((PutMetricDataRequest) any());
68 +     }
69 + }

```

src/test/java/be/telenet/cnpservice/mapper/PortingMessageDTOMapperTest.java

+65 -12 Viewed

↑	@@ -37,7 +37,6 @@ import be.telenet.cnpservice.repository.model.TaskBean;
37 37	import com.fasterxml.jackson.core.JsonProcessingException;
38 38	import com.fasterxml.jackson.databind.ObjectMapper;
39 39	import java.time.ZonedDateTime;
40 40	- import java.util.Arrays;
41 41	import java.util.Collections;
42 42	import java.util.List;
43 43	import java.util.UUID;
↓	@@ -51,9 +50,6 @@ import software.amazon.awssdk.enhanced.dynamodb.model.QueryEnhancedRequest;
51 50	@ExtendWith(MockitoExtension.class)
52 51	class PortingMessageDTOMapperTest {
53 52	
54 53	- public static final String PARTISION_KEY = "PORT_IN#9f0c160b-143c-47db-9fb0-d765321e1bed";
55 54	- public static final String SORT_KEY = "REQUEST#9f0c160b-143c-47db-9fb0-d765321e1bed";
56 55	- public static final String MESSAGE_TYPE = "NPR";
57 53	private static final String MSISDN = "0468352656";
58 54	private static final String NP_REQ_ID = "95d9d94e-3dcc-47b6-b39d-86bcad03b335";
59 55	private static final String NPR_NO = "BASE202023072716591606";
↓	@@ -68,14 +64,13 @@ class PortingMessageDTOMapperTest {
68 64	private Step step;
69 65	private PortingRequestBean portInInfo;
70 66	private PortingRequestBean portInStep;
71 67	- private PortingDetails portingDetails;
72 68	private Task task;

```

74   69     @BeforeEach
75   70     public void setup() {
76   71
77   72         portingMessageDTOMapper = new PortingMessageDTOMapper();
78 - 73     portingDetails = PortingDetails.builder()
79 + 73     PortingDetails portingDetails = PortingDetails.builder()
80
81   76         .msisdns(Collections.singletonList(MSISDN))
82         .nprNumber(NPR_NO)
83         .serviceClass(ServiceClass.PRE_PAID)

    ↓
↑    @@ -341,9 +336,7 @@ class PortingMessageDTOMapperTest {

341 336         .reasons("invalid json")
342 337         .build();

344 - assertThrows(CnpServiceException.class, () -> {
345 -     portingMessageDTOMapper.buildTaskFromTaskBean(taskBean);
346 - });
339 + assertThrows(CnpServiceException.class, () ->
340         portingMessageDTOMapper.buildTaskFromTaskBean(taskBean));

347 340     }

348 341     @Test

    ↓
↑    @@ -374,14 +367,74 @@ class PortingMessageDTOMapperTest {

374 367         .serviceClass(ServiceClass.PRE_PAID)
375 368         .orderType(OrderType.PORT_IN)
376 369         .portingStatus(PortingStatus.CLOSED)
377 - msisdn(Arrays.asList("0462005047"))
378 + msisdn(List.of("0462005047"))
379 - targetSystems(Arrays.asList(Systems.CRDC_BASE))
380 + targetSystems(List.of(Systems.CRDC_BASE))
381         .donorCode("BEMO")
382         .recipientCode("BASE")
383         .bssSystem("NETCRACKER")
384         .build();

384 - List<Operator> expectedOperators = Arrays.asList(Operator.BASE_RETAIL);
377 + List<Operator> expectedOperators = List.of(Operator.BASE_RETAIL);
378 + List<Operator> actualOperators =
379     portingMessageDTOMapper.findOperator(bffPortingRequestOrder);
380     assertEquals(expectedOperators, actualOperators);
381 }

382     @Test
383     void findOperatorPortOutTest() {
384         BffPortingRequestOrder bffPortingRequestOrder = BffPortingRequestOrder.builder()
385             .npRequestId("057d7e9c-bc6b-49a1-9b0c-3dfb301e3934")
386             .initiatedBy(Systems.NC_BSS)
387             .serviceClass(ServiceClass.PRE_PAID)
388             .orderType(OrderType.PORT_OUT)
389             .portingStatus(PortingStatus.CLOSED)
390             .msisdn(List.of("0462005047"))
391             .nprNumber("BASE202024090705040127")

```

```

392 +         .targetSystems(List.of(Systems.CRDC_BASE))
393 +         .donorCode("BEMO")
394 +         .recipientCode("BASE")
395 +         .bssSystem("NETCRACKER")
396 +         .build();
397 +     List<Operator> expectedOperators = List.of(Operator.UNKNOWN);
398 +     List<Operator> actualOperators =
399 +       portingMessageDTOMapper.findOperator(bffPortingRequestOrder);
400 +     assertEquals(expectedOperators, actualOperators);
401 +
402 +   @Test
403 +   void findOperatorDisconnectInTest() {
404 +     BffPortingRequestOrder bffPortingRequestOrder = BffPortingRequestOrder.builder()
405 +       .npRequestId("057d7e9c-bc6b-49a1-9b0c-3dfb301e3934")
406 +       .initiatedBy(Systems.NC_BSS)
407 +       .serviceClass(ServiceClass.PRE_PAID)
408 +       .orderType(OrderType.DISCONNECT_IN)
409 +       .portingStatus(PortingStatus.CLOSED)
410 +       .msisdn(List.of("0462005047"))
411 +       .nprNumber("BASE202024090705040127")
412 +       .targetSystems(List.of(Systems.CRDC_BASE))
413 +       .donorCode("BEMO")
414 +       .recipientCode("BASE")
415 +       .bssSystem("NETCRACKER")
416 +       .build();
417 +     List<Operator> expectedOperators = List.of();
418 +     List<Operator> actualOperators =
419 +       portingMessageDTOMapper.findOperator(bffPortingRequestOrder);
420 +     assertEquals(expectedOperators, actualOperators);
421 +
422 +   @Test
423 +   void findOperatorDisconnectOutTest() {
424 +     BffPortingRequestOrder bffPortingRequestOrder = BffPortingRequestOrder.builder()
425 +       .npRequestId("057d7e9c-bc6b-49a1-9b0c-3dfb301e3934")
426 +       .initiatedBy(Systems.NC_BSS)
427 +       .serviceClass(ServiceClass.PRE_PAID)
428 +       .orderType(OrderType.DISCONNECT_OUT)
429 +       .portingStatus(PortingStatus.CLOSED)
430 +       .msisdn(List.of("0462005047"))
431 +       .nprNumber("BASE202024090705040127")
432 +       .targetSystems(List.of(Systems.CRDC_BASE))
433 +       .donorCode("BEMO")
434 +       .recipientCode("BASE")
435 +       .bssSystem("NETCRACKER")
436 +       .build();
437 +     List<Operator> expectedOperators = List.of(Operator.BASE_RETAIL);
385 438     List<Operator> actualOperators =
386 439     portingMessageDTOMapper.findOperator(bffPortingRequestOrder);
387 440     assertEquals(expectedOperators, actualOperators);
}

```

src/test/java/be/telenet/cnpservice/repository/MsgOutToCrdcRepositoryTest.java +44 -0 Viewed

```
@@ -10,8 +10,11 @@ import static org.mockito.Mockito.when;
10 10     import be.telenet.cnpservice.model.MessageType;
11 11     import be.telenet.cnpservice.repository.model.MsgOutToCrdcBean;
12 12     import io.awspring.cloud.dynamodb.DynamoDbOperations;
13 + import java.time.DayOfWeek;
14 14     import java.time.Instant;
15 + import java.time.ZoneId;
16 16     import java.time.ZonedDateTime;
17 + import java.time.temporal.TemporalAdjusters;
18 18     import java.util.List;
19 19     import java.util.stream.Stream;
20 20     import org.junit.jupiter.api.BeforeEach;
```

@@ -117,4 +120,45 @@ class MsgOutToCrdcRepositoryTest {

```
117 120             .limit(limit)
118 121             .build();
119 122         }
123 +
124 +     @Test
125 +     void updateMsgOutToCrdcWithVisibilityTimeOut() {
126 +         String partitionKey = "samplePartitionKey";
127 +         int visibilityTimeOut = 3600;
128 +
129 +         given(dynamoDbOperations.load(Key.builder().partitionValue(partitionKey).build(),
130 +             MsgOutToCrdcBean.class))
131 +             .willReturn(msgOutToCrdcBean);
132 +
133 +         ZonedDateTime originalNextVisibilityTime = msgOutToCrdcBean.getNextVisibilityTime();
134 +
135 +         msgOutToCrdcRepository.updateMsgOutToCrdc(partitionKey, visibilityTimeOut);
136 +
137 +         verify(dynamoDbOperations).save(msgOutToCrdcBean);
138 +
139 +         assertThat(msgOutToCrdcBean.getTryCount()).isEqualTo(1);
140 +
141 +         ZonedDateTime expectedNextVisibilityTime =
142 +             originalNextVisibilityTime.plusSeconds(visibilityTimeOut);
143 +
144 +
145 +         assertThat(msgOutToCrdcBean.getNextVisibilityTime()).isEqualTo(expectedNextVisibilityTime);
146     }
147 +
148 +     @Test
149 +     void updateMsgOutToCrdcToNextMonday() {
150 +         String partitionKey = "samplePartitionKey";
151 +
152 +         given(dynamoDbOperations.load(Key.builder().partitionValue(partitionKey).build(),
153 +             MsgOutToCrdcBean.class))
154 +             .willReturn(msgOutToCrdcBean);
155 +
156 +         msgOutToCrdcRepository.updateMsgOutToCrdc(partitionKey);
157 +
158 +         verify(dynamoDbOperations).save(msgOutToCrdcBean);
159 +
160 +         ZonedDateTime expectedNextMonday = ZonedDateTime.now(ZoneId.of("Europe/Brussels"))
161 +             .with(TemporalAdjusters.next(DayOfWeek.MONDAY))
162 +             .withHour(6)
163 +             .withMinute(0)
164 +             .withSecond(0)
165 +             .withNano(0);
166 +
167 +         assertThat(msgOutToCrdcBean.getNextVisibilityTime()).isEqualTo(expectedNextMonday);
168     }
```

src/test/java/be/telenet/cnpservice/repository/PaginatedTaskRepositoryTest.java +42 -0 Viewed

```
0 → 100644
```

```
1 + package be.telenet.cnpservice.repository;
2 +
3 + import static org.junit.jupiter.api.Assertions.*;
4 + import static org.mockito.Mockito.*;
5 +
6 + import be.telenet.cnpservice.exception.TaskQueryException;
7 + import be.telenet.cnpservice.repository.model.TaskBean;
8 + import io.awsspring.cloud.dynamodb.DynamoDbOperations;
9 + import org.junit.jupiter.api.BeforeEach;
10 + import org.junit.jupiter.api.Test;
11 + import org.mockito.InjectMocks;
12 + import org.mockito.Mock;
13 + import org.mockito.MockitoAnnotations;
14 + import software.amazon.awssdk.enhanced.dynamodb.model.QueryEnhancedRequest;
15 +
16 + class PaginatedTaskRepositoryTest {
17 +
18 +     @Mock
19 +     private DynamoDbOperations dynamoDbOperations;
20 +
21 +     @InjectMocks
22 +     private PaginatedTaskRepository paginatedTaskRepository;
23 +
24 +     @BeforeEach
25 +     void setUp() {
26 +         MockitoAnnotations.openMocks(this);
27 +     }
28 +
29 +     @Test
30 +     void testGetTasksPaginated_ThrowsException() {
31 +         QueryEnhancedRequest mockRequest = mock(QueryEnhancedRequest.class);
32 +         String mockIndex = "mock-index";
33 +
34 +         when(dynamoDbOperations.query(mockRequest, TaskBean.class, mockIndex))
35 +             .thenThrow(new RuntimeException("DynamoDB query error"));
36 +
37 +         TaskQueryException exception = assertThrows(
38 +             TaskQueryException.class, () ->
39 +             paginatedTaskRepository.getTasksPaginated(mockRequest, mockIndex));
40 +         assertEquals("Failed to execute paginated query for tasks", exception.getMessage());
41 +         verify(dynamoDbOperations, times(1)).query(mockRequest, TaskBean.class, mockIndex);
42 +     }
43 }
```

src/test/java/be/telenet/cnpservice/service/MetricServiceTest.java 0 → +94 -0 Viewed

```

1 + package be.telenet.cnpservice.service;
2 +
3 + import static org.mockito.BDDMockito.given;
4 + import static org.mockito.Mockito.verify;
5 +
6 + import be.telenet.cnp.data.model.BffPortingRequestOrder;
7 + import be.telenet.cnp.data.model.Operator;
8 + import be.telenet.cnp.data.model.OrderType;
9 + import be.telenet.cnp.data.model.PortingStatus;
10 + import be.telenet.cnpservice.handler.MetricHandler;
11 + import be.telenet.cnpservice.repository.model.PortingRequestBean;
12 + import lombok.SneakyThrows;
13 + import org.junit.jupiter.api.BeforeEach;
14 + import org.junit.jupiter.api.Test;
15 + import org.junit.jupiter.api.extension.ExtendWith;
16 + import org.mockito.InjectMocks;
17 + import org.mockito.Mock;
18 + import org.mockito.junit.jupiter.MockitoExtension;
19 +
20 + @ExtendWith(MockitoExtension.class)
21 + class MetricServiceTest {
22 +
23 +     @Mock
24 +     private OrderService orderService;
25 +
26 +     @Mock
27 +     private MetricHandler metricHandler;
28 +
29 +     @InjectMocks
30 +     private MetricService metricService;
31 +
32 +     private PortingRequestBean portingRequestBean;
33 +
34 +     @BeforeEach
35 +     void setUp() {
36 +         portingRequestBean = PortingRequestBean.builder()
37 +             .npRequestId("057d7e9c-bc6b-49a1-90c-3dfb301e3934")
38 +             .orderType(OrderType.PORT_IN)
39 +             .portingStatus(PortingStatus.OPEN)
40 +             .serviceClass(null)
41 +             .recipientCode("12345")
42 +             .isSuccessfulOrder(true)
43 +             .donorCode("Donor")
44 +             .build();
45 +     }
46 +
47 +     @Test
48 +     @SneakyThrows
49 +     void createRequestMetricsTest() {
50 +         metricService.createRequestMetrics(portingRequestBean);
51 +     }

```

```

52 |         verify(metricHandler).publishMetric("INCOMINGREQUEST_CREATION", "RequestsByOrderType",
53 |             "PORT_IN");
54 |         verify(metricHandler).publishMetric("INCOMINGREQUEST_CREATION_PORT_IN",
55 |             "RequestsByRecipient", "12345");
56 |     }
57 |
58 |     @Test
59 |     @SneakyThrows
60 |     void createRequestMetricsWithServiceClassTest() {
61 |         portingRequestBean.setServiceClass(be.telenet.cnp.data.model.ServiceClass.PRE_PAID);
62 |
63 |         metricService.createRequestMetrics(portingRequestBean);
64 |
65 |         verify(metricHandler).publishMetric("INCOMINGREQUEST_CREATION", "RequestsByOrderType",
66 |             "PORT_IN");
67 |         verify(metricHandler).publishMetric("INCOMINGREQUEST_CREATION_PORT_IN",
68 |             "RequestsByRecipient", "12345");
69 |         verify(metricHandler).publishMetric("INCOMINGREQUEST_CREATION_PORT_IN",
70 |             "RequestsByServiceClass", "PRE_PAID");
71 |     }
72 |
73 |
74 |     @Test
75 |     @SneakyThrows
76 |     void createUpdatedRequestMetrics_SuccessfulOrderTest() {
77 |         given(orderService.getOrders(portingRequestBean.getNpRequestId()))
78 |             .willReturn(createMockBffPortingRequestOrder());
79 |
80 |         metricService.createUpdatedRequestMetrics(portingRequestBean);
81 |
82 |         verify(metricHandler).publishMetric("SUCCESSFUL_PORT_IN_ORDER_FOR_TELENET",
83 |             "SuccessFullOrders", "PORT_IN");
84 |
85 |
86 |         @Test
87 |         @SneakyThrows
88 |         void createUpdatedRequestMetrics_FailedOrderTest() {
89 |             portingRequestBean.setIsSuccessfulOrder(false);
90 |             given(orderService.getOrders(portingRequestBean.getNpRequestId()))
91 |                 .willReturn(createMockBffPortingRequestOrder());
92 |
93 |             metricService.createUpdatedRequestMetrics(portingRequestBean);
94 |
95 |             verify(metricHandler).publishMetric("FAILED_ORDER_FROM_Donor_TO_TELENET", "FailedOrders",
96 |                 "PORT_IN");
97 |         }
98 |
99 |         private BffPortingRequestOrder createMockBffPortingRequestOrder() {
100 |             return BffPortingRequestOrder.builder().operator(Operator.TELENET).build();
101 |         }
102 |     }

```

✓ src/test/java/be/telenet/cnpservice/util/EnhancedLastEvaluatedKeySerializerTes
 t.java 0 → 100644

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import static org.junit.jupiter.api.Assertions.*;
4 +
5 + import java.io.UncheckedIOException;
6 + import java.util.Collections;
7 + import java.util.HashMap;
8 + import java.util.Map;
9 + import org.junit.jupiter.api.Test;
10 + import software.amazon.awssdk.services.dynamodb.model.AttributeValue;
11 +
12 + class EnhancedLastEvaluatedKeySerializerTest {
13 +
14 +     @Test
15 +     void testSerializeNullOrEmptyInput() {
16 +         assertNull(EnhancedLastEvaluatedKeySerializer.serialize(null));
17 +
18 +         assertNull(EnhancedLastEvaluatedKeySerializer.serialize(Collections.emptyMap()));
19 +     }
20 +
21 +     @Test
22 +     void testSerialize() {
23 +         Map<String, AttributeValue> attributeValueMap = new HashMap<>();
24 +         attributeValueMap.put("id", AttributeValue.builder().s("123").build());
25 +         attributeValueMap.put("name", AttributeValue.builder().s("test-name").build());
26 +
27 +         String result = EnhancedLastEvaluatedKeySerializer.serialize(attributeValueMap);
28 +
29 +         assertNotNull(result);
30 +         assertTrue(result.length() > 0);
31 +     }
32 +
33 +     @Test
34 +     void testDeserializeNullOrEmptyInput() {
35 +         assertEquals(Collections.emptyMap(),
36 +             EnhancedLastEvaluatedKeySerializer.deserialize(null));
37 +
38 +         assertEquals(Collections.emptyMap(), EnhancedLastEvaluatedKeySerializer.deserialize(""));
39 +     }
40 +
41 +     @Test
42 +     void testDeserialize() {
43 +         Map<String, AttributeValue> attributeValueMap = new HashMap<>();
44 +         attributeValueMap.put("id", AttributeValue.builder().s("123").build());
45 +         attributeValueMap.put("name", AttributeValue.builder().s("test-name").build());
46 +
47 +         String serialized = EnhancedLastEvaluatedKeySerializer.serialize(attributeValueMap);
48 +
49 +         Map<String, AttributeValue> deserializedMap =
50 +             EnhancedLastEvaluatedKeySerializer.deserialize(serialized);
51 +
52 +         assertNotNull(deserializedMap);
53 +         assertEquals(2, deserializedMap.size());
54 +         assertEquals("123", deserializedMap.get("id").s());
55 +         assertEquals("test-name", deserializedMap.get("name").s());
56 +
57 +     @Test
58 +     void testDeserializeInvalidBase64() {
59 +         String invalidBase64 = "invalidBase64String";
60 +
61 +         assertThrows(UncheckedIOException.class, () ->
62 +             EnhancedLastEvaluatedKeySerializer.deserialize(invalidBase64));
63 +     }
64 + }

```

✓ src/test/java/be/telenet/cnpservice/util/FilterUtilTest.java 0 → 100644 +24 -0 Viewed

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import static org.junit.jupiter.api.Assertions.assertThrows;
4 +
5 + import java.lang.reflect.Constructor;
6 + import java.lang.reflect.InvocationTargetException;
7 + import org.junit.jupiter.api.Test;
8 +
9 + class FilterUtilTest {
10 +     @Test
11 +     void testPrivateConstructor() {
12 +         Constructor<FilterUtil> constructor;
13 +         try {
14 +             constructor = FilterUtil.class.getDeclaredConstructor();
15 +             constructor.setAccessible(true);
16 +             assertThrows(
17 +                 InvocationTargetException.class,
18 +                 constructor::newInstance,
19 +                 "Expected an UnsupportedOperationException when invoking the constructor.");
20 +         } catch (NoSuchMethodException e) {
21 +             throw new AssertionError("The private constructor is missing.", e);
22 +         }
23 +     }
24 + }

```

✓ src/test/java/be/telenet/cnpservice/util/LastEvaluatedKeySerializerTest.java 0 → 100644 +54 -0 Viewed

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import static org.junit.jupiter.api.Assertions.*;
4 +
5 + import java.util.Collections;
6 + import java.util.HashMap;
7 + import java.util.Map;
8 + import org.junit.jupiter.api.Test;
9 + import software.amazon.awssdk.services.dynamodb.model.AttributeValue;
10 +
11 + class LastEvaluatedKeySerializerTest {
12 +
13 +     @Test
14 +     void testSerializeNullOrEmptyInput() {
15 +         assertNull(LastEvaluatedKeySerializer.serialize(null));
16 +
17 +         assertNull(LastEvaluatedKeySerializer.serialize(Collections.emptyMap()));
18 +     }
19 +
20 +     @Test
21 +     void testSerialize() {
22 +         Map<String, AttributeValue> attributeValueMap = new HashMap<>();
23 +         attributeValueMap.put("id", AttributeValue.builder().s("123").build());
24 +         attributeValueMap.put("name", AttributeValue.builder().s("test-name").build());
25 +
26 +         String result = LastEvaluatedKeySerializer.serialize(attributeValueMap);

```

```

27 +         assertNotNull(result);
28 +         assertTrue(result.contains("\"id\":\"123\""));
29 +         assertTrue(result.contains("\"name\":\"test-name\""));
30 +
31     }
32 +
33     @Test
34     void testDeserializeNullOrEmptyInput() {
35         assertEquals(Collections.emptyMap(), LastEvaluatedKeySerializer.deserialize(null));
36 +
37         assertEquals(Collections.emptyMap(), LastEvaluatedKeySerializer.deserialize(""));
38     }
39 +
40     @Test
41     void testDeserialize() {
42         Map<String, AttributeValue> attributeValueMap = new HashMap<>();
43         attributeValueMap.put("id", AttributeValue.builder().s("123").build());
44         attributeValueMap.put("name", AttributeValue.builder().s("test-name").build());
45         String serializedJson = LastEvaluatedKeySerializer.serialize(attributeValueMap);
46 +
47         Map<String, AttributeValue> deserializedMap =
48             LastEvaluatedKeySerializer.deserialize(serializedJson);
49 +
50         assertNotNull(deserializedMap);
51         assertEquals(2, deserializedMap.size());
52         assertEquals("123", deserializedMap.get("id").s());
53         assertEquals("test-name".s(), deserializedMap.get("name").s());

```

src/test/java/be/telenet/cnpservice/util/PortingMapperUtilTest.java 0 → +40 -0 Viewed

100644

```

1 + package be.telenet.cnpservice.util;
2 +
3 + import static org.junit.jupiter.api.Assertions.assertEquals;
4 + import static org.junit.jupiter.api.Assertions.assertThrows;
5 +
6 + import be.telenet.cnpservice.model.BffPortingRequestOrder;
7 + import be.telenet.cnpservice.model.OrderType;
8 + import be.telenet.cnpservice.model.PortingStatus;
9 + import java.lang.reflect.Constructor;
10 + import java.lang.reflect.InvocationTargetException;
11 + import org.junit.jupiter.api.Test;
12 +
13 class PortingMapperUtilTest {
14     @Test
15     void testPrivateConstructor() {
16         Constructor<PortingMapperUtil> constructor;
17         try {
18             constructor = PortingMapperUtil.class.getDeclaredConstructor();
19             constructor.setAccessible(true);
20             assertThrows(
21                 InvocationTargetException.class,
22                 constructor::newInstance,
23                 "Expected an UnsupportedOperationException when invoking the constructor.");
24         } catch (NoSuchMethodException e) {
25             throw new AssertionError("The private constructor is missing.", e);
26         }

```

```
29 | +     @Test
30 | +     void testOrderTypePortingStatus() {
31 | +         BffPortingRequestOrder order = new BffPortingRequestOrder();
32 | +         order.setOrderType(OrderType.PORT_IN);
33 | +         order.setPortingStatus(PortingStatus.CLOSED);
34 |
35 | +         String result = PortingMapperUtil.orderTypePortingStatus(order);
36 |
37 | +         String expected = "PORT_IN#CLOSED";
38 | +         assertEquals(expected, result);
39 | +     }
40 | + }
```

src/test/java/be/telenet/cnpservice/util/TimeProviderTest.java 0 → 100644

+24 -0 Viewed

```
1 | + package be.telenet.cnpservice.util;
2 |
3 | + import static org.junit.jupiter.api.Assertions.assertThrows;
4 |
5 | + import java.lang.reflect.Constructor;
6 | + import java.lang.reflect.InvocationTargetException;
7 | + import org.junit.jupiter.api.Test;
8 |
9 | + class TimeProviderTest {
10 | +     @Test
11 | +     void testPrivateConstructor() {
12 | +         Constructor<TimeProvider> constructor;
13 | +         try {
14 | +             constructor = TimeProvider.class.getDeclaredConstructor();
15 | +             constructor.setAccessible(true);
16 | +             assertThrows(
17 | +                 InvocationTargetException.class,
18 | +                 constructor::newInstance,
19 | +                 "Expected an UnsupportedOperationException when invoking the constructor.");
20 | +         } catch (NoSuchMethodException e) {
21 | +             throw new AssertionError("The private constructor is missing.", e);
22 | +         }
23 | +     }
24 | + }
```

Task Request Handler (coverage)

Showing 4 changed files ▾ with 120 additions and 21 deletions

Hide whitespace changes | Inline | Side-by-side

src/main/java/be/telenet/cnp/handler/TaskRequestHandler.java

```
@@ -13,7 +13,7 @@ import lombok.extern.slf4j.Slf4j;
@Slf4j
public class TaskRequestHandler extends AbstractSQSEventHandler {
    private TaskRequestService taskRequestService;
+    private final TaskRequestService taskRequestService;
    public TaskRequestHandler() {
        this(new TaskRequestService());
    }
}
```

src/main/java/be/telenet/cnp/mapper/TaskMapper.java

```
+59 -20 @@ public class TaskMapper {
    ...
    public static Task setAdditionalFields(Step currentStep, Task task, PortingRequest portingRequest) {
        if (null != task.getStepName() && task.getStepName().equalsIgnoreCase("AUTHORIZE_MANUAL_FROM_NC")) {
            log.info("currentStep is {}", currentStep);
            task.setMsisdns(currentStep.getMsisdns());
            task.setDescription(
                "PortOut order could not be created in BSS due to open order /other issues. manual check is required ");
        }
        if (isAuthorizeManualStep(task)) {
            handleAuthorizeManualStep(currentStep, task);
        }
        if (null != currentStep.getValidation()
            && null != currentStep.getValidation().getType()) {
            task.setType(valueOf(currentStep.getValidation().getType()));
            task.setDescription(currentStep.getValidation().getDescription());
        } else if (NP_REJECT == currentStep.getMessageType() || NP_EXEC_REJECT == currentStep.getMessageType()) {
            setRejectDetails(currentStep, task, portingRequest);
        } else if (NACK == task.getType()) {
            String stepName = task.getStepName();
            task.setDescription("Received NACK while trying to send " + stepName.substring(0, stepName.indexOf("_"))
                + " to " + stepName.substring(stepName.lastIndexOf("_") + 1));
        } else if (!currentStep.getReason().isEmpty()
            && !currentStep.getStepName().contains("FEEDBACK_FROM_CRDC")) {
            String description = currentStep.getReason().get(0).getDescription();
        }
    }
}
```

```

103      -         task.setType(description.contains("timed out") || description.contains("Timeout") ? TIME_OUT : EXCEPTION);
104      -         task.setDescription(description);
105      +     if (isValidStep(currentStep)) {
106      +         handleValidationStep(currentStep, task);
107      +     } else if (isRejectStep(currentStep)) {
108      +         handleRejectDetails(currentStep, task, portingRequest);
109      +     } else if (isNackStep(task)) {
110      +         handleNackStep(task);
111      +     } else if (isTimeoutOrExceptionStep(currentStep)) {
112      +         handleTimeoutOrExceptionStep(currentStep, task);
113      +
114      95     }
115      96     return task;
116      97   }
117      98
118      99   private static boolean isAuthorizeManualStep(Task task) {
119      100      return task.getStepName() != null & task.getStepName().equalsIgnoreCase("AUTHORIZE_MANUAL_FROM_NC");
120      101    }
121      102    +
122      103    private static void handleAuthorizeManualStep(Step currentStep, Task task) {
123      104      log.info("currentStep is {}", currentStep);
124      105      task.setSisdns(currentStep.getMsisdns());
125      106      task.setDescription(
126      107          "PortOut order could not be created in BSS due to open order /other issues. manual check is required ");
127      108    }
128      109    +
129      110    private static boolean isValidationStep(Step currentStep) {
130      111      return currentStep.getValidation() != null
131      112          && currentStep.getValidation().getType() != null;
132      113    }
133      114    +
134      115    private static void handleValidationStep(Step currentStep, Task task) {
135      116      task.setType(valueOf(currentStep.getValidation().getType()));
136      117      task.setDescription(currentStep.getValidation().getDescription());
137      118    }
138      119    +
139      120    private static boolean isRejectStep(Step currentStep) {
140      121      return NP_REJECT == currentStep.getMessageType() || NP_EXEC_REJECT == currentStep.getMessageType();
141      122    }
142      123    +
143      124    private static void handleRejectDetails(Step currentStep, Task task, PortingRequest portingRequest) {
144      125      setRejectDetails(currentStep, task, portingRequest);
145      126    }
146      127    +
147      128    private static boolean isNackStep(Task task) {
148      129      return NACK == task.getType();
149      130    }
150      131    +
151      132    private static void handleNackStep(Task task) {
152      133      String stepName = task.getStepName();
153      134      task.setDescription("Received NACK while trying to send " + stepName.substring(0, stepName.indexOf("_"))
154          + " to " + stepName.substring(stepName.lastIndexOf("_") + 1));
155      136    }
156      137    +
157      138    private static boolean isTimeoutOrExceptionStep(Step currentStep) {
158      139      return !currentStep.getReason().isEmpty() && !currentStep.getStepName().contains("FEEDBACK_FROM_CRODC");
159      140    }
160      141    +
161      142    private static void handleTimeoutOrExceptionStep(Step currentStep, Task task) {
162      143      String description = currentStep.getReason().get(0).getDescription();
163      144      task.setType(description.contains("timed out") || description.contains("Timeout") ? TIME_OUT : EXCEPTION);
164      145      task.setDescription(description);
165      146    }
166      147    +
167      148    private static void setRejectDetails(Step currentStep, Task task, PortingRequest portingRequest) {
168      149      task.setReason(currentStep.getReason());
169      150      TaskType taskType;
170      ...
171      ...

```

src/test/java/be/telenet/cnp/handler/TaskRequestHandlerTest.java

```

... ...
2 2     @@ -2,6 +2,7 @@ package be.telenet.cnp.handler;
3 3     import static be.telenet.cnp.HandlerTestStub.readFileToString;
4 4     import static be.telenet.cnp.util.TaskHandlerUtils.deserialize;
5 +    import static org.junit.jupiter.api.Assertions.assertNotNull;
6 
7 8     import be.telenet.cnp.data.model.Event;
... ...
27 28     private static final String EVENT_JSON = "src/test/resources/Event.json";
29
30
31 +    @Test
32 +    void testDefaultConstructorHandleMessage() throws IOException, InterruptedException {
33 +        TaskRequestHandler handler = new TaskRequestHandler();
34 +        SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
35 +        sqsMessage.setBody(readFileToString(EVENT_JSON));
36 +        Event mockEvent = deserialize(sqsMessage.getBody(), Event.class);
37 +
38 +        assertNotNull(handler);
39 +
40 +        taskRequestHandler.handleMessage(sqsMessage);
41 +        Mockito.verify(taskRequestService).process(mockEvent);
42 +    }
43 +
44
45 +    @Test
46 +    void testHandleMessage() throws IOException, InterruptedException {
47 +        SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
48 ...
49 +        @@ -43,4 +57,19 @@ class TaskRequestHandlerTest {
50 +            sqsMessage.setBody("");
51 +            assertThrows(TaskRequestException.class, () -> taskRequestHandler.handleMessage(sqsMessage));
52 +        }
53 +
54 +        @Test
55 +        void testHandleMessageWithInterruptedException() throws InterruptedException, IOException {
56 +            SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
57 +            sqsMessage.setBody(readFileToString(EVENT_JSON));
58 +            Event mockEvent = deserialize(sqsMessage.getBody(), Event.class);
59 +
60 +            Mockito.doThrow(new InterruptedException("Simulated InterruptedException"))
61 +                .when(taskRequestService)
62 +                .process(mockEvent);
63 +
64 +            assertThrows(TaskRequestException.class, () -> taskRequestHandler.handleMessage(sqsMessage));
65 +
66 +            Mockito.verify(taskRequestService).process(mockEvent);
67 +        }
68 +
69 +    }
70
71 +}
72
73 +}
74 +
75 +}

```

src/test/java/be/telenet/cnp/service/TaskRequestServiceTest.java

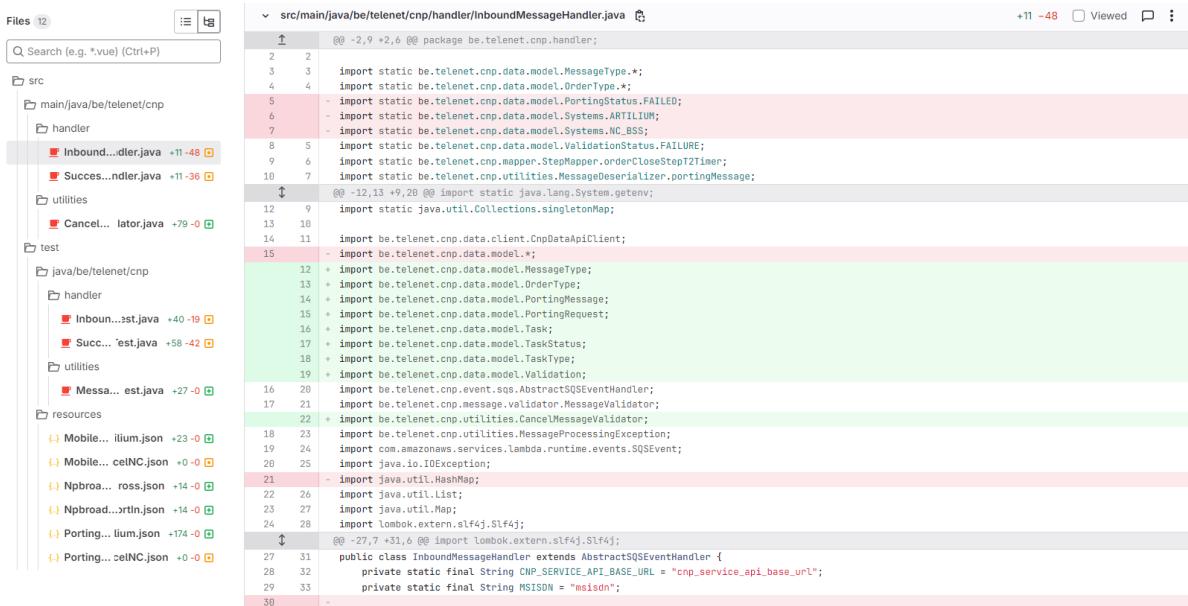
```

... ...
8 8     @@ -8,6 +8,8 @@ import static org.mockito.Mockito.when;
9
10 +    import be.telenet.cnp.data.client.CnpDataApiClient;
11 +    import be.telenet.cnp.data.model.Event;
12 +    import be.telenet.cnp.data.model.OrderType;
13 +    import be.telenet.cnp.data.model.PortingRequest;
14 +    import be.telenet.cnp.data.model.Task;
15 +    import be.telenet.cnp.data.model.TaskStatus;
16 +    import java.io.IOException;
17 ...
18 +    @@ -151,4 +153,33 @@ class TaskRequestServiceTest {
19 +        taskRequestService.process(event);
20 +        verify(cnpDataApiClient).createTask(any());
21 +    }
22 +
23 +    @Test
24 +    void testCreateTaskWithTargetSystems() throws IOException, InterruptedException {
25 +        Event mockEvent = deserialize(readFileToString(EVENT_JSON), Event.class);
26 +
27 +        PortingRequest portingRequest = generatePortingRequest(PORTINGREQUEST_JSON);
28 +        portingRequest.setOrderType(OrderType.PORT_ACROSS);
29 +
30 +        when(cnpDataApiClient.getPortingRequestById(mockEvent.getNpRequestId())).thenReturn(portingRequest);
31 +
32 +        taskRequestService.process(mockEvent);
33 +
34 +        verify(cnpDataApiClient).createTask(any());
35 +    }
36
37 +}

```

```
171 | +     @Test
172 | +     void testCreateTaskWithNullTargetSystems() throws IOException, InterruptedException {
173 | +         Event mockEvent = deserialize(readFileToString(EVENT_JSON), Event.class);
174 | +
175 | +         PortingRequest portingRequest = generatePortingRequest(PORTINGREQUEST_JSON);
176 | +         portingRequest.setOrderType(OrderType.PORT_ACROSS);
177 | +         portingRequest.setTargetSystems(null);
178 | +
179 | +         when(cnpDataApiClient.getPortingRequestById(mockEvent.getNpRequestId())).thenReturn(portingRequest);
180 | +
181 | +         taskRequestService.process(mockEvent);
182 | +
183 | +         verify(cnpDataApiClient).createTask(any());
184 | +
185 | }
```

CNP-Inbound-Message-Handler (coverage)



The screenshot shows a SonarQube coverage report for the file `InboundMessageHandler.java`. The report includes a navigation bar with tabs for 'Files' (selected), 'Issues', 'Code', and 'Duplications'. A search bar at the top right contains the placeholder 'Search (e.g. *.vue) (Ctrl+P)'. The left sidebar lists project structure under 'src' and 'test' directories, with specific files like `InboundMessageHandler.java`, `Succes...ndler.java`, and `Cancel...ator.java` highlighted. The main pane displays the Java code with color-coded coverage status: red for unexecuted code, green for partially executed, and grey for fully executed. The code imports various classes from the `be.telenet.cnp` package and other standard Java libraries. The coverage summary at the bottom indicates 11/48 lines covered.

```
src/main/java/be/telenet/cnp/handler/InboundMessageHandler.java
@ -2,9 +2,6 @@ package be.telenet.cnp.handler;
import static be.telenet.cnp.data.model.MessageType.*;
import static be.telenet.cnp.data.model.OrderType.*;
import static be.telenet.cnp.data.model.PortingStatus.FAILED;
import static be.telenet.cnp.data.model.Systems.ARTELUM;
import static be.telenet.cnp.data.model.Systems.NC_BSS;
import static be.telenet.cnp.data.model.ValidationStatus.FAILURE;
import static be.telenet.cnp.mapper.StepMapper.orderCloseStepT2Timer;
import static be.telenet.cnp.utilities.MessageDeserializer.portingMessage;
import static java.lang.System.getenv;
import static java.util.Collections.singletonList;
import be.telenet.cnp.data.client.CnpDataApiClient;
import be.telenet.cnp.data.model.*;
import be.telenet.cnp.data.model.MessageType;
import be.telenet.cnp.data.model.OrderType;
import be.telenet.cnp.data.model.PortingMessage;
import be.telenet.cnp.data.model.PortingRequest;
import be.telenet.cnp.data.model.Task;
import be.telenet.cnp.data.model.TaskStatus;
import be.telenet.cnp.data.model.TaskType;
import be.telenet.cnp.data.model.Validation;
import be.telenet.cnp.event.sqs.AbstractSQSEventHandler;
import be.telenet.cnp.message.validator.MessageValidator;
import be.telenet.cnp.utilities.CancelMessageValidator;
import be.telenet.cnp.utilities.MessageProcessingException;
import com.amazonaws.services.lambda.runtime.events.SQSEvent;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import lombok.extern.slf4j.Slf4j;
public class InboundMessageHandler extends AbstractSQSEventHandler {
    private static final String CNP_SERVICE_API_BASE_URL = "cnp_service_api_base_url";
    private static final String MSISDN = "msisdn";
}
```



```

▼ src/main/java/be/telenet/cnp/handler/SuccessValidationHandler.java ▲
+11 -36 □ Viewed □ ::

↑ @@ -31,16 +31,23 @@ import be.telenet.cnp.data.model.TaskType;
31 31     import be.telenet.cnp.data.model.Validation;
32 32     import be.telenet.cnp.model.InternalTimerStepName;
33 33     import be.telenet.cnp.model.OperatorRoutingNumber;
34 34 + import be.telenet.cnp.utilities.CancelMessageValidator;
35 35     import be.telenet.cnp.utilities.MessageProcessingException;
36 36     import java.io.IOException;
37 37 - import java.util.*;
38 38 + import java.util.ArrayList;
39 39 + import java.util.HashMap;
40 40 + import java.util.List;
41 41 + import java.util.Map;
42 42     import lombok.extern.slf4j.Slf4j;
43 43     @Slf4j
44 44     public class SuccessValidationHandler extends AbstractMessageHandler {
45 45
46 46     +     private final CnpDataApiClient client;
47 47
48 48     public SuccessValidationHandler(CnpDataApiClient client) {
49 49         super(client);
50 50 +         this.client = client;
51 51     }
52 52
53 53     @Override
↓ ↑ @@ -142,7 +149,7 @@ public class SuccessValidationHandler extends AbstractMessageHandler {
142 149     if (null != portingRequest
143 150         && portingRequest.getInternalPorting()
144 151         && isValidInternalTimerStepName(currentStep.getStepName())) {
145 151 -     taskTypes.addAll(Collections.singleton(INTERNALPORTING_TIMER_VIOLATION));
152 152 +     taskTypes.add(INTERNALPORTING_TIMER_VIOLATION);
146 153 } else {
147 154     taskTypes.addAll(findTimerViolation(currentStep.getStepName()));
148 155 }
176 183     private PortingRequest getPortingRequest(PortingMessage message) throws IOException, InterruptedException {
177 184         OrderType orderType = message.getOrderType();
178 185
179 186 -     if (isCancelMessage(message)) {
180 187 +     CancelMessageValidator cancelMessageValidator = new CancelMessageValidator(this.client);
181 188 +     if (cancelMessageValidator.isCancelMessage(message)) {
182 189             if (NC_BSS == message.getInitiatedBy()) return fetchPortingRequest(message);
183 190             if (ARTILIJUM == message.getInitiatedBy())
184 191                 return getPortingRequests(
↓ ↑ @@ -210,39 +218,6 @@ public class SuccessValidationHandler extends AbstractMessageHandler {
210 218         .orElse(null);
211 219     }
220
221 -     private boolean isCancelMessage(PortingMessage portingMessage) throws IOException, InterruptedException {
222 -         log.info("messageType: {}, orderType: {}", portingMessage.getMessageType(), portingMessage.getOrderType());
223 -         if (NP_CANCEL == portingMessage.getMessageType()) {
224 -             List<PortingRequest> requests = getPortingRequests(
225 -                 NC_BSS == portingMessage.getInitiatedBy()
226 -                     ? searchParams(
227 -                         portingMessage
228 -                             .getPortingDetails()
229 -                             .getMsisdns()
230 -                             .get(0),
231 -                         null)
232 -                     : searchParams(
233 -                         null,
234 -                         portingMessage.getPortingDetails().getNprNumber()));
235 -             if (requests.size() == 1) {
236 -                 return (NC_BSS == portingMessage.getInitiatedBy()
237 -                     && PORT_IN == requests.get(0).getOrderType()
238 -                     && FAILED == requests.get(0).getPortingStatus()
239 -                     || ARTILIJUM == portingMessage.getInitiatedBy());
240 -             } else if (requests.size() > 1) {
241 -                 return (NC_BSS == portingMessage.getInitiatedBy()
242 -                     && PORT_IN == portingMessage.getOrderType()
243 -                     && requests.stream().anyMatch(request -> request.getSteps()
244 -                         .get(0)
245 -                         .getTracking())

```

```

237      .getCorrelationId()
238      .equalsIgnoreCase(
239          portingMessage.getTracking().getCorrelationId())))
240      || (ARTILIUM == portingMessage.getInitiatedBy() && PORT_IN == portingMessage.getOrderType());
241  }
242 }
243 return false;
244 }
245
246 221 private Map<String, String> searchParams(PortingMessage message) {
247 222     String nprNumber = message.getPortingDetails().getNprNumber();
248 223     return (null != nprNumber)

```

```

▼ src/main/java/be/telenet/cnp/utilities/CancelMessageValidator.java 0 → 100644 +79 -0 □ Viewed □ ·
1 + package be.telenet.cnp.utilities;
2 +
3 + import static be.telenet.cnp.data.model.MessageType.NP_CANCEL;
4 + import static be.telenet.cnp.data.model.OrderType.PORT_IN;
5 + import static be.telenet.cnp.data.model.PortingStatus.FAILED;
6 + import static be.telenet.cnp.data.model.Systems.ARTILIUM;
7 + import static be.telenet.cnp.data.model.Systems.NC_BSS;
8 +
9 + import be.telenet.cnp.data.client.CnpDataApiClient;
10+ import be.telenet.cnp.data.model.PortingMessage;
11+ import be.telenet.cnp.data.model.PortingRequest;
12+ import java.io.IOException;
13+ import java.util.HashMap;
14+ import java.util.List;
15+ import java.util.Map;
16+ import lombok.extern.slf4j.Slf4j;
17+
18+ @Slf4j
19+ public class CancelMessageValidator {
20+     private final CnpDataApiClient client;
21+
22+     public CancelMessageValidator(CnpDataApiClient client) {

```

22+ public CancelMessageValidator(CnpDataApiClient client) {
23+ this.client = client;
24+ }
25+
26+ public boolean isCancelMessage(PortingMessage portingMessage) throws IOException, InterruptedException {
27+ log.info("messageType: {}, orderType: {}", portingMessage.getMessageType(), portingMessage.getOrderType());
28+
29+ if (isCancelMessageType(portingMessage)) {
30+ List<PortingRequest> requests = fetchPortingRequests(portingMessage);
31+
32+ if (requests.size() == 1) {
33+ return validateSingleRequest(portingMessage, requests.get(0));
34+ } else if (requests.size() > 1) {
35+ return validateMultipleRequests(portingMessage, requests);
36+ }
37+ }
38+
39+ return false;
40+ }
41+
42+ private boolean isCancelMessageType(PortingMessage portingMessage) {
43+ return NP_CANCEL == portingMessage.getMessageType();
44+ }
45+
46+ private List<PortingRequest> fetchPortingRequests(PortingMessage portingMessage)
47+ throws IOException, InterruptedException {
48+ Map<String, String> params = NC_BSS == portingMessage.getInitiatedBy()
49+ ? searchParams(portingMessage.getPortingDetails().getMsidsns().get(0), null)
50+ : searchParams(null, portingMessage.getPortingDetails().getNprNumber());
51+ return client.getPortingRequests(params);
52+ }
53+
54+ private boolean validateSingleRequest(PortingMessage portingMessage, PortingRequest request) {
55+ return (NC_BSS == portingMessage.getInitiatedBy()
56+ && PORT_IN == request.getOrderType()
57+ && FAILED == request.getPortingStatus())
58+ || ARTILIUM == portingMessage.getInitiatedBy();
59+ }

```

61 +     private boolean validateMultipleRequests(PortingMessage portingMessage, List<PortingRequest> requests) {
62 +         return (NC_BSS == portingMessage.getInitiatedBy()
63 +                 && PORT_IN == portingMessage.getOrderType()
64 +                 && requests.stream().anyMatch(request -> request.getSteps()
65 +                     .get(0)
66 +                     .getTracking()
67 +                     .getCorrelationId()
68 +                     .equalsIgnoreCase(portingMessage.getTracking().getCorrelationId())));
69 +     }
70 +
71 +
72 +     private Map<String, String> searchParams(String msisdn, String npnNumber) {
73 +         Map<String, String> params = new HashMap<>();
74 +         if (null != npnNumber) params.put("npnNumber", npnNumber);
75 +         if (null != msisdn) params.put("msisdn", msisdn);
76 +         params.put("portingStatus", "OPEN_FAILED");
77 +         return params;
78 +     }
79 + }

```

src/test/java/be/telenet/cnp/handler/InboundMessageHandlerTest.java

+40 -19 Viewed

```

15 15     @@ -15,6 +15,7 @@ import be.telenet.cnp.utilities.MessageProcessingException;
16 16     import com.amazonaws.services.lambda.runtime.events.SQSEvent;
17 17     import java.io.IOException;
18 +    import java.util.ArrayList;
18 19     import java.util.Collections;
19 20     import java.util.List;
20 21     import lombok.SneakyThrows;
21 22     import org.junit.jupiter.api.Test;
22 23
23 24     @@ -30,10 +31,10 @@ class InboundMessageHandlerTest {
24 25         private static final String NPBROADCAST_JSON_INTERNAL = "src/test/resources/NpbroadcastInternalPorting.json";
25 26         private static final String NP_DONE = "src/test/resources/MobileBroadcastDone.json";
26 27         private static final String NP_REQUEST = "src/test/resources/NpRequest.json";
27 28         - private static final String NP_CANCEL = "src/test/resources/MobilePortInCancel.json";
28 29         + private static final String NP_CANCEL = "src/test/resources/MobilePortInCancelNC.json";
29 30         private static final String NP_CANCEL_ARTILIUM = "src/test/resources/Npcancel.json";
30 31
31 32     private static final String PORTING_REQUEST = "src/test/resources/PortingRequestCancelNC.json";
32 33     private static final String PORTING_REQUEST_SAMPLE = "src/test/resources/porting-request-internal.json";
33 34
34 35     @InjectMocks
35 36
36 37     @@ -80,9 +81,7 @@ class InboundMessageHandlerTest {
37 38         void testHandleMessageForIOException() {
38 39             SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
39 40             sqsMessage.setBody("");
40 41
41 42             assertThrows(RuntimeException.class, () -> {
42 43                 handler.handleMessage(sqsMessage);
43 44             });
44 45             assertThrows(RuntimeException.class, () -> handler.handleMessage(sqsMessage));
45 46         }
46 47
47 48         @Test
48 49             @@ -95,9 +94,7 @@ class InboundMessageHandlerTest {
49 50                 doThrow(new MessageProcessingException("Handling Failed!", new Exception())
50 51                         .when(successHandler)
51 52                         .handleMessage(any(), eq(validation)));
52 53
53 54             - assertThrows(MessageProcessingException.class, () -> {
54 55                 handler.handleMessage(sqsMessage);
55 56             });
56 57             + assertThrows(MessageProcessingException.class, () -> handler.handleMessage(sqsMessage));
57 58         }
58 59
59 60         @Test
60 61             @@ -144,6 +141,30 @@ class InboundMessageHandlerTest {
61 62                 verify(successHandler).handleMessage(any(PortingMessage.class), eq(null));
62 63             }
63 64
64 65             + @Test
65 66             + @SneakyThrows
66 67             void testPortOutNpdoneNoRequests() {
67 68                 SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
68 69                 sqsMessage.setBody(readFileToString(NP_DONE));
69 70                 when(client.getPortingRequests(any())).thenReturn(null);

```

```
150 +         handler.handleMessage(sqsMessage);
151 +     verify(successHandler).handleMessage(any(PortingMessage.class), eq(null));
152 + }
153 +
154 + @Test
155 + @SneakyThrows
156 + void testPortOutNpdoneWithPortAcrossRequest() {
157 +     SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
158 +     sqsMessage.setBody(readFileToString(NP_DONE));
159 +     List<PortingRequest> portingRequests = new ArrayList<>();
160 +     portingRequests.add(generatePortingRequestPO(OrderType.PORT_ACROSS, Systems.CRDC_BASE));
161 +     Step testStep = Step.builder().stepName("BROADCAST_TO_NC").build();
162 +     portingRequests.forEach(portingRequest -> portingRequest.setSteps(Collections.singletonList(testStep)));
163 +     when(client.getPortingRequests(any())).thenReturn(portingRequests);
164 +     handler.handleMessage(sqsMessage);
165 +     verify(successHandler).handleMessage(any(PortingMessage.class), eq(null));
166 + }
167 +
168 + @Test
169 + @SneakyThrows
170 + void testPortOutNprequest() {
171 +     @@ -177,22 +198,22 @@ class InboundMessageHandlerTest {
172 +         @Test
173 +         @SneakyThrows
174 +         void testCancelMessageArtium() {
175 +             SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
176 +             sqsMessage.setBody(readFileToString(NP_CANCEL_ARTIUM));
177 +             List<PortingRequest> portingRequests = new ArrayList<>();
178 +             portingRequests.add(createPortingRequest(PORTING_REQUEST));
179 +             portingRequests.add(createPortingRequest(PORTING_REQUEST_SAMPLE));
180 +             when(client.getPortingRequests(any())).thenReturn(portingRequests);
181 +             handler.handleMessage(sqsMessage);
182 +             verify(successHandler).handleMessage(any(PortingMessage.class), eq(null));
183 +         }
184 +         void testCancelMessageNc() {
185 +             runCancelMessage(NP_CANCEL);
186 +         }
187 +     }
188 +
189 +     @Test
190 +     @SneakyThrows
191 +     void testCancelMessageNc() {
192 +         void testCancelMessageArtium() {
193 +             runCancelMessage(NP_CANCEL_ARTIUM);
194 +         }
195 +         // Helper methods
196 +         @SneakyThrows
197 +         void runCancelMessage(String filepath) {
198 +             SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
199 +             sqsMessage.setBody(readFileToString(NP_CANCEL));
200 +             sqsMessage.setBody(readFileToString(filepath));
201 +             List<PortingRequest> portingRequests = new ArrayList<>();
202 +             portingRequests.add(createPortingRequest(PORTING_REQUEST));
203 +             portingRequests.add(createPortingRequest(PORTING_REQUEST_SAMPLE));
204 +         }
205 +     }
206 + }
```

```
src/test/java/be/telenet/cnp/handler/SuccessValidationHandlerTest.java
```

11	11	@@ -11,6 +11,7 @@ import be.telenet.cnp.data.client.CnpDataApiClient;
12	12	import be.telenet.cnp.data.model.*;
13	13	import java.io.IOException;
14	14	import java.util.ArrayList;
15	15	+ import java.util.Collections;
16	16	import java.util.List;
17	17	import lombok.SneakyThrows;
		import org.junit.jupiter.api.Test;
35	35	@@ -35,17 +36,19 @@ class SuccessValidationHandlerTest {
36	36	private static final String NP_EXEC = "src/test/resources/Npexec.json";
37	37	private static final String NP_READY = "src/test/resources/Npready.json";
38	38	private static final String NP_BROADCAST = "src/test/resources/Npbroadcast.json";
39	39	+ private static final String NP_BROADCAST_PORT_IN = "src/test/resources/NpbroadcastPortIn.json";
40	40	+ private static final String NP_BROADCAST_PORT_ACROSS = "src/test/resources/NpbroadcastPortAcross.json";
41	41	- private static final String NC_ACCEPT = "src/test/resources/NcNpaccept.json";
42	42	private static final String NP_DISCONNECT_IN = "src/test/resources/Npdisconnect.json";

```

41 43     private static final String NP_SYSTEM_UP = "src/test/resources/Npsystemup.json";
42 44     private static final String NP_SYSTEM_DOWN = "src/test/resources/Npsystemdown.json";
43 -     private static final String PORTING_REQUEST_CANCEL = "src/test/resources/PortingRequestCancel.json";
44 -
45 +     private static final String PORTING_REQUEST_CANCEL_NC = "src/test/resources/PortingRequestCancelNC.json";
46 +     private static final String PORTING_REQUEST_CANCEL_ARTILIJUM =
47 +         "src/test/resources/PortingRequestCancelArtijum.json";
45 48     private static final String PORTING_REQUEST = "src/test/resources/PortingRequest.json";
46 49     private static final String PORTING_REQUEST_INTERNAL = "src/test/resources/porting-request-internal.json";
47 -     private static final String NP_CANCEL = "src/test/resources/MobilePortInCancel.json";
48 -
49 +     private static final String NP_CANCEL_NC = "src/test/resources/MobilePortInCancelNC.json";
50 +     private static final String NP_CANCEL_ARTILIJUM = "src/test/resources/MobilePortInCancelArtijum.json";
51 +
52 53     private static final String READY_PORTING_REQUEST = "src/test/resources/ReadyPortingRequest.json";
51 54
51 54     @Test
52 ↓
52 54     @@ -157,13 +160,15 @@ class SuccessValidationHandlerTest {
53
54 160     }
55 161
56 162     @Test
57 -     void testHandleMessageForNpbroadcast() throws IOException, InterruptedException {
58 -         PortingMessage portingMessage = generatePortingMessage(NP_BROADCAST);
59 +
60 +     void testHandleMessageForMultipleMsisdn() throws IOException, InterruptedException {
61 +         PortingMessage portingMessage = generatePortingMessage(NC_ACCEPT);
62
63 165         Validation validation = null;
64 166         List<PortingRequest> portingRequestList = new ArrayList<>();
65 167         PortingRequest portingRequest = PortingRequest.builder()
66 168             .portingDetails(PortingDetails.builder()
67 169                 .msisdns(List.of("0412345678", "0412345679"))
68 170                 .recipient(Recipient.builder().code("BASE").build())
69 171                 .serviceClass(ServiceClass.POST_PAID_COMPLEX)
70 172                 .build()
71 173                 .internalPorting(false)
72 174                 .orderType(OrderType.PORT_OUT)
73
74 ↓
74 177 -177,62 +182,73 @@ class SuccessValidationHandlerTest {
75
76 182     }
77 183
78 184     @Test
79 -     void testHandleMessageForMultipleMsisdn() throws IOException, InterruptedException {
80 -         PortingMessage portingMessage = generatePortingMessage(NC_ACCEPT);
81 -         Validation validation = null;
82 -         List<PortingRequest> portingRequestList = new ArrayList<>();
83 -         PortingRequest portingRequest = PortingRequest.builder()
84 -             .portingDetails(PortingDetails.builder()
85 -                 .msisdns(List.of("0412345678", "0412345679"))
86 -                 .recipient(Recipient.builder().code("BASE").build())
87 -                 .serviceClass(ServiceClass.POST_PAID_COMPLEX)
88 -                 .build()
89 -                 .internalPorting(false)
90 -                 .orderType(OrderType.PORT_OUT)
91 -                 .build());
92
93         portingRequestList.add(portingRequest);
94 +
95 +     void testHandleMessageForNpbroadcast() throws IOException, InterruptedException {
96 +         runHandleMessage(OrderType.PORT_OUT, NP_BROADCAST);
97 +
98 194 -         when(client.getPortingRequests(any())).thenReturn(portingRequestList);
99 +
100 +     @Test
101 +     void testHandleMessageForNpbroadcastPortIN() throws IOException, InterruptedException {
102 +         runHandleMessage(OrderType.PORT_IN, NP_BROADCAST_PORT_IN);
103 +
104 196 -         successHandler.handleMessage(portingMessage, validation);
105 -         verify(client).createPortingRequest(any(PortingRequest.class));
106 +
107 +     @Test
108 +     void testHandleMessageForNpbroadcastPortAcross() throws IOException, InterruptedException {
109 +         Step testStep = Step.builder().stepName("testStep").build();
110 +         List<Step> steps = Collections.singletonList(testStep);
111 +
112 +         runHandleMessage(OrderType.PORT_ACROSS, NP_BROADCAST_PORT_ACROSS, steps);
113     }
114
115 200 201
116 202     @Test
117 -     void testHandleMessageForNpbroadcastPortIn() throws IOException, InterruptedException {
118 -         PortingMessage portingMessage = PortingMessage.builder()
119 -

```

```

204 -         .messageType(MessageType.NP_BROADCAST)
205 -         .orderType(OrderType.PORT_IN)
206 -         .portingDetails(
207 -             PortingDetails.builder().msisdns(List.of("0412345678")).build())
208 -             .initiatedBy(Systems.NC_BSS)
209 -             .build();
210 -         Validation validation = null;
211 -         List<PortingRequest> portingRequestList = new ArrayList<>();
212 +     @SneakyThrows
213 +     void testCancelMessageNC() {
214 +         runCancelMessageTest(PORTING_REQUEST_CANCEL_NC, NP_CANCEL_NC);
215 +     }
216 +     @Test
217 +     @SneakyThrows
218 +     void testCancelMessageArtilium() {
219 +         runCancelMessageTest(PORTING_REQUEST_CANCEL_ARTILIUM, NP_CANCEL_ARTILIUM);
220 +     }
221 +     // Helper methods
222 +
223 +     private void runHandleMessage(OrderType orderType, String portingMessageType)
224 +         throws IOException, InterruptedException {
225 +         runHandleMessage(orderType, portingMessageType, Collections.emptyList());
226 +     }
227 +
228 +     private void runHandleMessage(OrderType orderType, String portingMessageType, List<Step> steps)
229 +         throws IOException, InterruptedException {
230 +         PortingMessage portingMessage = generatePortingMessage(portingMessageType);
231 +
232 +         PortingRequest portingRequest = PortingRequest.builder()
233 +             .portingDetails(PortingDetails.builder()
234 +                 .recipient(Recipient.builder().code("BASE").build())
235 +                 .build())
236 +                 .internalPorting(false)
237 +                 .orderType(OrderType.PORT_IN)
238 +                 .orderType(orderType)
239 +                 .steps(steps)
240 +             .build());
241 +         portingRequestList.add(portingRequest);
242 +
243 +         List<PortingRequest> portingRequestList = Collections.singletonList(portingRequest);
244 +
245 +         when(client.getPortingRequests(any())).thenReturn(portingRequestList);
246 +
247 +         successHandler.handleMessage(portingMessage, validation);
248 +         successHandler.handleMessage(portingMessage, null);
249 +
250 +         verify(client).createPortingRequest(any(PortingRequest.class));
251 +
252 +     }
253 +
254 +     @Test
255 +     @SneakyThrows
256 +     void testCancelMessageNc() {
257 +
258 +         private void runCancelMessageTest(String portingRequestType, String portingMessageType) {
259 +             List<PortingRequest> portingRequests = new ArrayList<>();
260 +             portingRequests.add(createPortingRequest(PORTING_REQUEST_CANCEL));
261 +             portingRequests.add(createPortingRequest(portingRequestType));
262 +             portingRequests.add(createPortingRequest(PORTING_REQUEST_INTERNAL));
263 +             when(client.getPortingRequests(any())).thenReturn(portingRequests);
264 +
265 +             successHandler.handleMessage(generatePortingMessage(NP_CANCEL), null);
266 +
267 +             successHandler.handleMessage(generatePortingMessage(portingMessageType), null);
268 +
269 +             verify(client).createPortingRequest(any(PortingRequest.class));
270 +
271 +         }
272 +
273 +     }

```

src/test/java/be/telenet/cnp/utilities/MessageDeserializerTest.java 0 → 100644

+27 -0 Viewed

```

1 + package be.telenet.cnp.utilities;
2 +
3 + import static org.junit.jupiter.api.Assertions.assertThrows;
4 + import static org.junit.jupiter.api.Assertions.assertTrue;
5 +

```

```

6 + import java.lang.reflect.Constructor;
7 + import java.lang.reflect.InvocationTargetException;
8 + import java.lang.reflect.Modifier;
9 + import lombok.SneakyThrows;
10 + import org.junit.jupiter.api.Test;
11 +
12 + class MessageDeserializerTest {
13 +
14 +     @Test
15 +     @SneakyThrows
16 +     void testMessageDeserializerConstructor() {
17 +         Constructor<MessageDeserializer> constructor = MessageDeserializer.class.getDeclaredConstructor();
18 +
19 +         assertTrue(Modifier.isPrivate(constructor.getModifiers()));
20 +
21 +         constructor.setAccessible(true);
22 +         InvocationTargetException exception = assertThrows(InvocationTargetException.class, constructor::newInstance);
23 +
24 +         assertTrue(exception.getCause() instanceof IllegalStateException);
25 +         assertTrue(exception.getCause().getMessage().contains("MessageDeserializer class"));
26 +     }
27 + }

```

src/test/resources/MobilePortInCancelArtilium.json 0 → 100644 +23 -0 Viewed

```

1 + {
2 +     "messageType": "NP_CANCEL",
3 +     "orderType": "PORT_IN",
4 +     "initiatedBy": "ARTILIUM",
5 +     "portingDetails": {
6 +         "msisdns": [
7 +             "0455009795"
8 +         ],
9 +         "serviceClass": null,
10 +        "nprNumber": null,
11 +        "recipient": null,
12 +        "donor": null,
13 +        "customer": null,
14 +        "routing": null
15 +    },
16 +    "tracking": {
17 +        "requestId": "00e277f0-6f8c-4dd3-9e46-e335a24ba28e",
18 +        "messageId": "014e84b5-307b-4203-abc3-9b1e6c983af3",
19 +        "correlationId": "01d02034-bacb-48d1-81cd-3ab755a24b22"
20 +    },
21 +    "reason": null,
22 +    "creationDateTime": null
23 + }

```

src/test/resources/MobilePortInCancel.json → src/test/resources/MobilePortInCancelNC.json +0 -0 Viewed

File renamed with no changes. Show file contents

src/test/resources/NpbroadcastPortAcross.json 0 → 100644 +14 -0 Viewed

```

1 + {
2 +     "messageType": "NP_DONE",
3 +     "orderType": "PORT_ACROSS",
4 +     "initiatedBy": "CRDC_BASE",
5 +     "creationDateTime": "2023-04-20T15:32:21.675+02:00",
6 +     "portingDetails": {
7 +         "msisdns": [
8 +             "0468352656"
9 +         ],
10 +        "routing" : {
11 +            "number": "123"
12 +        }
13 +    }
14 + }

```

\ No newline at end of file

src/test/resources/NpbroadcastPortIn.json 0 → 100644 +14 -0 Viewed

```

1 + {
2 +     "messageType": "NP_BROADCAST",

```

```

3 + "orderType": "PORT_IN",
4 + "initiatedBy": "CRDC_BASE",
5 + "creationDateTime": "2023-04-20T15:32:21.675+02:00",
6 + "portingDetails": {
7 +   "msisdns": [
8 +     "8468352656"
9 +   ],
10 +   "routing": {
11 +     "number": "123"
12 +   }
13 + }
14 +
\ No newline at end of file

```

src/test/resources/PortingRequestCancelArtilium.json 0 → 100644

+174 -0 Viewed

```

1 + {
2 +   "id": "37191da0-452c-45cf-9c60-8b31d013353e",
3 +   "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
4 +   "orderType": "PORT_IN",
5 +   "creationUser": "CNP",
6 +   "modificationUser": "CNP",
7 +   "portingDetails": {
8 +     "msisdns": [
9 +       "8455009795"
10 +     ],
11 +     "serviceClass": "PRE_PAID",
12 +     "nprNumber": "BASE202024103109484901",
13 +     "recipient": {
14 +       "code": "BASE",
15 +       "simCard": "8932030000106741465"
16 +     },
17 +     "donor": {
18 +       "simCard": "8932070010106734007",
19 +       "code": "BEMO"
20 +     },
21 +     "customer": {},
22 +     "routing": {}
23 +
"flag": "MOBILE",
24 +   "role": "RECIPIENT",
25 +   "portingStatus": "FAILED",
26 +   "targetSystems": [
27 +     "CRDC_BASE"
28 +   ],
29 +   "initiatedBy": "ARTILIUM",
30 +   "internalPorting": false,
31 +   "useType": "FULL_MVNO",
32 +   "bssSystem": "",
33 +   "originalOwner": "BASE",
34 +   "steps": [
35 +     {
36 +       "id": "bd9d73f8-c928-470d-9439-b477f0691bef",
37 +       "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
38 +       "messageType": "NPR",
39 +       "creationUser": "CNP",
40 +       "originalMessage": "<?xml version='1.0' encoding='UTF-8'?><MobilePortInRequest
+xmlns='http://xmlns.telenet.be/integration/messages/NumberPorting/MobilePortInRequest/v001'
+xmlns:ns1='http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002'>
<ns1:IntegrationHeader>          <ns1:OrganizationalInfo>
<ns1:Company>TN</ns1:Company>      <ns1:OrganizationalInfo>          <ns1:MessageIdentification>          <ns1:CorrelationId>4db6c409-529a-
439d-a710-0140b9198e5f</ns1:CorrelationId>    <ns1:RequestId>e8f52440-6272-4558-9889-9ba598e73bc1</ns1:RequestId>
<ns1:MessageId>e8f52440-6272-4558-9889-9ba598e73bc1</ns1:MessageId>    <ns1:MessageIdentification>          <ns1:MessageTimestamps>
<ns1:CreationTimestamp>2023-08-07T10:44:48.468+02:00</ns1:CreationTimestamp>    <ns1:MessageTimestamps>          <ns1:SystemInfo>
<ns1:Environment>SUP</ns1:Environment>        <ns1:Application>          <ns1:ApplicationName>ARTILIUM</ns1:ApplicationName>
</ns1:Application>        <ns1:SystemInfo>          <ns1:UserInfor>          <ns1:Identity>          <ns1:IdentityId>internal</ns1:IdentityId>
<ns1:IdentityType>user login</ns1:IdentityType>    <ns1:Identity>          </ns1:UserInfor>          </ns1:IntegrationHeader>
<MobilePortInRequestDetails>          <MSISDN>+32455009795</MSISDN>    <MobileServiceClass>PRE_PAID</MobileServiceClass>
<RecipientServiceProviderCode>BASE</RecipientServiceProviderCode>          <DonorSimCard>8932070010106734007</DonorSimCard>
<ReceiverSimCard>8932030000106741465</ReceiverSimCard>    <MobilePortInRequestDetails></MobilePortInRequest>",
41 +
"source": "ARTILIUM",
42 +   "stepName": "REQUEST RECEIVED FROM ARTILIUM",
43 +   "stepStatus": "OK",
44 +   "validation": {
45 +     "status": "SUCCESS"
46 +   },
47 +   "tracking": {
48 +     "requestId": "e8f52440-6272-4558-9889-9ba598e73bc1"
49 +

```

```

50 +         "messageId": "e0ff52440-6272-4558-9889-9ba598e73bc1",
51 +         "correlationId": "01d02034-bacb-48d1-81cd-3ab755a24b22"
52 +     },
53 +     "reason": []
54 +   },
55 +   {
56 +     "id": "4ed443e1-0fb7-464a-8f59-bd1bfbfb51e",
57 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
58 +     "creationUser": "CNP",
59 +     "stepName": "MSISON_REQUEST_MAPPING",
60 +     "stepStatus": "OK",
61 +     "validation": {},
62 +     "tracking": {},
63 +     "reason": []
64 +   },
65 +   {
66 +     "id": "9998974d-42a8-45d7-895e-3d28ed0bee38",
67 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
68 +     "creationUser": "CNP",
69 +     "stepName": "REQUEST_ENRICHED",
70 +     "stepStatus": "OK",
71 +     "validation": {},
72 +     "tracking": {},
73 +     "reason": []
74 +   },
75 +   {
76 +     "id": "7c4d63a3-b6a9-430e-a3c7-1247184a39a1",
77 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
78 +     "creationUser": "CNP",
79 +     "stepName": "DUPLICATE_CHECKED",
80 +     "stepStatus": "OK",
81 +     "validation": {
82 +       "status": "SUCCESS"
83 +     },
84 +     "tracking": {},
85 +     "reason": []
86 +   },
87 +   {
88 +     "id": "737dbc68-56e2-4190-9e67-294e0204a1b8",
89 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
90 +     "messageType": "NP_REQUEST",
91 +     "creationUser": "CNP",
92 +     "originalMessage": "<nprequest>\n      <version>01.00</version>\n      <datetime>2024-10-31T09:49:12</datetime>\n      <operatorid>BASE</operatorid>\n      <prepaid>\n        <number>8455809795</number>\n        <simcardnumber>893207001010673400</simcardnumber>\n      </prepaid>\n      <nprnumber>BASE202410310948901</nprnumber>\n    </nprequest>",
93 +     "source": "ARTILIUM",
94 +     "destination": "CRDC_BASE",
95 +     "stepName": "REQUEST_TO_CRDC",
96 +     "stepStatus": "OK",
97 +     "validation": {},
98 +     "tracking": {},
99 +     "reason": []
100 +   },
101 +   {
102 +     "id": "0134b761-ce5a-46c2-96cb-4d5b63dce236",
103 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
104 +     "messageType": "NP_REQUEST_FEEDBACK_RESPONSE",
105 +     "creationUser": "CNP",
106 +     "originalMessage": "<npsopserverresponse>\n      <version>01.00</version>\n      <datetime>2024-10-31T09:49:12.940362554</datetime>\n    </npsopserverresponse>",
107 +     "source": "CRDC_BASE",
108 +     "destination": "ARTILIUM",
109 +     "stepName": "REQUEST_FEEDBACK_FROM_CRDC",
110 +     "stepStatus": "OK",
111 +     "validation": {},
112 +     "tracking": {},
113 +     "reason": []
114 +   },
115 +   {
116 +     "id": "72383e6d-71df-49dd-8ac7-54376c8309c3",
117 +     "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
118 +     "messageType": "NPR_SNT",
119 +     "creationUser": "CNP",
120 +     "originalMessage": "<?xml version='1.0' encoding='UTF-8' standalone='yes'?>\n<v001:MobileInformativeResponse\n  xmlns:v002='http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002'\n  xmlns:v001='http://xmlns.telenet.be/integration/messages/NumberPorting/MobileInformativeResponse/v001'>\n    <v002:IntegrationHeader>\n
```

```

<v002:OrganizationalInfo/>\n          <v002:MessageIdentification>\n          <v002:CorrelationId>4db6c409-529a-439d-a710-\n0140b9198e5f</v002:CorrelationId>\n          <v002:RequestId>e8f52440-6272-4558-9889-9ba598e73bc1</v002:RequestId>\n<v002:MessageId>d7fe4483-42f7-4e3d-9757-c0fd4d21ce5cb</v002:MessageId>\n          </v002:MessageIdentification>\n          <v002:MessageTimestamps>\n          <v002:CreationTimestamp>2024-10-31T09:49:16.210+01:00</v002:CreationTimestamp>\n          </v002:MessageTimestamps>\n<v002:SystemInfo>\n          <v002:Environment>dev</v002:Environment>\n          <v002:Application>\n<v002:ApplicationName>CNP</v002:ApplicationName>\n          </v002:Application>\n          </v002:SystemInfo>\n          </v002:IntegrationHeader>\n<v001:MobileInformativeResponseDetails>\n          <v001:MSISDN>+32455009795</v001:MSISDN>\n<v001:Response>PORTIN_REQUEST_SENT</v001:Response>\n          </v001:MobileInformativeResponseDetails>\n          </v001:MobileInformativeResponse>\n",
121 +   "source": "CRDC_BASE",
122 +   "destination": "ARTILIUM",
123 +   "stepName": "REQUEST_FEEDBACK_TO_ARTILIUM",
124 +   "stepStatus": "OK",
125 +   "validation": {},
126 +   "tracking": {
127 +     "requestId": "e8f52440-6272-4558-9889-9ba598e73bc1",
128 +     "messageId": "d7fe4483-42f7-4e3d-9757-c0fd4d21ce5cb",
129 +     "correlationId": "4db6c409-529a-439d-a710-0140b9198e5f"
130 +   },
131 +   "reason": []
132 + },
133 + {
134 +   "id": "fd717bec-cd25-4c0e-9b2f-bac2a7f7dd57",
135 +   "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
136 +   "messageType": "NP_REJECT",
137 +   "creationUser": "CNP",
138 +   "npnNumber": "BASE202024103109484901",
139 +   "originalMessage": "<nreject>      <version>01.00</version>      <datetime>2024-10-31T09:53:56.376138161</datetime>\n<nprnumber>BASE202024103109484901</nprnumber>      <rejectreason>      <rejectcode>145</rejectcode>      <rejectstring>Account number/ vat number\nmismatch</rejectstring>      </rejectreason></nreject>",
140 +   "source": "CRDC_BASE",
141 +   "destination": "ARTILIUM",
142 +   "stepName": "REJECT_FROM_CRDC",
143 +   "stepStatus": "OK",
144 +   "validation": {
145 +     "status": "SUCCESS"
146 +   },
147 +   "tracking": {},
148 +   "reason": [
149 +     {
150 +       "code": "145",
151 +       "description": "Account number/ vat number mismatch"
152 +     }
153 +   ],
154 + },
155 + {
156 +   "id": "f2001a7b-d57b-4dee-8bd8-5700b1ce3b32",
157 +   "npRequestId": "37191da0-452c-45cf-9c60-8b31d013353e",
158 +   "messageType": "NP_REJECT_RCV",
159 +   "creationUser": "CNP",
160 +   "originalMessage": "<?xml version='1.0'?>\n<!-- standalone="yes" -->\n<v001:MobilePortInReject\nxmlns:v002='http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002'\nxmlns:v001='http://xmlns.telenet.be/integration/messages/NumberPorting/MobilePortInReject/v001'>\n          <v002:CorrelationId>4db6c409-529a-439d-a710-\n0140b9198e5f</v002:CorrelationId>\n          <v002:RequestId>3865f790-0295-4ed8-9886-24e5bb805eed</v002:RequestId>\n          <v002:MessageId>15f46b19-5930-4da8-bb98-9e21e8fdca11</v002:MessageId>\n          </v002:MessageIdentification>\n          <v002:MessageTimestamps>\n          <v002:CreationTimestamp>2024-10-31T09:53:57.487+01:00</v002:CreationTimestamp>\n          </v002:MessageTimestamps>\n<v002:SystemInfo>\n          <v002:Environment>dev</v002:Environment>\n          <v002:Application>\n<v002:ApplicationName>CNP</v002:ApplicationName>\n          </v002:Application>\n          </v002:SystemInfo>\n          </v002:IntegrationHeader>\n<v001:MobilePortInRejectDetails>\n          <v001:MSISDN>+32455009795</v001:MSISDN>\n          <v001:RejectCode>145</v001:RejectCode>\n</v001:MobilePortInRejectDetails>\n</v001:MobilePortInReject>\n",
161 +   "source": "CRDC_BASE",
162 +   "destination": "ARTILIUM",
163 +   "stepName": "REJECT_TO_ARTILIUM",
164 +   "stepStatus": "OK",
165 +   "validation": {},
166 +   "tracking": {
167 +     "requestId": "3865f790-0295-4ed8-9886-24e5bb805eed",
168 +     "messageId": "15f46b19-5930-4da8-bb98-9e21e8fdca11",
169 +     "correlationId": "4db6c409-529a-439d-a710-0140b9198e5f"
170 +   },
171 +   "reason": []
172 + },
173 + ]
174 + }
\\ No newline at end of file

```

Bff-portingrequest-datafeed (coverage)

src/main/java/be/telenet/cnp/data_feed/utils/Constant.java

```
+0 -3 View file @ 621deed7
```

```
1 1 package be.telenet.cnp.data_feed.utils;
2 2
3 3 - import lombok.extern.slf4j.Slf4j;
4 4 -
5 5 - @Slf4j
6 3 public class Constant {
7 4
8 5     public static final String CNP_SERVICE_API_BASE_URL = "cnp_service_api_base_url";
... ...
```

src/test/java/be/telenet/cnp/data_feed/handler/DataFeedProcessorHandlerTest.java

```
+15 -0 View file @ 621deed7
```

```
... ...
1 2 @@ -2,6 +2,7 @@ package be.telenet.cnp.data_feed.handler;
2 2
3 3 import static be.telenet.cnp.data_feed.TestStub.readFileToString;
4 4 import static be.telenet.cnp.data_feed.utils.Mapper.deserialize;
5 5 + import static org.junit.jupiter.api.Assertions.assertNotNull;
6 6
7 7 import be.telenet.cnp.data.model.Event;
8 8 import be.telenet.cnp.data_feed.processor.DataFeedProcessor;
... ...
41 42 @@ -41,4 +42,18 @@ class DataFeedProcessorHandlerTest {
42 43     dataFeedHandler.handleMessage(sqsMessage);
43 44     Mockito.verify(dataFeedProcessor).process(mockEvent);
44 45
45 46     @Test
46 47     void testDefaultConstructor() throws Exception {
47 48         SQSEvent.SQSMessages sqsMessage = new SQSEvent.SQSMessages();
48 49         sqsMessage.setBody(readFileToString(EVENT_JSON));
49 50         DataFeedHandler handler = new DataFeedHandler();
50 51
51 52         assertNotNull(handler, "DataFeedHandler instance should not be null");
52 53
53 54         Event mockEvent = deserialize(sqsMessage.getBody(), Event.class);
54 55
55 56         dataFeedHandler.handleMessage(sqsMessage);
56 57         Mockito.verify(dataFeedProcessor).process(mockEvent);
57 58     }
58 59 }
```

src/test/java/be/telenet/cnp/data_feed/processor/DataFeedProcessorTest.java

```
+14 -4 View file @ 621deed7
```

```
... ...
2 2 @@ -2,13 +2,13 @@ package be.telenet.cnp.data_feed.processor;
3 3 import static be.telenet.cnp.data_feed.TestStub.readFileToString;
4 4 import static be.telenet.cnp.data_feed.utils.Mapper.deserialize;
5 5 + import static org.junit.jupiter.api.Assertions.assertNotNull;
5 6 import static org.mockito.ArgumentMatchers.any;
6 7 - import static org.mockito.Mockito.verify;
7 8 - import static org.mockito.Mockito.when;
7 9 + import static org.mockito.Mockito.*;
8 8
9 9 import be.telenet.cnp.data.client.CnPDataApiClient;
10 10 - import be.telenet.cnp.data.model.Event;
11 11 - import be.telenet.cnp.data.model.PortingRequest;
11 12 + import be.telenet.cnp.data.model.*;
11 13 + import be.telenet.cnp.data_feed.utils.Constant;
12 12
13 13 import java.io.IOException;
14 14 import org.junit.jupiter.api.BeforeEach;
14 14 import org.junit.jupiter.api.Test;
... ...
65 65
66 66     verify(cnPDataApiClient).updatePortingRequestOrder(any());
67 67 }
68 68
69 69     @Test
70 70     void testDefaultConstructor() {
71 71         String mockUrl = "http://mock-cnp-service-api";
72 72         System.setProperty(Constant.CNP_SERVICE_API_BASE_URL, mockUrl);
73 73
74 74         DataFeedProcessor processor = new DataFeedProcessor();
75 75
76 76         assertNotNull(processor, "Processor should not be null");
77 77     }
68 78 }
```

CRDC-Outbound-message-handler (coverage)

src/main/java/be/telenet/cnp/crdc/outbound/message/factory/OutboundMessageFactory.java

```
... ...
5   5   @@ -5,7 +5,11 @@ import be.telenet.cnp.crdc.outbound.message.factory.port_in.NpbroadcastMessageFactory;
6   6   import be.telenet.cnp.crdc.outbound.message.factory.port_in.NpcancelMessageFactory;
7   7   import be.telenet.cnp.crdc.outbound.message.factory.port_in.NpexecMessageFactory;
8 - import be.telenet.cnp.crdc.outbound.message.factory.port_out.*;
9 + import be.telenet.cnp.crdc.outbound.message.factory.port_out.NpacceptMessageFactory;
10  + import be.telenet.cnp.crdc.outbound.message.factory.port_out.NpdoneMessageFactory;
11  + import be.telenet.cnp.crdc.outbound.message.factory.port_out.NprejectMessageFactory;
12  + import be.telenet.cnp.crdc.outbound.message.factory.port_out.NprejectMessageFactory;
13  import be.telenet.cnp.data.model.PortingRequest;
14  import be.telenet.cnp.data.model.Step;
15  import javax.xml.datatype.DatatypeConfigurationException;
... ...
```

src/main/java/be/telenet/cnp/crdc/outbound/service/OutboundMessageProcessor.java

```
... ...
255 255   @@
256 256   private void logFeedbackStep(Step feedbackStep, PortingRequest request) throws IOException, InterruptedException {
257 257     ...
258 258     - if (null != feedbackStep && OK == feedbackStep.getStepStatus()) {
259 259     + if (OK == feedbackStep.getStepStatus()) {
260 260       closeTimerTask(feedbackStep, request);
261 261       if (feedbackStep.getStepName().equalsIgnoreCase("REJECT_FEEDBACK_FROM_CRDC")) {
262 262         updatePortingRequest(feedbackStep);
... ...
```

src/test/java/be/telenet/cnp/crdc/outbound/dispatch/DLQDispatcherTest.java

```
... ...
1  + package be.telenet.cnp.crdc.outbound.dispatch;
2  +
3  + import static org.junit.jupiter.api.Assertions.assertEquals;
4  + import static org.mockito.ArgumentMatchers.any;
5  + import static org.mockito.Mockito.*;
6  +
7  + import be.telenet.cnp.crdc.outbound.config.EnvironmentPropertiesConfig;
8  + import be.telenet.cnp.data.model.Event;
9  + import org.junit.jupiter.api.BeforeEach;
10 + import org.junit.jupiter.api.Test;
11 + import org.junit.jupiter.api.extension.ExtendWith;
12 + import org.mockito.InjectMocks;
13 + import org.mockito.Mock;
14 + import org.mockito.junit.jupiter.MockitoExtension;
15 + import software.amazon.awssdk.services.sqs.SqsClient;
16 + import software.amazon.awssdk.services.sqs.model.SendMessageRequest;
17 + import software.amazon.awssdk.services.sqs.model.SendMessageResponse;
18 +
19 + @ExtendWith(MockitoExtension.class)
20 + class DLQDispatcherTest {
21 +
22 +   @Mock
23 +   private SqsClient client;
24 +
25 +   @Mock
26 +   private EnvironmentPropertiesConfig environmentProperties;
27 +
28 +   @InjectMocks
29 +   private DLQDispatcher dlqDispatcher;
30 +
31 +   private final String testDlqUrl = "https://sqs.test-url.com/test-dlq";
32 +   private final String testEventId = "123";
33 +   private final String testMessageId = "abc-123";
34 +
35 +   private Event testEvent;
36 +
37 +   @BeforeEach
38 +   void setUp() {
39 +     testEvent = Event.builder().build();
40 +     testEvent.setId(testEventId);
41 +   }
42 +
43 +   @Test
44 +   void testDispatch() {
45 +     when(environmentProperties.getDlqUrl()).thenReturn(testDlqUrl);
46 +
47 +     SendMessageResponse mockResponse = mock(SendMessageResponse.class);
48 +     when(mockResponse.getMessageId()).thenReturn(testMessageId);
49 +     when(client.sendMessage(any(SendMessageRequest.class))).thenReturn(mockResponse);
50 +
51 +     String resultMessageId = dlqDispatcher.dispatch(testEvent);
52 +
53 +     assertEquals(testMessageId, resultMessageId, "Returned message ID should match");
54 +
55 +     verify(environmentProperties).getDlqUrl();
56 +     verify(client).sendMessage(any(SendMessageRequest.class));
57 +
58 +     verifyNoMoreInteractions(environmentProperties, client);
59 +   }
60 +
61 + }
```

NCBSS-Outbound-message-handler (coverage)

```

  ✓ src/main/java/be/telenet/cnp/ncbss/outbound/model/OutboundMessageType.java ↗
+6 -6 Viewed ⌂ ⋮
↑ @@ -111,14 +111,14 @@ public enum OutboundMessageType {
111 111     NOK,
112 112     PORTOUT_REJECT_SENT.name());
113 113
114 -     private List<OrderType> orderTypes;
115 -     private List<MessageType> messageType;
116 -     private StepStatus status;
117 -     private String response;
118 +
119 +     private final List<OrderType> orderTypes;
120 +     private final List<MessageType> messageType;
121 +     private final StepStatus status;
122 +     private final String response;
123
124     public static OutboundMessageType getOutboundMessageType(PortingRequest request, Step currentStep) {
125         if (currentStep.getStepName().equalsIgnoreCase("DUPLICATE_CHECKED"))
126             && request.getInternalPorting()
127             if (Boolean.TRUE.equals(
128                 currentStep.getStepName().equalsIgnoreCase("DUPLICATE_CHECKED") && request.getInternalPorting())
129             && OK == currentStep.getStepStatus())
130         return OutboundMessageType.NP_REQUEST_RCV;
131     }
132
133     ↓
  ✓ src/main/java/be/telenet/cnp/ncbss/outbound/processor/OutboundMessageProcessor.java ↗
+19 -10 Viewed ⌂ ⋮
↑ @@ -23,7 +23,13 @@ import static java.util.Collections.reverse;
23 23     import static java.util.UUID.randomUUID;
24 24
25 25     import be.telenet.cnp.data.client.CnpDataApiClient;
26 -     import be.telenet.cnp.data.model.*;
27 +     import be.telenet.cnp.data.model.Event;
28 +     import be.telenet.cnp.data.model.MessageType;
29 +     import be.telenet.cnp.data.model.PortingRequest;
30 +     import be.telenet.cnp.data.model.Step;
31 +     import be.telenet.cnp.data.model.Task;
32 +     import be.telenet.cnp.data.model.TaskStatus;
33 +     import be.telenet.cnp.data.model.Tracking;
34
35     import be.telenet.cnp.ncbss.outbound.model.OutboundMessageType;
36
37     import be.telenet.cnp.ncbss.outbound.model.StepNameTimer;
38     import be.telenet.cnp.ncbss.outbound.sender.MessageSender;
39
40     ↓ @@ -38,13 +44,11 @@ import software.amazon.awssdk.services.sqs.SqsClient;
41     @Slf4j
42     public class OutboundMessageProcessor {
43
44         -     private CnpDataApiClient client;
45         -     private MessageSender messageSender;
46
47         +     private final CnpDataApiClient client;
48         +     private final MessageSender messageSender;
49
50         public static final String STEP_FORMAT = "STEP#%s#%";
51         public static final String AUTH_MANUAL_RESUBMIT = "AUTH_MANUAL_RESUBMIT";
52         public static final String ORDER_CLOSURE_REQUEST = "ORDER_CLOSURE_REQUEST";
53
54         public OutboundMessageProcessor(CnpDataApiClient client) {
55
56             ↓ @@ -93,11 +97,16 @@ public class OutboundMessageProcessor {
57
58                 throws IOException, InterruptedException, JAXBException {
59
60                     var stepName = event.getKey().split("#")[1];
61                     var msisdn = stepName.split("@")[1];
62
63                     -     Step currentStep = portingRequest.getSteps().stream()
64                     +     Optional<Step> optionalCurrentStep = portingRequest.getSteps().stream()
65                         .filter(step -> step.getStepName().equals(DUPLICATE_CHECKED))
66                         .findFirst()
67                         .get();
68
69                     -     processRequest(outboundStep, portingRequest, msisdn, currentStep);
70                         .findFirst();
71
72                     if (optionalCurrentStep.isPresent()) {
73                         Step currentStep = optionalCurrentStep.get();
74                         processRequest(outboundStep, portingRequest, msisdn, currentStep);
75
76                     } else {
77                         throw new IllegalStateException("No step with name DUPLICATE_CHECKED found in the porting request steps.");
78
79                 }
  
```

```

101 110    }
102 111    private String getMsisdn(PortingRequest portingRequest, Step currentStep) {
103 112        @@ -125,7 +134,7 @@ public class OutboundMessageProcessor {
125 134            enrichStepWithOutboundDetails(
126 135                outboundStep, outboundMessageType, outboundMessage, tracking, portingRequest, msisdn, currentStep);
127 136            dispatchOutboundMessage(outboundMessage);
128 137 -        if (isInternalPorting(portingRequest) && null != outboundStep && OK == outboundStep.getStepStatus()) {
129 138 +        if (isInternalPorting(portingRequest) && OK == outboundStep.getStepStatus()) {
130 139            log.info("Internal porting request timer Closure ");
131 140            closeTimerTask(outboundStep, portingRequest);
132 141        }
133 142    }

src/test/java/be/telenet/cnp/ncbss/outbound/handler/NcBssOutboundMessageHandlerTest.java
+9 -0 Viewed ::

17 17    @@ -17,6 +17,7 @@ import com.amazonaws.services.lambda.runtime.tests.annotations.Event;
18 18    import java.io.IOException;
19 19    import javax.xml.bind.JAXBException;
20 20 +    import lombok.SneakyThrows;
21 21    import org.junit.jupiter.api.Test;
22 22    import org.junit.jupiter.api.extension.ExtendWith;
23 23    import org.junit.jupiter.params.ParameterizedTest;
24 24    import org.mockito.InjectMocks;

@@ -74,4 +75,12 @@ class NcBssOutboundMessageHandlerTest {
74 75        SQSEvent.SQSMessages message = event.getRecords().get(0);
75 76        assertThrows(NcBssOutboundHandlerException.class, () -> messageHandler.handleMessage(message));
76 77    }

78 78 +    @Test
79 79 +    void testNcBssOutboundHandlerExceptionWithMessage() {
80 80         String expectedMessage = "An error occurred while processing the message.";
81 81         NcBssOutboundHandlerException exception = new NcBssOutboundHandlerException(expectedMessage);
82 82         assertEquals(expectedMessage, exception.getMessage());
83 83     }
84 84 +    }
85 85 }
86 86 }

src/test/java/be/telenet/cnp/ncbss/outbound/message/factory/impl/PortOutRequestMessageFactoryTest.java
+53 -0 Viewed ::

1 1 + package be.telenet.cnp.ncbss.outbound.message.factory.impl;
2 2 +
3 3 + import static org.junit.jupiter.api.Assertions.*;
4 4 +
5 5 + import be.telenet.cnp.data.model.Donor;
6 6 + import be.telenet.cnp.data.model.PortingDetails;
7 7 + import be.telenet.cnp.data.model.PortingRequest;
8 8 + import be.telenet.cnp.ncbss.outbound.messages.MobilePortOutRequest;
9 9 + import be.telenet.cnp.ncbss.outbound.model.OutboundMessageType;
10 10 + import be.telenet.cnp.ncbss.outbound.util.NcBssOutboundHandlerException;
11 11 + import org.junit.jupiter.api.Test;
12 12 +
13 13 + class PortOutRequestMessageFactoryTest {
14 14 +
15 15     private final PortOutRequestMessageFactory factory = new PortOutRequestMessageFactory();
16 16 +
17 17     @Test
18 18     void testLuhnSimCard() {
19 19         String simCardNumber = "123456789012345678";
20 20         PortingRequest request = createPortingRequest(simCardNumber);
21 21 +
22 22         MobilePortOutRequest result = (MobilePortOutRequest)
23 23             factory.create(OutboundMessageType.NP_REQUEST_RCV, request, null, null, "1234567890");
24 24 +
25 25         assertNotNull(result.getMobilePortOutRequestDetails().getDonorSimCard());
26 26         assertEquals(
27 27             19, result.getMobilePortOutRequestDetails().getDonorSimCard().length());
28 28     }
29 29 +
30 30     @Test
31 31     void testLuhnSimCard_withInvalidCharacter() {
32 32         String invalidSimCard = "12345678901234567X";
33 33         PortingRequest request = createPortingRequest(invalidSimCard);

```

```

35 +         NcBssOutboundHandlerException exception = assertThrows(
36 +             NcBssOutboundHandlerException.class,
37 +             () -> factory.create(OutboundMessageType.NP_REQUEST_RCV, request, null, null, "1234567890"));
38 +         assertTrue(exception.getMessage().contains("is an invalid character"));
39 +     }
40 +
41 +     private PortingRequest createPortingRequest(String simCardNumber) {
42 +         Donor donor = Donor.builder().build();
43 +         donor.setSimCard(simCardNumber);
44 +
45 +         PortingDetails portingDetails = PortingDetails.builder().build();
46 +         portingDetails.setDonor(donor);
47 +
48 +         PortingRequest request = new PortingRequest();
49 +         request.setPortingDetails(portingDetails);
50 +
51 +         return request;
52 +     }
53 + }

```

src/test/java/be/telenet/cnp/ncbss/outbound/processor/OutboundMessageProcessorTest.java

+45 -59 Viewed

↑	@@ -4,9 +4,7 @@ import static org.junit.jupiter.api.Assertions.assertEquals;
4 4	import static org.junit.jupiter.api.Assertions.assertThrows;
5 5	import static org.mockito.ArgumentMatchers.any;
6 6	import static org.mockito.ArgumentMatchers.eq;
7 7	- import static org.mockito.Mockito.times;
8 8	- import static org.mockito.Mockito.verify;
9 9	- import static org.mockito.Mockito.when;
10 10	+ import static org.mockito.Mockito.*;
11 11	import be.telenet.cnp.data.client.CnpDataApiClient;
12 12	import be.telenet.cnp.data.model.*;
↓	@@ -21,6 +19,7 @@ import lombok.SneakyThrows;
21 19	import org.junit.jupiter.api.Test;
22 20	import org.junit.jupiter.api.extension.ExtendWith;
23 21	import org.junit.jupiter.params.ParameterizedTest;
22 22	+ import org.junit.jupiter.params.provider.Arguments;
23 23	import org.junit.jupiter.params.provider.MethodSource;
24 24	import org.mockito.InjectMocks;
25 25	import org.mockito.Mock;
↓ ↑	@@ -66,67 +65,41 @@ class OutboundMessageProcessorTest {
66 65	}
67 66	@SneakyThrows
68 67	@Test
69 -	void testOutboundMessageProcessorForOKFeedbackResponse() {
70 -	Event event = Event.builder()
71 -	.id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11eg")
72 -	.npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ec")
73 -	.pKey("PORT_IN#REQUEST#7ca3e6b6-2fbf-4294-b826-c45e2c0a11eg")
74 -	.sKey("STEP#REQUEST_FEEDBACK_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11eg")
75 -	.build();
76 -	when(client.getPortingRequestById("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ec"))
77 -	.thenReturn(portingRequestStub());
78 -	when(messageSender.sendMessage(any(), eq("ncbss-outbound"))).thenReturn("message-id");
79 -	processor.process(event);
80 -	verifyTest();
81 -	}
82 -	
83 -	@SneakyThrows
84 -	@Test
85 -	void testOutboundMessageProcessorForNOKFeedbackResponse() {
86 -	Event event = Event.builder()
87 -	.id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11eh")
88 -	.npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ec")
89 -	.pKey("PORT_IN#REQUEST#7ca3e6b6-2fbf-4294-b826-c45e2c0a11eh")
90 -	.sKey("STEP#REQUEST_FEEDBACK_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11eh")
91 -	.build();
92 -	when(client.getPortingRequestById("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ec"))
93 -	.thenReturn(portingRequestStub());
94 -	when(messageSender.sendMessage(any(), eq("ncbss-outbound"))).thenReturn("message-id");
95 -	processor.process(event);
96 -	verifyTest();
97 -	}

```

100 -     @SneakyThrows
101 -     @Test
102 -     void testOutboundMessageProcessorForNpAccept() {
103 -         Event event = Event.builder()
104 -             .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
105 -             .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
106 -             .pKey("PORT_IN#READY#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
107 -             .sKey("STEP#ACCEPT_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
108 -             .build();
109 -         when(client.getPortingRequestById("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee"))
110 -             .thenReturn(portingRequestStub());
111 +     @ParameterizedTest
112 +     @MethodSource("provideEventsForProcessing")
113 +     void testOutboundMessageProcessor(Event event) {
114 +         when(client.getPortingRequestById(event.getNpRequestId())).thenReturn(portingRequestStub());
115 +         when(messageSender.sendMessage(any(), eq("ncbss-outbound"))).thenReturn("message-id");
116 +         processor.process(event);
117 +         verifyTest();
118 +     }
119 -     @SneakyThrows
120 -     @Test
121 -     void testOutboundMessageProcessorForNpReject() {
122 -         Event event = Event.builder()
123 -             .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
124 -             .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
125 -             .pKey("PORT_IN#REJECT#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
126 -             .sKey("STEP#REJECT_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
127 -             .build();
128 -         when(client.getPortingRequestById("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee"))
129 -             .thenReturn(portingRequestStub());
130 -         when(messageSender.sendMessage(any(), eq("ncbss-outbound"))).thenReturn("message-id");
131 -         processor.process(event);
132 -         verifyTest();
133 +     private static Stream<Arguments> provideEventsForProcessing() {
134 +         return Stream.of(
135 +             Arguments.of(Event.builder()
136 +                 .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
137 +                 .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
138 +                 .pKey("PORT_IN#REQUEST#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
139 +                 .sKey("STEP#REQUEST_FEEDBACK_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
140 +                 .build()),
141 +             Arguments.of(Event.builder()
142 +                 .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
143 +                 .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
144 +                 .pKey("PORT_IN#REQUEST#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
145 +                 .sKey("STEP#REQUEST_FEEDBACK_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
146 +                 .build()),
147 +             Arguments.of(Event.builder()
148 +                 .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
149 +                 .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
150 +                 .pKey("PORT_IN#READY#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
151 +                 .sKey("STEP#ACCEPT_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
152 +                 .build()),
153 +             Arguments.of(Event.builder()
154 +                 .id("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
155 +                 .npRequestId("7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
156 +                 .pKey("PORT_IN#REJECT#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
157 +                 .sKey("STEP#REJECT_FROM_CRDC#7ca3e6b6-2fbf-4294-b826-c45e2c0a11ee")
158 +                 .build())));
159 +     }
160 +     private void verifyTest() throws IOException, InterruptedException {
161 +         @@ -143,6 +116,19 @@ class OutboundMessageProcessorTest {
162 +             verify(client.createStep(any(Step.class)));
163 +         }
164 +     }
165 +     @Test
166 +     @SneakyThrows
167 +     void testProcessInterruptedException() {
168 +         Event mockEvent = mock(Event.class);
169 +         when(mockEvent.getNpRequestId()).thenReturn("test-id");
170 +         when(client.getPortingRequestById("test-id")).thenThrow(new InterruptedException("Simulated interruption"));
171 +         assertThrows(InterruptedException.class, () -> processor.process(mockEvent));
172 +     }
173 +     verify(client, times(1)).getPortingRequestById("test-id");
174 +     assert Thread.currentThread().isInterrupted();
175 + }
176 +     @SneakyThrows
177 +     @ParameterizedTest
178 +     @MethodSource("portOutEventsProvider")
179 + }
```

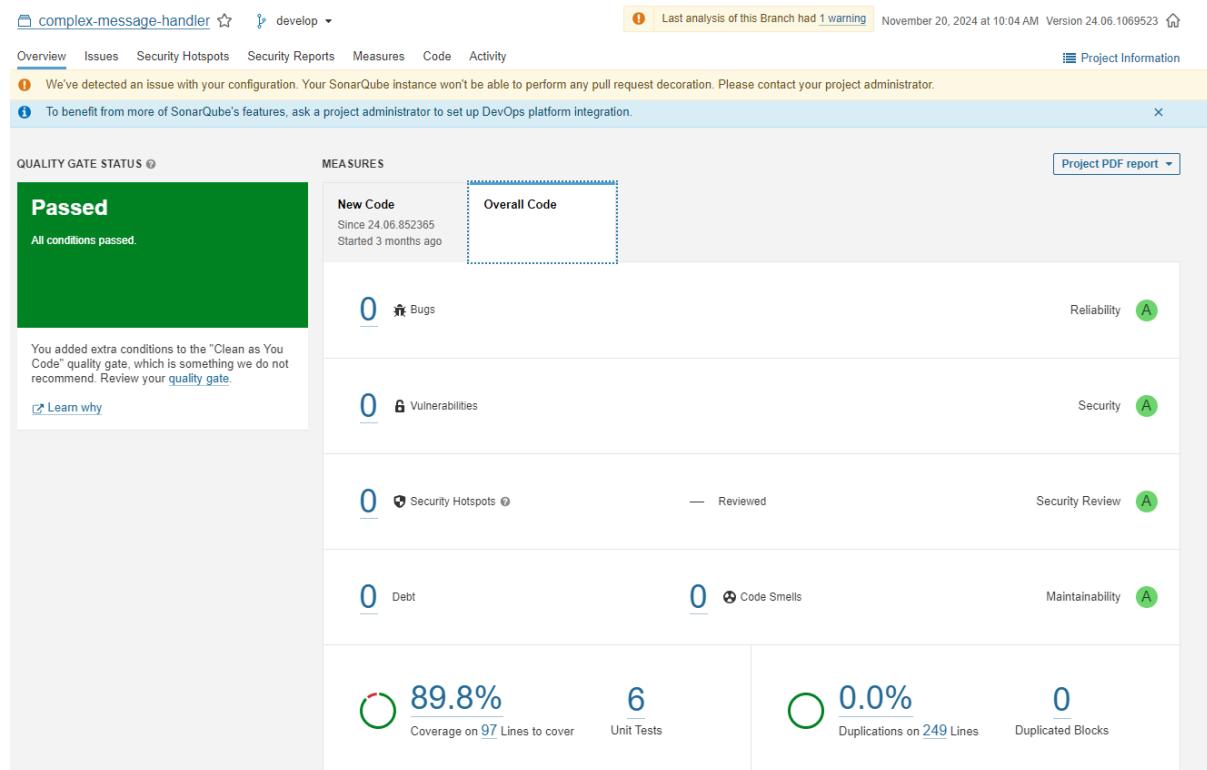
CRDC-Inbound-message-handler (Coverage)

```
src/test/java/be/telenet/cnp/crdc/inbound/handler/InboundMetricPublisherTest.java 0 → 100644 +56 -0 Viewed □ :
```

```
1 + package be.telenet.cnp.crdc.inbound.handler;
2 +
3 + import static org.junit.jupiter.api.Assertions.assertEquals;
4 + import static org.mockito.Mockito.*;
5 +
6 + import be.telenet.cnp.crdc.inbound.utils.SOAPMessageUtils;
7 + import lombok.SneakyThrows;
8 + import org.junit.jupiter.api.BeforeEach;
9 + import org.junit.jupiter.api.Test;
10 + import org.mockito.ArgumentCaptor;
11 + import org.mockito.MockedStatic;
12 + import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
13 + import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
14 +
15 + class InboundMetricPublisherTest {
16 +
17 +     private CloudWatchClient mockCloudWatchClient;
18 +     private InboundMetricPublisher metricPublisher;
19 +
20 +     @BeforeEach
21 +     void setUp() {
22 +         try (MockedStatic<CloudWatchClient> mockedStatic = mockStatic(CloudWatchClient.class)) {
23 +             mockCloudWatchClient = mock(CloudWatchClient.class);
24 +             mockedStatic.when(CloudWatchClient::create).thenReturn(mockCloudWatchClient);
25 +             metricPublisher = new InboundMetricPublisher("TestNamespace");
26 +         }
27 +     }
28 +
29 +     @Test
30 +     @SneakyThrows
31 +     void testPublishMetric_NullRequestBody() {
32 +         metricPublisher.publishMetric(null);
33 +
34 +         verifyNoInteractions(mockCloudWatchClient);
35 +     }
36 +
37 +     @Test
38 +     @SneakyThrows
39 +     void testPublishMetric_ValidRequestBody() {
40 +         String requestBody = "<SOAPMessage></SOAPMessage>";
41 +         try (MockedStatic<SOAPMessageUtils> mockedUtils = mockStatic(SOAPMessageUtils.class)) {
42 +             mockedUtils.when(() -> SOAPMessageUtils.getRootElement(requestBody)).thenReturn("TestMessageType");
43 +
44 +             metricPublisher.publishMetric(requestBody);
45 +
46 +             ArgumentCaptor<PutMetricDataRequest> captor = ArgumentCaptor.forClass(PutMetricDataRequest.class);
47 +             verify(mockCloudWatchClient).putMetricData(captor.capture());
48 +
49 +             PutMetricDataRequest request = captor.getValue();
50 +             assertEquals("TestNamespace", request.namespace());
51 +             assertEquals(1, request.metricData().size());
52 +             assertEquals("TestMessageType", request.metricData().get(0).metricName());
53 +             assertEquals(1.0, request.metricData().get(0).value());
54 +         }
55 +     }
56 + }
```

Complex-message-handler (coverage)

```
src/main/java/be/telenet/cnp/mapper/StepMapper.java
+  private StepMapper() {
+      throw new UnsupportedOperationException("Utility class");
+
public static Step createStep(PortingRequest request, String stepName) {
    StepNameMessageType stepDetails = getStepDetailsByStepName(stepName.substring(stepName.indexOf("_") + 1));
    String actualStepName = extractAfterFirstUnderscore(stepName);
...
src/test/java/be/telenet/cnp/handler/ComplexMessageHandlerTest.java
+  SQSEvent.SQSMessage sqsMessage = new SQSEvent.SQSMessage();
sqsMessage.setBody(readFileToString(EVENT_JSON));
Event mockEvent = deserialize(sqsMessage.getBody(), Event.class);
+
complexMessageHandler.handleMessage(sqsMessage);
ArgumentCaptor<Event> eventCaptor = ArgumentCaptor.forClass(Event.class);
+
ArgumentCaptor<Event> eventCaptor = ArgumentCaptor.forClass(Event.class);
Mockito.verify(complexMessageProcessor).process(eventCaptor.capture());
-
assertEquals(mockEvent, mockEvent, "The captured event should match the mock event");
+
Event capturedEvent = eventCaptor.getValue();
assertEquals(mockEvent, capturedEvent, "The captured event should match the deserialized mock event");
}
}
```



Message transformers

The following development story was my first Fixed Number Porting story. The FNP project needs to be able to transform the messages received by the CRDC, NCOSS & NCBSS. This meant making xsl files to help transform the messages we received. I used unit tests to control if the received messages are transformed correctly.

NCOSS inbound message transformer

The screenshot shows three code editor panes side-by-side, each with a light green background and white text. The top pane contains Java code for a message transformer. The middle and bottom panes contain XSLT code for transforming messages.

Top Pane (Java Code):

```
1 + package be.telenet.message.inbound;
2 +
3 + import be.telenet.cnp.message.handler.MessageHandler;
4 + import java.util.Objects;
5 +
6 + public class FnpNcoossInboundMessageTransformer extends MessageHandler {
7 +
8 +     public FnpNcoossInboundMessageTransformer() {
9 +         super(
10 +             Objects.requireNonNull(
11 +                 FnpNcoossInboundMessageTransformer.class.getResource("/styles/inbound/inbound.transform.xsl")),
12 +             Objects.requireNonNull(System.getenv("OUTBOUND_QUEUE_URL"), "OUTBOUND_QUEUE_URL not provided"));
13 +     }
14 + }
```

Middle Pane (XSLT Code):

```
1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedBroadcastDone/v001">
5 +
6 +   <xsl:template match="ns1:FixedBroadcastDone">
7 +     {
8 +       "orderType": "PORT_ACROSS",
9 +       "messageType": "NP_BROADCAST_DONE",
10 +      "originalMessage": "<xsl:value-of select=\"$input\"/>",
11 +      <xsl:apply-templates/>
12 +    }
13 +   </xsl:template>
14 +
15 +   <xsl:template match="ns1:FixedBroadcastDoneDetails">
16 +     "portingDetails": {
17 +       <xsl:if test="ns1:Range">
18 +         "numberRange": [
19 +           <xsl:for-each select="ns1:Range">
20 +             {
21 +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",

22 +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 +             }<xsl:if test="position() != last()">,</xsl:if>
24 +             <xsl:for-each>
25 +               ]
26 +             </xsl:if>
27 +           }
28 +         </xsl:template>
29 +
30 +   </xsl:stylesheet>
```

Bottom Pane (XSLT Code):

```
1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedDeactivateDone/v001">
5 +
6 +   <xsl:template match="ns1:FixedDeactivateDone">
7 +     {
8 +       "orderType": "DISCONNECT_IN",
9 +       "messageType": "NP_DEACT_DONE",
10 +      "originalMessage": "<xsl:value-of select=\"$input\"/>",
11 +      <xsl:apply-templates/>
12 +    }
13 +   </xsl:template>
14 +
15 +   <xsl:template match="ns1:FixedDeactivateDoneDetails">
16 +     "portingDetails": {
17 +       <xsl:if test="ns1:Range">
18 +         "numberRange": [
19 +           <xsl:for-each select="ns1:Range">
20 +             {
21 +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 +             }<xsl:if test="position() != last()">,</xsl:if>
24 +             <xsl:for-each>
25 +               ]
26 +             </xsl:if>
```

```

27 | +         }
28 | +     </xsl:template>
29 | +
30 | + </xsl:stylesheet>
```

src/main/resources/styles/inbound/FnpDeactivatedOut.xsl 0 → 100644

```

1 | + <?xml version="1.0"?>
2 | + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 | +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 | +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedDeactivatedOut/v001">
5 |
6 | +   <xsl:template match="ns1:FixedDeactivatedOut">
7 | +     {
8 | +       "orderType": "DISCONNECT_OUT",
9 | +       "messageType": "NP_DEACTIVATED",
10 | +       "originalMessage": "<xsl:value-of select='$input' />",
11 | +       <xsl:apply-templates/>
12 | +     }
13 | +   </xsl:template>
14 |
15 | +   <xsl:template match="ns1:FixedDeactivatedOutDetails">
16 | +     "portingDetails": {
17 | +       <xsl:if test="ns1:Range">
18 | +         "numberRange": [
19 | +           <xsl:for-each select="ns1:Range">
20 | +             {
21 | +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 | +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 | +             }<xsl:if test="position() != last()">,</xsl:if>
24 | +           </xsl:for-each>
25 | +         ]
26 | +       </xsl:if>
27 | +     }
28 | +   </xsl:template>
29 |
30 | + </xsl:stylesheet>
```

src/main/resources/styles/inbound/FnpDisconnectInDone.xsl 0 → 100644

```

1 | + <?xml version="1.0"?>
2 | + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 | +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 | +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedDisconnectInDone/v001">
5 |
6 | +   <xsl:template match="ns1:FixedDisconnectInDone">
7 | +     {
8 | +       "orderType": "DISCONNECT_IN",
9 | +       "messageType": "NP_DISC_DONE",
10 | +       "originalMessage": "<xsl:value-of select='$input' />",
11 | +       <xsl:apply-templates/>
12 | +     }
13 | +   </xsl:template>
14 |
15 | +   <xsl:template match="ns1:FixedDisconnectInDoneDetails">
16 | +     "portingDetails": {
17 | +       <xsl:if test="ns1:Range">
18 | +         "numberRange": [
19 | +           <xsl:for-each select="ns1:Range">
20 | +             {
21 | +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 | +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 | +             }<xsl:if test="position() != last()">,</xsl:if>
24 | +           </xsl:for-each>
25 | +         ]
26 | +       </xsl:if>
27 | +     }
28 | +   </xsl:template>
29 |
30 | + </xsl:stylesheet>
```

src/main/resources/styles/inbound/FnpDisconnectOut.xsl 0 → 100644

```

1 | + <?xml version="1.0"?>
2 | + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 | +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 | +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedDisconnectOut/v001">
5 | + 
```

```

6 +     <xsl:template match="ns1:FixedDisconnectOut">
7 +         {
8 +             "orderType": "DISCONNECT_OUT",
9 +             "messageType": "NP_DISCONNECT_OUT",
10 +            "originalMessage": "<xsl:value-of select='$input' />",
11 +            <xsl:apply-templates/>
12 +        }
13 +    </xsl:template>
14 +
15 +    <xsl:template match="ns1:FixedDisconnectOutDetails">
16 +        "portingDetails": {
17 +            <xsl:if test="ns1:Range">
18 +                "numberRange": [
19 +                    <xsl:for-each select="ns1:Range">
20 +                        {
21 +                            "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +                            "numberTo": "<xsl:value-of select='ns1>LastPhoneNumber' />"
23 +                        }<xsl:if test="position() != last()">,</xsl:if>
24 +                    </xsl:for-each>
25 +                ],
26 +            </xsl:if>
27 +            <xsl:if test="ns1:RecipientServiceProviderCode">
28 +                "recipientId": "<xsl:value-of select='ns1:RecipientServiceProviderCode' />"
29 +            </xsl:if>
30 +        }
31 +    </xsl:template>
32 +
33 + </xsl:stylesheet>
```

▼ src/main/resources/styles/inbound/FnpPortInExec.xsl 0 → 100644

+33 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +           xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +           xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInExec/v001">
5 +
6 +     <xsl:template match="ns1:FixedPortInExec">
7 +         {
```

8 + "orderType": "PORT_IN",
9 + "messageType": "NP_EXEC",
10 + "originalMessage": "<xsl:value-of select='\$input' />",
11 + <xsl:apply-templates/>
12 + }
13 + </xsl:template>
14 +
15 + <xsl:template match="ns1:FixedPortInExecDetails">
16 + "portingDetails": {
17 + <xsl:if test="ns1:Range">
18 + "numberRange": [
19 + <xsl:for-each select="ns1:Range">
20 + {
21 + "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 + "numberTo": "<xsl:value-of select='ns1>LastPhoneNumber' />"
23 + }<xsl:if test="position() != last()">,</xsl:if>
24 + </xsl:for-each>
25 +],
26 + </xsl:if>
27 + <xsl:if test="ns1:RoutingNumber">
28 + "routingNumber": "<xsl:value-of select='ns1:RoutingNumber' />"
29 + </xsl:if>
30 + }
31 + </xsl:template>
32 +
33 + </xsl:stylesheet>

▼ src/main/resources/styles/inbound/FnpPortInNonRFS.xsl 0 → 100644

+30 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +           xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +           xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInNonRFS/v001">
5 +
6 +     <xsl:template match="ns1:FixedPortInNonRFS">
7 +         {
8 +             "orderType": "PORT_IN",
9 +             "messageType": "NP_NON_RFS",
10 +            "originalMessage": "<xsl:value-of select='$input' />",
11 +        }
```

```

11 +         <xsl:apply-templates/>
12 +
13 +     </xsl:template>
14 +
15 +     <xsl:template match="ns1:FixedPortInNonRFSDetails">
16 +         "portingDetails": {
17 +             <xsl:if test="ns1:Range">
18 +                 "numberRange": [
19 +                     <xsl:for-each select="ns1:Range">
20 +                         {
21 +                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +                             "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />" 
23 +                         }<xsl:if test="position() != last()">,</xsl:if>
24 +                     </xsl:for-each>
25 +                 ]
26 +             </xsl:if>
27 +         }
28 +     </xsl:template>
29 +
30 + </xsl:stylesheet>

```

✓ src/main/resources/styles/inbound/FnpPortInRFS.xsl 0 → 100644

+30 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +     xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +     xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInRFS/v001">
5 +
6 +     <xsl:template match="ns1:FixedPortInRFS">
7 +         {
8 +             "orderType": "PORT_IN",
9 +             "messageType": "NP_RFS",
10 +             "originalMessage": "<xsl:value-of select='$input' />",
11 +             <xsl:apply-templates/>
12 +         }
13 +     </xsl:template>
14 +
15 +     <xsl:template match="ns1:FixedPortInRFSDetails">
16 +         "portingDetails": {
17 +             <xsl:if test="ns1:Range">
18 +                 "numberRange": [
19 +                     <xsl:for-each select="ns1:Range">
20 +                         {
21 +                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +                             "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />" 
23 +                         }<xsl:if test="position() != last()">,</xsl:if>
24 +                     </xsl:for-each>
25 +                 ]
26 +             </xsl:if>
27 +         }
28 +     </xsl:template>
29 +
30 + </xsl:stylesheet>

```

✓ src/main/resources/styles/inbound/FnpPortOutReady.xsl 0 → 100644

+30 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +     xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +     xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortOutReady/v001">
5 +
6 +     <xsl:template match="ns1:FixedPortOutReady">
7 +         {
8 +             "orderType": "PORT_OUT",
9 +             "messageType": "NP_READY",
10 +             "originalMessage": "<xsl:value-of select='$input' />",
11 +             <xsl:apply-templates/>
12 +         }
13 +     </xsl:template>
14 +
15 +     <xsl:template match="ns1:FixedPortOutReadyDetails">
16 +         "portingDetails": {
17 +             <xsl:if test="ns1:Range">
18 +                 "numberRange": [
19 +                     <xsl:for-each select="ns1:Range">
20 +                         {

```

```

21 +             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +             "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 +         }<xsl:if test="position()!<last()">,</xsl:if>
24 +         </xsl:for-each>
25 +     ]
26 +   </xsl:if>
27 + }
28 + </xsl:template>
29 +
30 + </xsl:stylesheet>

```

src/main/resources/styles/inbound/FnpUpdateInCompleted.xsl 0 → 100644 +30 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedUpdateInCompleted/v001">
5 +
6 +   <xsl:template match="ns1:FixedUpdateInCompleted">
7 +     {
8 +       "orderType": "UPDATE_IN",
9 +       "messageType": "NP_UPDATE_COMPLETED",
10 +      "originalMessage": "<xsl:value-of select='$input' />",
11 +      <xsl:apply-templates/>
12 +    }
13 +   </xsl:template>
14 +
15 +   <xsl:template match="ns1:FixedUpdateInCompletedDetails">
16 +     "portingDetails": {
17 +       <xsl:if test="ns1:Range">
18 +         "numberRange": [
19 +           <xsl:for-each select="ns1:Range">
20 +             {
21 +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 +             }<xsl:if test="position()!<last()">,</xsl:if>
24 +           </xsl:for-each>
25 +         ]
26 +         </xsl:if>
27 +       }
28 +     </xsl:template>
29 +
30 + </xsl:stylesheet>

```

src/main/resources/styles/inbound/FnpUpdateOut.xsl 0 → 100644 +33 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedUpdateOut/v001">
5 +
6 +   <xsl:template match="ns1:FixedUpdateOut">
7 +     {
8 +       "orderType": "UPDATE_OUT",
9 +       "messageType": "NP_UPDATE",
10 +      "originalMessage": "<xsl:value-of select='$input' />",
11 +      <xsl:apply-templates/>
12 +    }
13 +   </xsl:template>
14 +
15 +   <xsl:template match="ns1:FixedUpdateOutDetails">
16 +     "portingDetails": {
17 +       <xsl:if test="ns1:Range">
18 +         "numberRange": [
19 +           <xsl:for-each select="ns1:Range">
20 +             {
21 +               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
22 +               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
23 +             }<xsl:if test="position()!<last()">,</xsl:if>
24 +           </xsl:for-each>
25 +         ],
26 +         </xsl:if>
27 +         <xsl:if test="ns1:RoutingNumber">
28 +           "routingNumber": "<xsl:value-of select='ns1:RoutingNumber' />"
29 +         </xsl:if>
30 +       }

```

33 + </xsl:stylesheet>

src/main/resources/styles/inbound/IntegrationHeader.xsl 0 → 100644 +23 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0"
3 +   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002">
5 +
6 +   <xsl:template match="ns2:IntegrationHeader" name="integration-header">
7 +     <xsl:apply-templates/>
8 +   </xsl:template>
9 +
10 +   <xsl:template match="ns2:SystemInfo" name="system-info">
11 +     ,"initiatedBy" : "<xsl:value-of select='ns2:Application/ns2:ApplicationName' />" 
12 +   </xsl:template>
13 +
14 +   <xsl:template match="ns2:MessageIdentification" name="message-identification">
15 +     ,"tracking" : {
16 +       "correlationId" : "<xsl:value-of select='ns2:CorrelationId' />",
17 +       "requestId" : "<xsl:value-of select='ns2:RequestId' />",
18 +       "messageId" : "<xsl:value-of select='ns2:MessageId' />"
19 +     }
20 +   </xsl:template>
21 +
22 + </xsl:stylesheet>
23 +

```

src/main/resources/styles/inbound/inbound.transform.xsl 0 → 100644 +26 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0"
3 +   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4 +
5 +   <xsl:import href="FnpDisconnectInDone.xsl"/>
6 +   <xsl:import href="FnpDeactivateDone.xsl"/>
7 +   <xsl:import href="FnpDisconnectOut.xsl"/>
8 +   <xsl:import href="FnpDeactivatedOut.xsl"/>
9 +   <xsl:import href="FnpBroadcastDone.xsl"/>
10 +  <xsl:import href="FnpPortInExec.xsl"/>
11 +  <xsl:import href="FnpPortInRFS.xsl"/>
12 +  <xsl:import href="FnpPortInNonRFS.xsl"/>
13 +  <xsl:import href="FnpPortOutReady.xsl"/>
14 +  <xsl:import href="FnpUpdateInCompleted.xsl"/>
15 +  <xsl:import href="FnpUpdateOut.xsl"/>
16 +
17 +  <xsl:output method="text" indent="yes"/>
18 +  <xsl:param name="input"/>
19 +  <xsl:param name="initiatedBy"/>
20 +  <xsl:template match="text()"/>
21 +
22 +  <xsl:template match="/">
23 +    <xsl:apply-templates/>
24 +  </xsl:template>
25 +
26 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/logback.xml 0 → 100644 +33 -0 Viewed

```

1 +
2 + <configuration>
3 +   <property name="LOG_LEVEL" value="${LOG_LEVEL:-info}" />
4 +   <property name="LOG_LEVEL_ROOT" value="${LOG_LEVEL_ROOT:-error}" />
5 +
6 +   <!-- Console Appender -->
7 +   <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
8 +     <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
9 +   </appender>
10 +
11 +   <!-- Lambda Appender -->
12 +   <appender name="lambda" class="ch.qos.logback.core.ConsoleAppender">
13 +     <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
14 +     <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
15 +       <level>${LAMBDA_LOG_LEVEL:-info}</level>
16 +     </filter>

```

```

17 | +     </appender>
18 |
19 | +     <!-- Root Logger -->
20 | +     <root level="${LOG_LEVEL_ROOT}">
21 | +         <appender-ref ref="console"/>
22 | +     </root>
23 |
24 | +     <!-- Application Logger -->
25 | +     <logger name="be.telenet" level="${LOG_LEVEL}" additivity="false">
26 | +         <appender-ref ref="console"/>
27 | +     </logger>
28 |
29 | +     <!-- Lambda Logger -->
30 | +     <logger name="com.amazonaws.services.lambda" level="${LOG_LEVEL:-info}" additivity="false">
31 | +         <appender-ref ref="lambda"/>
32 | +     </logger>
33 | + </configuration>
34 | \\
35 | \\ No newline at end of file

```

src/main/resources/transformer.properties 0 → 100644 +0 -0 Viewed

src/test/java/be/telenet/message/inbound/FnpNcossInboundMessageTransformerTest.java 0 → 100644 +23 -0 Viewed

```

1 | + package be.telenet.message.inbound;
2 |
3 | + import static org.junit.jupiter.api.Assertions.assertNotNull;
4 |
5 | + import org.junit.jupiter.api.Test;
6 | + import org.junit.jupiter.api.extension.ExtendWith;
7 | + import org.mockito.junit.jupiter.MockitoExtension;
8 | + import uk.org.webcompere.systemstubs.SystemStubs;
9 | + import uk.org.webcompere.systemstubs.jupiter.SystemStubsExtension;
10 |
11 | + @ExtendWith({MockitoExtension.class, SystemStubsExtension.class})
12 | + class FnpNcossInboundMessageTransformerTest {
13 |
14 | +     @Test
15 | +     void testConstructor() throws Exception {
16 | +         SystemStubs.withEnvironmentVariable("OUTBOUND_QUEUE_URL", "dummy_url")
17 | +             .and("AWS_REGION", "us-east-1")
18 | +             .execute(() -> {
19 | +                 FnpNcossInboundMessageTransformer transformer = new FnpNcossInboundMessageTransformer();
20 | +                 assertNotNull(transformer);
21 | +             });
22 | +     }
23 | + }

```

.gitignore 0 → 100644 +33 -0 Viewed

```

1 | + HELP.md
2 | + target/
3 | + !.mvn/wrapper/maven-wrapper.jar
4 | + !**/src/main/**/target/
5 | + !**/src/test/**/target/
6 |
7 | + ### STS ###
8 | + .apt_generated
9 | + .classpath
10 | + .factorypath
11 | + .project
12 | + .settings
13 | + .springBeans
14 | + .sts4-cache
15 |
16 | + ### IntelliJ IDEA ###
17 | + .idea
18 | + *.iws
19 | + *.iml
20 | + *.ipr
21 |
22 | + ### NetBeans ###
23 | + /nbproject/private/
24 | + /nbbuild/
25 | + /dist/

```

```

26 + /nbdist/
27 + /.nb-gradle/
28 + build/
29 + !**/src/main/**/build/
30 + !**/src/test/**/build/
31 +
32 + ### VS Code ###
33 + .vscode/

```

```

▼ .gitlab-ci.yml 📄 0 → 100644
  1 + variables:
  2 +   CI_PROJECT_NAME: "fnp-ncoss-message-transformer"
  3 +
  4 + include:
  5 +   - project: 'tif/number-management/pipeline'
  6 +     file: '.gitlab-ci-application-fnp.yml'
    \ No newline at end of file

```

```

▼ Dockerfile 📄 0 → 100644
  1 + FROM public.ecr.aws/lambda/java:17
  2 + COPY target/fnp-ncoss-message-transformer*.jar ${LAMBDA_TASK_ROOT}/lib/

```

```

▼ lombok.config 📄 0 → 100644
  1 + lombok.addLombokGeneratedAnnotation = true
    \ No newline at end of file

```

```

▼ pom.xml 📄 0 → 100644
  1 + <?xml version="1.0" encoding="UTF-8"?>
  2 + <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  3 +   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  4 +
  5 +   <parent>
  6 +     <groupId>be.telenet.cnp</groupId>
  7 +     <artifactId>cnp-lambda-parent</artifactId>
    <version>1.0.299795</version>
  9 +   </parent>
 10 +
 11 +   <groupId>be.telenet</groupId>
 12 +   <artifactId>fnp-ncoss-message-transformer</artifactId>
 13 +   <version>24.06.0-SNAPSHOT</version>
 14 +
 15 +   <properties>
 16 +     <maven.compiler.source>17</maven.compiler.source>
 17 +     <maven.compiler.target>17</maven.compiler.target>
 18 +   </properties>
 19 +
 20 +   <dependencies>
 21 +     <dependency>
 22 +       <groupId>be.telenet.cnp</groupId>
 23 +       <artifactId>lib.cnp-message-transformer</artifactId>
 24 +       <version>24.06.552413</version>
 25 +     </dependency>
 26 +     <dependency>
 27 +       <groupId>com.amazonaws</groupId>
 28 +       <artifactId>aws-java-sdk-core</artifactId>
 29 +     </dependency>
 30 +     <dependency>
 31 +       <groupId>com.amazonaws</groupId>
 32 +       <artifactId>aws-java-sdk-sqs</artifactId>
 33 +     </dependency>
 34 +     <dependency>
 35 +       <groupId>com.amazonaws</groupId>
 36 +       <artifactId>aws-lambda-java-core</artifactId>
 37 +       <version>1.2.2</version>
 38 +     </dependency>
 39 +     <dependency>
 40 +       <groupId>com.amazonaws</groupId>
 41 +       <artifactId>aws-lambda-java-events</artifactId>
 42 +       <version>3.11.1</version>
 43 +     </dependency>
 44 +     <!-- Testing -->
 45 +   <dependency>

```

```

46 +         <groupId>org.junit.jupiter</groupId>
47 +         <artifactId>junit-jupiter-engine</artifactId>
48 +         <scope>test</scope>
49 +     </dependency>
50 +     <dependency>
51 +         <groupId>org.junit.jupiter</groupId>
52 +         <artifactId>junit-jupiter-params</artifactId>
53 +         <scope>test</scope>
54 +     </dependency>
55 +     <dependency>
56 +         <groupId>com.amazonaws</groupId>
57 +         <artifactId>aws-lambda-java-tests</artifactId>
58 +         <scope>test</scope>
59 +     </dependency>
60 +     <dependency>
61 +         <groupId>org.mockito</groupId>
62 +         <artifactId>mockito-core</artifactId>
63 +         <scope>test</scope>
64 +     </dependency>
65 +     <dependency>
66 +         <groupId>org.mockito</groupId>
67 +         <artifactId>mockito-junit-jupiter</artifactId>
68 +         <scope>test</scope>
69 +     </dependency>
70 +     <dependency>
71 +         <groupId>uk.org.webcompere</groupId>
72 +         <artifactId>system-stubs-core</artifactId>
73 +         <version>2.0.2</version>
74 +         <scope>test</scope>
75 +     </dependency>
76 +     <dependency>
77 +         <groupId>uk.org.webcompere</groupId>
78 +         <artifactId>system-stubs-jupiter</artifactId>
79 +         <version>2.0.2</version>
80 +         <scope>test</scope>
81 +     </dependency>
82 +
83 +   </dependencies>
84 +
85 +   <profiles>
86 +     <profile>
87 +       <id>coverage</id>
88 +       <build>
89 +         <plugins>
90 +           <plugin>
91 +             <groupId>org.jacoco</groupId>
92 +             <artifactId>jacoco-maven-plugin</artifactId>
93 +             <version>${jacoco.maven.version}</version>
94 +             <executions>
95 +               <execution>
96 +                 <id>report</id>
97 +                 <goals>
98 +                   <goal>report</goal>
99 +                 </goals>
100 +                 <phase>verify</phase>
101 +               </execution>
102 +             </executions>
103 +           </plugin>
104 +         </plugins>
105 +       </build>
106 +     </profile>
107 +   </profiles>
108 +
109 + </project>

```

✓ settings.xml 0 → 100644 +94 -0 Viewed

```

1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
3 +   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 +   xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 https://maven.apache.org/xsd/settings-1.0.0.xsd">
5 +
6 +   <servers>
7 +     <server>
8 +       <id>telnet.technicalorderfulfilment_releases</id>
9 +       <username>deployment</username>
10 +      <password>nxsdp1rel83</password>

```

```

11 +         </server>
12 +
13 +         <server>
14 +             <id>telenet.technicalorderfulfilment_snapshots</id>
15 +             <username>deployment</username>
16 +             <password>nxsdp1rel83</password>
17 +         </server>
18 +     </servers>
19 +
20 +     <mirrors>
21 +         <mirror>
22 +             <id>telenet.mirror</id>
23 +             <name>telenet.mirror</name>
24 +             <mirrorOf>*,nexus-tn2-public</mirrorOf>
25 +             <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public</url>
26 +         </mirror>
27 +     </mirrors>
28 +
29 +     <profiles>
30 +         <profile>
31 +             <id>sonar</id>
32 +             <activation>
33 +                 <activeByDefault>true</activeByDefault>
34 +             </activation>
35 +             <properties>
36 +                 <sonar.host.url>
37 +                     https://sonarqube.cmt.apps.telenet.be/
38 +                 </sonar.host.url>
39 +             </properties>
40 +         </profile>
41 +         <profile>
42 +             <id>nexus-tn2</id>
43 +             <activation>
44 +                 <activeByDefault>true</activeByDefault>
45 +             </activation>
46 +             <repositories>
47 +                 <repository>
48 +                     <id>nexus-tn2-public</id>
49 +                     <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public</url>
50 +
51 +                     <releases>
52 +                         <enabled>true</enabled>
53 +                         <checksumPolicy>warn</checksumPolicy>
54 +                     </releases>
55 +                     <snapshots>
56 +                         <enabled>false</enabled>
57 +                         <checksumPolicy>warn</checksumPolicy>
58 +                     </snapshots>
59 +                 </repository>
60 +             <repository>
61 +                 <id>telenet.technicalorderfulfilment_releases</id>
62 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_releases</url>
63 +                 <releases><enabled>true</enabled></releases>
64 +                 <snapshots><enabled>false</enabled></snapshots>
65 +             </repository>
66 +             <repository>
67 +                 <id>telenet.technicalorderfulfilment_snapshots</id>
68 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_snapshots</url>
69 +                 <releases><enabled>false</enabled></releases>
70 +                 <snapshots><enabled>true</enabled></snapshots>
71 +
72 +             <pluginRepositories>
73 +                 <pluginRepository>
74 +                     <id>nexus-tn2-public</id>
75 +                     <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public</url>
76 +                     <releases><enabled>true</enabled></releases>
77 +                     <snapshots><enabled>false</enabled></snapshots>
78 +                 </pluginRepository>
79 +                 <pluginRepository>
80 +                     <id>telenet.technicalorderfulfilment_releases</id>
81 +                     <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_releases</url>
82 +                     <releases><enabled>true</enabled></releases>
83 +                     <snapshots><enabled>false</enabled></snapshots>
84 +                 </pluginRepository>
85 +                 <pluginRepository>
86 +                     <id>telenet.technicalorderfulfilment_snapshots</id>

```

NCBSS inbound message transformer

```
✓ src/main/java/be/telenet/message/inbound/FnpNbssInboundMessageTransformer.java 0 → 100644 +14 -0 Viewed □ ⋮
1 + package be.telenet.message.inbound;
2 +
3 + import be.telenet.cnp.message.handler.MessageHandler;
4 + import java.util.Objects;
5 +
6 + public class FnpNbssInboundMessageTransformer extends MessageHandler {
7 +
8 +     public FnpNbssInboundMessageTransformer() {
9 +         super(
10 +             Objects.requireNonNull(
11 +                 FnpNbssInboundMessageTransformer.class.getResource("/styles/inbound/inbound.transform.xsl")),
12 +             Objects.requireNonNull(System.getenv("OUTBOUND_QUEUE_URL"), "OUTBOUND_QUEUE_URL not provided")));
13 +     }
14 + }
```

```
✓ src/main/resources/styles/inbound/FnpPortInCancel.xsl 0 → 100644 +31 -0 Viewed □ ⋮
1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInCancel/v001"
5 +   xmlns:ns3="http://xmlns.telenet.be/integration/model/PhoneNumberIdentifiers/v003">
6 +
7 +     <xsl:template match="ns1:FixedPortInCancel">
8 +         {
9 +             "orderType": "PORT_IN",
10 +             "messageType": "NP_CANCEL",
11 +             "originalMessage": "<xsl:value-of select='$input' />",
12 +             <xsl:apply-templates/>
13 +         }
14 +     </xsl:template>
15 +
16 +     <xsl:template match="ns1:FixedPortInCancelDetails">
17 +         "portingDetails": {
18 +             <xsl:if test="ns1:Range">
19 +                 "numberRange": [
20 +                     <xsl:for-each select="ns1:Range">
21 +                         {
22 +                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
23 +                             "numberTo": "<xsl:value-of select='ns1>LastPhoneNumber' />"}
24 +                         <xsl:if test="position() != last()">, </xsl:if>
25 +                         </xsl:for-each>
26 +                     ]
27 +                 </xsl:if>
28 +             }
29 +         </xsl:template>
30 +
31 +     </xsl:stylesheet>
32 + \ No newline at end of file
```

```
✓ src/main/resources/styles/inbound/FnpPortInRequest.xsl 0 → 100644 +56 -0 Viewed □ ⋮
1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 +   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 +   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInRequest/v001"
5 +   xmlns:ns3="http://xmlns.telenet.be/integration/model/PhoneNumberIdentifiers/v003">
6 +
7 +     <xsl:template match="ns1:FixedPortInRequest">
8 +         {
9 +             "orderType": "PORT_IN",
10 +             "messageType": "NP_REQUEST",
11 +             "originalMessage": "<xsl:value-of select='$input' />",
12 +             <xsl:apply-templates/>
13 +         }
14 +     </xsl:template>
15 +
16 +     <xsl:template match="ns1:FixedPortInRequestDetails">
17 +         "portingDetails": {
18 +             <xsl:if test="ns1:Range">
19 +                 "numberRange": [
20 +                     <xsl:for-each select="ns1:Range">
21 +                         {
22 +                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
23 +                             "numberTo": "<xsl:value-of select='ns1>LastPhoneNumber' />"}
24 +                         <xsl:if test="position() != last()">, </xsl:if>
```

```

25+           </xsl:for-each>
26+           ],
27+         <xsl:if>
28+           "dueDate": "<xsl:value-of select='ns1:DueDate' />",
29+           "recipientId": "<xsl:value-of select='ns1:RecipientServiceProviderCode' />",
30+           "addressInfo": {
31+             <xsl:choose>
32+               <xsl:when test="ns1:InstallationAddress/ns1:AddressId">
33+                 "addressId": "<xsl:value-of select='ns1:InstallationAddress/ns1:AddressId' />"
34+               </xsl:when>
35+               <xsl:when test="ns1:InstallationAddress/ns1:AddressInfo">
36+                 "street": "<xsl:value-of select='ns1:InstallationAddress/ns1:AddressInfo/ns1:Street' />",
37+                 "houseNumber": "<xsl:value-of select='ns1:InstallationAddress/ns1:AddressInfo/ns1:HouseNumber' />",
38+                 "zipCode": "<xsl:value-of select='ns1:InstallationAddress/ns1:AddressInfo/ns1:ZipCode' />"
39+               </xsl:when>
40+             </xsl:choose>
41+           },
42+           <xsl:if test="ns1:VAT">
43+             "vat": "<xsl:value-of select='ns1:VAT' />",
44+           </xsl:if>
45+           "numberCount": <xsl:value-of select="ns1:NumberCount"/>,
46+           "complexityClass": "<xsl:value-of select='ns1:ComplexityClass' />"
47+           <xsl:if test="ns1:AccountNumber">
48+             "accountNumber": "<xsl:value-of select='ns1:AccountNumber' />"
49+           </xsl:if>
50+           <xsl:if test="ns1:CodedId">
51+             "codedId": "<xsl:value-of select='ns1:CodedId' />"
52+           </xsl:if>
53+         }
54+       </xsl:template>
55+
56+     </xsl:stylesheet>
      \ No newline at end of file

```

src/main/resources/styles/inbound/FnpPortOutAccept.xsl 0 → 100644 +32 -0 Viewed

```

1+ <?xml version="1.0"?>
2+ <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3+   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4+   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortOutAuthorizeAccept/v001"
5+   xmlns:ns3="http://xmlns.telenet.be/integration/model/PhoneNumberIdentifiers/v003">
6+
7+   <xsl:template match="ns1:FixedPortOutAuthorizeAccept">
8+     {
9+       "orderType": "PORT_OUT",
10+      "messageType": "NP_ACCEPT",
11+      "originalMessage": "<xsl:value-of select='$input' />",
12+      <xsl:apply-templates/>
13+    }
14+   </xsl:template>
15+
16+   <xsl:template match="ns1:FixedPortOutAuthorizeAcceptDetails">
17+     "portingDetails": {
18+       <xsl:if test="ns1:Range">
19+         "numberRange": [
20+           <xsl:for-each select="ns1:Range">
21+             {
22+               "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
23+               "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"><xsl:if test="position()=last()">,</xsl:if>
24+             </xsl:for-each>
25+           ],
26+         </xsl:if>
27+         "complexityClass": "<xsl:value-of select='ns1:ComplexityClass' />"
28+       }
29+     }
30+   </xsl:template>
31+
32+ </xsl:stylesheet>
      \ No newline at end of file

```

src/main/resources/styles/inbound/FnpPortOutAuthorizeManual.xsl 0 → 100644 +34 -0 Viewed

```

1+ <?xml version="1.0"?>
2+ <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3+   xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4+   xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortOutAuthorizeManual/v001">

```

```

5 |+         xmlns:ns3="http://xmlns.telenet.be/integration/model/PhoneNumberIdentifiers/v003">
6 |
7 |+     <xsl:template match="ns1:FixedPortOutAuthorizeManual">
8 |+         {
9 |+             "orderType": "PORT_OUT",
10 |+             "messageType": "NP_AUTHORIZE_MAN",
11 |+             "originalMessage": "<xsl:value-of select='$input' />",
12 |+             <xsl:apply-templates/>
13 |+         }
14 |+     </xsl:template>
15 |
16 |+     <xsl:template match="ns1:FixedPortOutAuthorizeManualDetails">
17 |+         "PortingDetails": {
18 |+             <xsl:if test="ns1:Range">
19 |+                 "numberRange": [
20 |+                     <xsl:for-each select="ns1:Range">
21 |+                         {
22 |+                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
23 |+                             "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
24 |+                         }<xsl:if test="position() != last()">,</xsl:if>
25 |+                     </xsl:for-each>
26 |+                 ],
27 |+             </xsl:if>
28 |+             "rejectReason": {
29 |+                 "rejectReasonCode": <xsl:value-of select="ns1:RejectReasons/ns1:RejectReason"/>
30 |+             }
31 |+         }
32 |+     </xsl:template>
33 |
34 |+ </xsl:stylesheet>

```

✓ src/main/resources/styles/inbound/FnpPortOutReject.xsl 0 → 100644 +36 -0 Viewed

```

1 |+ <?xml version="1.0"?>
2 |+ <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3 |+             xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002"
4 |+             xmlns:ns1="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortOutAuthorizeReject/v001"
5 |+             xmlns:ns3="http://xmlns.telenet.be/integration/model/PhoneNumberIdentifiers/v003">
6 |
7 |+     <xsl:template match="ns1:FixedPortOutAuthorizeReject">
8 |+         {
9 |+             "orderType": "PORT_OUT",
10 |+             "messageType": "NP_REJECT",
11 |+             "originalMessage": "<xsl:value-of select='$input' />",
12 |+             <xsl:apply-templates/>
13 |+         }
14 |+     </xsl:template>
15 |
16 |+     <xsl:template match="ns1:FixedPortOutAuthorizeRejectDetails">
17 |+         "portingDetails": {
18 |+             <xsl:if test="ns1:Range">
19 |+                 "numberRange": [
20 |+                     <xsl:for-each select="ns1:Range">
21 |+                         {
22 |+                             "numberFrom": "<xsl:value-of select='ns1:FirstPhoneNumber' />",
23 |+                             "numberTo": "<xsl:value-of select='ns1:LastPhoneNumber' />"
24 |+                         }<xsl:if test="position() != last()">,</xsl:if>
25 |+                     </xsl:for-each>
26 |+                 ],
27 |+             </xsl:if>
28 |+             "complexityClass": "<xsl:value-of select='ns1:ComplexityClass' />",
29 |+             "rejectInfo": {
30 |+                 "rejectCode": "<xsl:value-of select='ns1:PortOutRejectInfo/PortOutRejectCode' />",
31 |+                 "rejectDescription": "<xsl:value-of select='ns1:PortOutRejectInfo/PortOutRejectDescription' />"
32 |+             }
33 |+         }
34 |+     </xsl:template>
35 |
36 |+ </xsl:stylesheet>

```

\ No newline at end of file

✓ src/main/resources/styles/inbound/IntegrationHeader.xsl 0 → 100644 +22 -0 Viewed

```

1 |+ <?xml version="1.0"?>
2 |+ <xsl:stylesheet version="3.0"
3 |+             xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4 |+             xmlns:ns2="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002">

```

```

6 +     <xsl:template match="ns2:IntegrationHeader" name="integration-header">
7 +         <xsl:apply-templates/>
8 +     </xsl:template>
9 +
10 +    <xsl:template match="ns2:SystemInfo" name="system-info">
11 +        , "initiatedBy" : "<xsl:value-of select="ns2:Application/ns2:ApplicationName"/>""
12 +    </xsl:template>
13 +
14 +    <xsl:template match="ns2:MessageIdentification" name="message-identification">
15 +        , "tracking" : {
16 +            "correlationId" : "<xsl:value-of select="ns2:CorrelationId"/>",
17 +            "requestId" : "<xsl:value-of select="ns2:RequestId"/>",
18 +            "messageId" : "<xsl:value-of select="ns2:MessageId"/>"
19 +        }
20 +    </xsl:template>
21 +
22 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/inbound.transform.xsl 0 → 100644

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0"
3 +   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4 +
5 +   <xsl:import href="FnPortInRequest.xsl"/>
6 +   <xsl:import href="FnPortInCancel.xsl"/>
7 +   <xsl:import href="FnPortOutAccept.xsl"/>
8 +   <xsl:import href="FnPortOutReject.xsl"/>
9 +   <xsl:import href="FnPortOutAuthorizeManual.xsl"/>
10 +
11 +   <xsl:output method="text" indent="yes"/>
12 +   <xsl:param name="input"/>
13 +   <xsl:param name="initiatedBy"/>
14 +   <xsl:template match="text()"/>
15 +
16 +   <xsl:template match="/">
17 +     <xsl:apply-templates/>
18 +
19 +
20 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/logback.xml 0 → 100644

```

1 +
2 + <configuration>
3 +   <property name="LOG_LEVEL" value="${LOG_LEVEL:-info}" />
4 +   <property name="LOG_LEVEL_ROOT" value="${LOG_LEVEL_ROOT:-error}" />
5 +
6 +   <!-- Console Appender -->
7 +   <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
8 +     <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
9 +   </appender>
10 +
11 +   <!-- Lambda Appender -->
12 +   <appender name="lambda" class="ch.qos.logback.core.ConsoleAppender">
13 +     <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
14 +     <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
15 +       <level>${LAMBDA_LOG_LEVEL:-info}</level>
16 +     </filter>
17 +   </appender>
18 +
19 +   <!-- Root Logger -->
20 +   <root level="${LOG_LEVEL_ROOT}">
21 +     <appender-ref ref="console"/>
22 +   </root>
23 +
24 +   <!-- Application Logger -->
25 +   <logger name="be.telenet" level="${LOG_LEVEL}" additivity="false">
26 +     <appender-ref ref="console"/>
27 +   </logger>
28 +
29 +   <!-- Lambda Logger -->
30 +   <logger name="com.amazonaws.services.lambda" level="${LOG_LEVEL:-info}" additivity="false">
31 +     <appender-ref ref="lambda"/>

```

```

32 | +     </logger>
33 | +   </configuration>
\ No newline at end of file

```

src/main/resources/transformer.properties 0 → 100644 +0 -0 Viewed

src/test/java/be/telenet/message/inbound/FnpNcbssInboundMessageTransformerTest.java 0 → 100644 +72 -0 Viewed

```

1 | + package be.telenet.message.inbound;
2 |
3 | + import static org.junit.jupiter.api.Assertions.assertNotNull;
4 |
5 | + import be.telenet.cnp.message.transfomer.MessageTransformer;
6 | + import java.io.IOException;
7 | + import java.net.URL;
8 | + import java.util.Objects;
9 | + import org.junit.jupiter.api.BeforeEach;
10 | + import org.junit.jupiter.api.Test;
11 | + import org.junit.jupiter.api.extension.ExtendWith;
12 | + import org.mockito.junit.jupiter.MockitoExtension;
13 | + import uk.org.webcompere.systemstubs.SystemStubs;
14 | + import uk.org.webcompere.systemstubs.jupiter.SystemStubsExtension;
15 |
16 | + @ExtendWith({MockitoExtension.class, SystemStubsExtension.class})
17 | + class FnpNcbssInboundMessageTransformerTest {
18 |
19 | +     private static final String PORT_IN_REQUEST_XML = "/FnpPortInRequest.xml";
20 | +     private static final String PORT_OUT_ACCEPT_XML = "/FnpPortOutAccept.xml";
21 | +     private MessageTransformer transformer;
22 |
23 | +     @BeforeEach
24 | +     void setUp() {
25 | +         URL stylesheet = FnpNcbssInboundMessageTransformer.class.getResource("/styles/inbound/inbound.transform.xsl");
26 | +         transformer = new MessageTransformer(stylesheet, true);
27 | +     }
28 |
29 | +     @Test
30 | +     void testConstructor() throws Exception {
31 | +         SystemStubs.withEnvironmentVariable("OUTBOUND_QUEUE_URL", "dummy_url")
32 | +             .and("AWS_REGION", "us-east-1")
33 | +             .execute(() -> {
34 | +                 FnpNcbssInboundMessageTransformer transformer = new FnpNcbssInboundMessageTransformer();
35 | +                 assertNotNull(transformer);
36 | +             });
37 | +     }
38 |
39 | +     @Test
40 | +     void testPortInRequest() throws Exception {
41 | +         // Given
42 | +         String xmlString = readFileToString(PORT_IN_REQUEST_XML);
43 |
44 | +         // When
45 | +         String transformedMessage = transformer.transform(xmlString, null);
46 |
47 | +         // Check result on: https://jsonformatter.org/ (remove originalmessage </FixedPortInRequest>)
48 | +         System.out.println(transformedMessage);
49 |
50 | +         // Then
51 | +         assertNotNull(transformedMessage);
52 | +     }
53 |
54 | +     @Test
55 | +     void testPortOutAccept() throws Exception {
56 | +         // Given
57 | +         String xmlString = readFileToString(PORT_OUT_ACCEPT_XML);
58 |
59 | +         // When
60 | +         String transformedMessage = transformer.transform(xmlString, null);
61 |
62 | +         System.out.println(transformedMessage);
63 |
64 | +         // Then
65 | +         assertNotNull(transformedMessage);
66 | +     }

```

```

67 +
68 +     public String readFileToString(String filePath) throws IOException {
69 +         return new String(Objects.requireNonNull(this.getClass().getResourceAsStream(filePath))
70 +             .readAllBytes());
71 +     }
72 +

```

src/test/resources/FnpPortInRequest.xml 0 → 100644

+43 -0 Viewed

```

1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <FixedPortInRequest xmlns="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortInRequest/v001"
3 +   xmlns:ns1="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002">
4 +   <ns1:IntegrationHeader>
5 +     <ns1:OrganizationalInfo>
6 +       <ns1:Company>TN</ns1:Company>
7 +     </ns1:OrganizationalInfo>
8 +     <ns1:MessageIdentification>
9 +       <ns1:CorrelationId>a6c0b01d-6bca-4018-b6ac-68000f2bfa51</ns1:CorrelationId>
10 +      <ns1:RequestId>bf02c51d-66be-4c79-9827-3562bb077e65</ns1:RequestId>
11 +      <ns1:MessageId>bf02c51d-66be-4c79-9827-3562bb077e66</ns1:MessageId>
12 +    </ns1:MessageIdentification>
13 +    <ns1:MessageTimestamps>
14 +      <ns1:CreationTimestamp>2022-01-10T05:27:32.536+02:00</ns1:CreationTimestamp>
15 +    </ns1:MessageTimestamps>
16 +    <ns1:SystemInfo>
17 +      <ns1:Environment>EDP</ns1:Environment>
18 +      <ns1:Application>
19 +        <ns1:ApplicationName>NC_BSS</ns1:ApplicationName>
20 +      </ns1:Application>
21 +    </ns1:SystemInfo>
22 +    <ns1:UserInfo>
23 +      <ns1:Identity>
24 +        <ns1:IdentityId>internal</ns1:IdentityId>
25 +        <ns1:IdentityType>user login</ns1:IdentityType>
26 +      </ns1:Identity>
27 +    </ns1:UserInfo>
28 +  </ns1:IntegrationHeader>
29 +  <FixedPortInRequestDetails>
30 +    <Range>
31 +      <FirstPhoneNumber>+3211226332</FirstPhoneNumber>
32 +      <LastPhoneNumber>+3211226332</LastPhoneNumber>
33 +    </Range>
34 +    <DueDate>2022-01-10T07:00:32.536+02:00</DueDate>
35 +    <RecipientServiceProviderCode>TELE</RecipientServiceProviderCode>
36 +    <InstallationAddress>
37 +      <AddressId>6cc8d3007q-9w8y44077v</AddressId>
38 +    </InstallationAddress>
39 +    <VAT/>
40 +    <NumberCount>1</NumberCount>
41 +    <ComplexityClass>SIMPLE</ComplexityClass>
42 +    <AccountNumber>1200009001</AccountNumber>
43 +  </FixedPortInRequestDetails>
44 + </FixedPortInRequest>
45 + \ No newline at end of file

```

src/test/resources/FnpPortOutAccept.xml 0 → 100644

+35 -0 Viewed

```

1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <FixedPortOutAuthorizeAccept xmlns="http://xmlns.telenet.be/integration/messages/NumberPortingFixed/FixedPortOutAuthorizeAccept/v001"
3 +   xmlns:ns1="http://xmlns.telenet.be/integration/headers/IntegrationHeader/v002">
4 +   <ns1:IntegrationHeader>
5 +     <ns1:OrganizationalInfo>
6 +       <ns1:Company>TN</ns1:Company>
7 +     </ns1:OrganizationalInfo>
8 +     <ns1:MessageIdentification>
9 +       <ns1:CorrelationId>68afc586-22bc-4318-b8ee-52b81155aceb</ns1:CorrelationId>
10 +      <ns1:RequestId>28349ad7-b053-41ca-9522-a31cfed802bc</ns1:RequestId>
11 +      <ns1:MessageId>28349ad7-b053-41ca-9522-a31cfed802bc</ns1:MessageId>
12 +    </ns1:MessageIdentification>
13 +    <ns1:MessageTimestamps>
14 +      <ns1:CreationTimestamp>2021-11-19T11:24:23.307+01:00</ns1:CreationTimestamp>
15 +    </ns1:MessageTimestamps>
16 +    <ns1:SystemInfo>
17 +      <ns1:Environment>LAT</ns1:Environment>
18 +      <ns1:Application>

```

```

19 | +         </ns1:Application>
20 | +     </ns1:SystemInfo>
21 | +     <ns1:UserInfo>
22 | +         <ns1:Identity>
23 | +             <ns1:IdentityId>internal</ns1:IdentityId>
24 | +             <ns1:IdentityType>user login</ns1:IdentityType>
25 | +         </ns1:Identity>
26 | +     </ns1:UserInfo>
27 | + </ns1:IntegrationHeader>
28 | + <FixedPortOutAuthorizeAcceptDetails>
29 | +     <Range>
30 | +         <FirstPhoneNumber>+3211720516</FirstPhoneNumber>
31 | +         <LastPhoneNumber>+3211720516</LastPhoneNumber>
32 | +     </Range>
33 | +     <ComplexityClass>SIMPLE</ComplexityClass>
34 | + </FixedPortOutAuthorizeAcceptDetails>
35 | + </FixedPortOutAuthorizeAccept>
36 | + \ No newline at end of file

```

✓ .gitignore 0 → 100644

```

1 | + HELP.md
2 | + target/
3 | + !mvn/wrapper/maven-wrapper.jar
4 | + !**/src/main/**/target/
5 | + !**/src/test/**/target/
6 |
7 | + ### STS ###
8 | + .apt_generated
9 | + .classpath
10 | + .factorypath
11 | + .project
12 | + .settings
13 | + .springBeans
14 | + .sts4-cache
15 |
16 | + ### IntelliJ IDEA ###
17 | + .idea
18 | + *.iws
19 | + *.iml
20 | + *.ipr
21 |
22 | + ### NetBeans ###
23 | + /nbproject/private/
24 | + /nbbuild/
25 | + /dist/
26 | + /nbdist/
27 | + /.nb-gradle/
28 | + build/
29 | + !**/src/main/**/build/
30 | + !**/src/test/**/build/
31 |
32 | + ### VS Code ###
33 | + .vscode/

```

✓ .gitlab-ci.yml 0 → 100644

```

1 | + variables:
2 | +   CI_PROJECT_NAME: "fnp-ncbss-message-transformer"
3 |
4 | + include:
5 | +   - project: 'tif/number-management/pipeline'
6 | +     file: '.gitlab-ci-application.yml'
7 | + \ No newline at end of file

```

✓ Dockerfile 0 → 100644

```

1 | + FROM public.ecr.aws/lambda/java:17
2 | + COPY target/fnp-ncbss-message-transformer*.jar ${LAMBDA_TASK_ROOT}/lib/

```

✓ lombok.config 0 → 100644

```

1 | + lombok.addLombokGeneratedAnnotation = true
2 | + \ No newline at end of file

```

pom.xml 0 → 100644

```

1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 +   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4 +
5 +   <parent>
6 +     <groupId>be.telenet.cnp</groupId>
7 +     <artifactId>cnp-lambda-parent</artifactId>
8 +     <version>1.0.299795</version>
9 +   </parent>
10 +
11 +   <groupId>be.telenet</groupId>
12 +   <artifactId>fnp-ncbss-message-transformer</artifactId>
13 +   <version>24.06.0-SNAPSHOT</version>
14 +
15 +   <properties>
16 +     <maven.compiler.source>17</maven.compiler.source>
17 +     <maven.compiler.target>17</maven.compiler.target>
18 +   </properties>
19 +
20 +   <dependencies>
21 +     <dependency>
22 +       <groupId>be.telenet.cnp</groupId>
23 +       <artifactId>lib.cnp-message-transformer</artifactId>
24 +       <version>24.06.552413</version>
25 +     </dependency>
26 +     <dependency>
27 +       <groupId>com.amazonaws</groupId>
28 +       <artifactId>aws-java-sdk-core</artifactId>
29 +     </dependency>
30 +     <dependency>
31 +       <groupId>com.amazonaws</groupId>
32 +       <artifactId>aws-java-sdk-sqs</artifactId>
33 +     </dependency>
34 +     <dependency>
35 +       <groupId>com.amazonaws</groupId>
36 +       <artifactId>aws-lambda-java-core</artifactId>
37 +       <version>1.2.2</version>
38 +     </dependency>
39 +     <dependency>
40 +       <groupId>com.amazonaws</groupId>
41 +       <artifactId>aws-lambda-java-events</artifactId>
42 +       <version>3.11.1</version>
43 +     </dependency>
44 +     <!-- Testing -->
45 +     <dependency>
46 +       <groupId>org.junit.jupiter</groupId>
47 +       <artifactId>junit-jupiter-engine</artifactId>
48 +       <scope>test</scope>
49 +     </dependency>
50 +     <dependency>
51 +       <groupId>org.junit.jupiter</groupId>
52 +       <artifactId>junit-jupiter-params</artifactId>
53 +       <scope>test</scope>
54 +     </dependency>
55 +     <dependency>
56 +       <groupId>com.amazonaws</groupId>
57 +       <artifactId>aws-lambda-java-tests</artifactId>
58 +       <scope>test</scope>
59 +     </dependency>
60 +     <dependency>
61 +       <groupId>org.mockito</groupId>
62 +       <artifactId>mockito-core</artifactId>
63 +       <scope>test</scope>
64 +     </dependency>
65 +     <dependency>
66 +       <groupId>org.mockito</groupId>
67 +       <artifactId>mockito-junit-jupiter</artifactId>
68 +       <scope>test</scope>
69 +     </dependency>
70 +     <dependency>
71 +       <groupId>uk.org.webcompere</groupId>
72 +       <artifactId>system-stubs-core</artifactId>
73 +       <version>2.0.2</version>
74 +       <scope>test</scope>
75 +     </dependency>

```

```

76 +     <dependency>
77 +         <groupId>uk.org.webcompere</groupId>
78 +         <artifactId>system-stubs-jupiter</artifactId>
79 +         <version>2.0.2</version>
80 +         <scope>test</scope>
81 +     </dependency>
82 +
83 +   </dependencies>
84 +
85 +   <profiles>
86 +     <profile>
87 +       <id>coverage</id>
88 +       <build>
89 +         <plugins>
90 +           <plugin>
91 +             <groupId>org.jacoco</groupId>
92 +             <artifactId>jacoco-maven-plugin</artifactId>
93 +             <version>${jacoco.maven.version}</version>
94 +             <executions>
95 +               <execution>
96 +                 <id>report</id>
97 +                 <goals>
98 +                   <goal>report</goal>
99 +                 </goals>
100 +               <phase>verify</phase>
101 +             </execution>
102 +           </executions>
103 +         </plugin>
104 +       </plugins>
105 +     </build>
106 +   </profile>
107 + </profiles>
108 +
109 + </project>

```

✓ settings.xml 0 → 100644

+94 -0 Viewed

```

1 | + <?xml version="1.0" encoding="UTF-8"?>
2 | + <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
3 | +   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 | +   xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 https://maven.apache.org/xsd/settings-1.0.0.xsd">
5 |
6 |   <servers>
7 |     <server>
8 |       <id>telenet.technicalorderfulfilment_releases</id>
9 |       <username>deployment</username>
10 |      <password>nxsdp1rel83</password>
11 |    </server>
12 |    <server>
13 |      <id>telenet.technicalorderfulfilment_snapshots</id>
14 |      <username>deployment</username>
15 |      <password>nxsdp1rel83</password>
16 |    </server>
17 |  </servers>
18 |
19 |  <mirrors>
20 |    <mirror>
21 |      <id>telenet.mirror</id>
22 |      <name>telenet.mirror</name>
23 |      <mirrorOf>*,!nexus-tn2-public</mirrorOf>
24 |      <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public/</url>
25 |    </mirror>
26 |  </mirrors>
27 |
28 |  <profiles>
29 |    <profile>
30 |      <id>sonar</id>
31 |      <activation>
32 |        <activeByDefault>true</activeByDefault>
33 |      </activation>
34 |      <properties>
35 |        <sonar.host.url>
36 |          https://sonarqube.cmt.apps.telenet.be/
37 |        </sonar.host.url>
38 |      </properties>
39 |    </profile>

```

```

40 +         <profile>
41 +             <id>nexus-tn2</id>
42 +             <activation>
43 +                 <activeByDefault>true</activeByDefault>
44 +             </activation>
45 +             <repositories>
46 +                 <repository>
47 +                     <id>nexus-tn2-public</id>
48 +                     <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public</url>
49 +                     <releases>
50 +                         <enabled>true</enabled>
51 +                         <checksumPolicy>warn</checksumPolicy>
52 +                     </releases>
53 +                     <snapshots>
54 +                         <enabled>false</enabled>
55 +                         <checksumPolicy>warn</checksumPolicy>
56 +                     </snapshots>
57 +                 </repository>
58 +             <repository>
59 +                 <id>telenet.technicalorderfulfilment_releases</id>
60 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_releases</url>
61 +                 <releases><enabled>true</enabled></releases>
62 +                 <snapshots><enabled>false</enabled></snapshots>
63 +             </repository>
64 +             <repository>
65 +                 <id>telenet.technicalorderfulfilment_snapshots</id>
66 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_snapshots</url>
67 +                 <releases><enabled>false</enabled></releases>
68 +                 <snapshots><enabled>true</enabled></snapshots>
69 +             </repository>
70 +         </repositories>
71 +
72 +         <pluginRepositories>
73 +             <pluginRepository>
74 +                 <id>nexus-tn2-public</id>
75 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/maven-public</url>
76 +                 <releases><enabled>true</enabled></releases>
77 +                 <snapshots><enabled>false</enabled></snapshots>
78 +             </pluginRepository>
79 +             <pluginRepository>
80 +                 <id>telenet.technicalorderfulfilment_releases</id>
81 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_releases</url>
82 +                 <releases><enabled>true</enabled></releases>
83 +                 <snapshots><enabled>false</enabled></snapshots>
84 +             </pluginRepository>
85 +             <pluginRepository>
86 +                 <id>telenet.technicalorderfulfilment_snapshots</id>
87 +                 <url>https://artifact-repo.rel.apps.telenet.be/repository/technicalorderfulfilment_snapshots</url>
88 +                 <releases><enabled>false</enabled></releases>
89 +                 <snapshots><enabled>true</enabled></snapshots>
90 +             </pluginRepository>
91 +         </pluginRepositories>
92 +     </profile>
93 +   </profiles>
94 + </settings>

```

CRDC inbound message transformer

```
src/main/java/be/telenet/message/inbound/FnpCrdcInboundMessageTransformer.java 0 → 100644 +14 -0 Viewed □ ⋮  
1 + package be.telenet.message.inbound;  
2 +  
3 + import be.telenet.cnp.message.handler.MessageHandler;  
4 + import java.util.Objects;  
5 +  
6 + public class FnpCrdcInboundMessageTransformer extends MessageHandler {  
7 +  
8 +     public FnpCrdcInboundMessageTransformer() {  
9 +         super(  
10 +             Objects.requireNonNull(  
11 +                 FnpCrdcInboundMessageTransformer.class.getResource("/styles/inbound/inbound.transform.xsl")),  
12 +                 Objects.requireNonNull(System.getenv("OUTBOUND_QUEUE_URL"), "OUTBOUND_QUEUE_URL  
not provided"));  
13 +     }  
14 + }
```



```
src/main/resources/styles/inbound/Fnpabort.xsl 0 → 100644 +35 -0 Viewed □ ⋮  
1 + <?xml version="1.0"?>  
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
3 +  
4 +     <xsl:template match="fnpabort">  
5 +         {  
6 +             "orderType": "PORT_OUT",  
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",  
8 +             "messageType": "FNP_ABORT",  
9 +             "originalMessage": "<xsl:value-of select='$input' />",  
10 +             "portingDetails": {  
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",  
12 +                 "donorId": "<xsl:value-of select='donorid' />",  
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",  
14 +                 "numberCount": "<xsl:value-of select='numbercount' />"  
15 +                 <xsl:if test="portingdetails">  
16 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />"  
17 +                 </xsl:if>  
18 +                 <xsl:if test="numberrange">  
19 +                     , "numberRange": [  
20 +                         <xsl:for-each select="numberrange">  
21 +                             {  
22 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",  
23 +                                 "numberTo": "<xsl:value-of select='numberto' />"  
24 +                                 <xsl:if test="codedid">  
25 +                                     , "codedId": "<xsl:value-of select='codedid' />"  
26 +                                 </xsl:if>  
27 +                                 }<xsl:if test="position()!=last()" />, </xsl:if>  
28 +                         </xsl:for-each>  
29 +                     ]  
30 +                 </xsl:if>  
31 +             }  
32 +         }  
33 +     </xsl:template>  
34 +  
35 + </xsl:stylesheet>
```

src/main/resources/styles/inbound/Fnpabortactivated.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpabortactivated">
5 +         {
6 +             "orderType": "PORT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_ABORT_ACTIVATED",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />"
15 +                 <xsl:if test="portingdetails">
16 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />" 
17 +                 </xsl:if>
18 +                 <xsl:if test="numberrange">
19 +                     , "numberRange": [
20 +                         <xsl:for-each select="numberrange">
21 +                             {
22 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",
23 +                                 "numberTo": "<xsl:value-of select='numberto' />" 
24 +                                 <xsl:if test="codedid">
25 +                                     , "codedId": "<xsl:value-of select='codedid' />" 
26 +                                 </xsl:if>
27 +                                 }<xsl:if test="position() != last()" />, </xsl:if>
28 +                         </xsl:for-each>
29 +                     ]
30 +                 </xsl:if>
31 +             }
32 +         }
33 +     </xsl:template>
34 +
35 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpabortactivatedbycrdc.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpabortactbycrdc">
5 +         {
6 +             "orderType": "PORT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_ABORT_ACTIVATED_BY_CRDC",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />" 
15 +                 <xsl:if test="portingdetails">
16 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />" 
17 +                 </xsl:if>

```

```

18+         <xsl:if test="numberrange">
19+             , "numberRange": [
20+                 <xsl:for-each select="numberrange">
21+                     {
22+                         "numberFrom": "<xsl:value-of select='numberfrom' />",
23+                         "numberTo": "<xsl:value-of select='numberto' />"
24+                         <xsl:if test="codedid">
25+                             , "codedId": "<xsl:value-of select='codedid' />" 
26+                         </xsl:if>
27+                         }<xsl:if test="position() != last()">,</xsl:if>
28+                     </xsl:for-each>
29+                 ]
30+             </xsl:if>
31+         }
32+     }
33+   </xsl:template>
34+
35+ </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpaccept.xsl 0 → 100644 +36 -0 Viewed

```

1+ <?xml version="1.0"?>
2+ <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3+
4+     <xsl:template match="fnpaccept">
5+         {
6+             "orderType": "PORT_IN",
7+             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8+             "messageType": "FNP_ACCEPT",
9+             "originalMessage": "<xsl:value-of select='$input' />",
10+            "portingDetails": {
11+                "recipientId": "<xsl:value-of select='recipientid' />",
12+                "donorId": "<xsl:value-of select='donorid' />",
13+                "idNumber": "<xsl:value-of select='idnumber' />",
14+                "numberCount": "<xsl:value-of select='numbercount' />" 
15+                <xsl:if test="portingdetails">
16+                    , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
17+                    "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />" 
18+                </xsl:if>
19+                <xsl:if test="numberrange">
20+                    , "numberRange": [
21+                        <xsl:for-each select="numberrange">
22+                            {
23+                                "numberFrom": "<xsl:value-of select='numberfrom' />",
24+                                "numberTo": "<xsl:value-of select='numberto' />" 
25+                                <xsl:if test="codedid">
26+                                    , "codedId": "<xsl:value-of select='codedid' />" 
27+                                </xsl:if>
28+                                }<xsl:if test="position() != last()">,</xsl:if>
29+                            </xsl:for-each>
30+                        ]
31+                    </xsl:if>
32+                }
33+            }
34+        </xsl:template>

```

```

35 + </xsl:stylesheet>
36 + \ No newline at end of file

```

src/main/resources/styles/inbound/Fnpcancel.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +   <xsl:template match="fnpcancel">
5 +     {
6 +       "orderType": "PORT_OUT",
7 +       "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +       "messageType": "FNP_CANCEL",
9 +       "originalMessage": "<xsl:value-of select='$input' />",
10 +      "portingDetails": {
11 +        "recipientId": "<xsl:value-of select='recipientid' />",
12 +        "donorId": "<xsl:value-of select='donorid' />",
13 +        "idNumber": "<xsl:value-of select='idnumber' />",
14 +        "numberCount": "<xsl:value-of select='numbercount' />"
15 +        <xsl:if test="portingdetails">
16 +          "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />"
17 +        </xsl:if>
18 +        <xsl:if test="numberrange">
19 +          "numberRange": [
20 +            <xsl:for-each select="numberrange">
21 +              {
22 +                "numberFrom": "<xsl:value-of select='numberfrom' />",

23 +                "numberTo": "<xsl:value-of select='numberto' />"
24 +                <xsl:if test="codedid">
25 +                  "codedId": "<xsl:value-of select='codedid' />"
26 +                </xsl:if>
27 +                }<xsl:if test="position() != last()">,</xsl:if>
28 +              </xsl:for-each>
29 +            ]
30 +          </xsl:if>
31 +        }
32 +      }
33 +    </xsl:template>
34 +
35 +  </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpcancelledbyportingauthority.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +   <xsl:template match="fnpcancelledbycrdc">
5 +     {
6 +       "orderType": "PORT_IN",
7 +       "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +       "messageType": "FNP_CANCELLED_BY_PORTING_AUTHORITY",
9 +       "originalMessage": "<xsl:value-of select='$input' />",
10 +      "portingDetails": {

```

```

11 +         "recipientId": "<xsl:value-of select='recipientid' />",
12 +         "donorId": "<xsl:value-of select='donorid' />",
13 +         "idNumber": "<xsl:value-of select='idnumber' />",
14 +         "numberCount": "<xsl:value-of select='numbercount' />"
15 +         <xsl:if test="portingdetails">
16 +             , "npDueDate": "<xsl:value-of select='portingdetails/npdue date' />""
17 +         </xsl:if>
18 +         <xsl:if test="numberrange">
19 +             , "numberRange": [
20 +                 <xsl:for-each select="numberrange">
21 +                     {
22 +                         "numberFrom": "<xsl:value-of select='numberfrom' />",
23 +                         "numberTo": "<xsl:value-of select='numberto' />""
24 +                         <xsl:if test="codedid">
25 +                             , "codedId": "<xsl:value-of select='codedid' />""
26 +                         </xsl:if>
27 +                         }<xsl:if test="position() != last()">, </xsl:if>
28 +                     </xsl:for-each>
29 +                 ]
30 +             </xsl:if>
31 +         }
32 +     }
33 +     </xsl:template>
34 +
35 + </xsl:stylesheet>

```

\ No newline at end of file

src/main/resources/styles/inbound/Fnpchange.xsl 0 → 100644 +37 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpchange">
5 +         {
6 +             "orderType": "PORT_OUT",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_CHANGE",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />""
15 +                 <xsl:if test="portingdetails">
16 +                     , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
17 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npdue date' />"",
18 +                     , "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />""
19 +                 </xsl:if>
20 +                 <xsl:if test="numberrange">
21 +                     , "numberRange": [
22 +                         <xsl:for-each select="numberrange">
23 +                             {
24 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",
25 +                                 "numberTo": "<xsl:value-of select='numberto' />""
26 +                                 <xsl:if test="codedid">
27 +                                     , "codedId": "<xsl:value-of select='codedid' />""
28 +                                 </xsl:if>

```

```

29 +             }<xsl:if test="position() != last()">,</xsl:if>
30 +         </xsl:for-each>
31 +     ]
32 +   </xsl:if>
33 + }
34 + }
35 + </xsl:template>
36 +
37 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpchangeaccept.xsl 0 → 100644 +36 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +   <xsl:template match="fnpchangeaccept">
5 +     {
6 +       "orderType": "PORT_IN",
7 +       "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +       "messageType": "FNP_CHANGE_ACCEPT",
9 +       "originalMessage": "<xsl:value-of select='$input' />",
10 +      "portingDetails": {
11 +        "recipientId": "<xsl:value-of select='recipientid' />",
12 +        "donorId": "<xsl:value-of select='donorid' />",
13 +        "idNumber": "<xsl:value-of select='idnumber' />",
14 +        "numberCount": "<xsl:value-of select='numbercount' />"
15 +        <xsl:if test="portingdetails">
16 +          "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />",
17 +          "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />"
18 +        </xsl:if>
19 +        <xsl:if test="numberrange">
20 +          "numberRange": [
21 +            <xsl:for-each select="numberrange">
22 +              {
23 +                "numberFrom": "<xsl:value-of select='numberfrom' />",
24 +                "numberTo": "<xsl:value-of select='numberto' />"
25 +                <xsl:if test="codedid">
26 +                  "codedId": "<xsl:value-of select='codedid' />"
27 +                </xsl:if>
28 +                }<xsl:if test="position() != last()">,</xsl:if>
29 +              </xsl:for-each>
30 +            ]
31 +          </xsl:if>
32 +        }
33 +      }
34 +    </xsl:template>
35 +
36 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpchangereject.xsl 0 → 100644 +37 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +

```

```

4 +     <xsl:template match="fnpcchange">
5 +     {
6 +         "orderType": "PORT_IN",
7 +         "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +         "messageType": "FNP_CHANGE_REJECT",
9 +         "originalMessage": "<xsl:value-of select='$input' />",
10 +        "portingDetails": {
11 +            "recipientId": "<xsl:value-of select='recipientid' />",
12 +            "donorId": "<xsl:value-of select='donorid' />",
13 +            "idNumber": "<xsl:value-of select='idnumber' />",
14 +            "numberCount": "<xsl:value-of select='numbercount' />"
15 +            <xsl:if test="portingdetails">
16 +                , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
17 +                "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />",
18 +                "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />"
19 +            </xsl:if>
20 +            <xsl:if test="numberrange">
21 +                , "numberRange": [
22 +                    <xsl:for-each select="numberrange">
23 +                        {
24 +                            "numberFrom": "<xsl:value-of select='numberfrom' />",
25 +                            "numberTo": "<xsl:value-of select='numberto' />"
26 +                            <xsl:if test="codedid">
27 +                                , "codedId": "<xsl:value-of select='codedid' />"
28 +                            </xsl:if>
29 +                            }<xsl:if test="position() != last()" />, </xsl:if>
30 +                        </xsl:for-each>
31 +                    ]
32 +                </xsl:if>
33 +            }
34 +        }
35 +    </xsl:template>
36 +
37 + </xsl:stylesheet>
   \ No newline at end of file

```

src/main/resources/styles/inbound/Fnpdeactbroadcast.xsl 0 → 100644 +32 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpdeactbroadcast">
5 +         {
6 +             "orderType": "DISCONNECT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_DEACT_BROADCAST",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +            "portingDetails": {
11 +                "recipientId": "<xsl:value-of select='recipientid' />",
12 +                "donorId": "<xsl:value-of select='donorid' />",
13 +                "idNumber": "<xsl:value-of select='idnumber' />",
14 +                "numberCount": "<xsl:value-of select='numbercount' />"
15 +                <xsl:if test="numberrange">
16 +                    , "numberRange": [
17 +                        <xsl:for-each select="numberrange">
18 +                            {

```

```

19 +         "numberFrom": "<xsl:value-of select='numberfrom' />",
20 +         "numberTo": "<xsl:value-of select='numberto' />"
21 +         <xsl:if test="codedid">
22 +             , "codedId": "<xsl:value-of select='codedid' />" 
23 +             </xsl:if>
24 +             }<xsl:if test="position() != last()">, </xsl:if>
25 +             </xsl:for-each>
26 +             ]
27 +         </xsl:if>
28 +     }
29 +   }
30 + 
```

\ No newline at end of file

▼ src/main/resources/styles/inbound/Fnpdiscdone.xsl 0 → 100644 +32 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpdiscdone">
5 +         {
6 +             "orderType": "DISCONNECT_OUT",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_DISC_DONE",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />"
15 +                 <xsl:if test="numberrange">
16 +                     , "numberRange": [
17 +                         <xsl:for-each select="numberrange">
18 +                             {
19 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",
20 +                                 "numberTo": "<xsl:value-of select='numberto' />"
21 +                                 <xsl:if test="codedid">
22 +                                     , "codedId": "<xsl:value-of select='codedid' />" 
23 +                                     </xsl:if>
24 +                                     }<xsl:if test="position() != last()">, </xsl:if>
25 +                                     </xsl:for-each>
26 +                                     ]
27 +                                 </xsl:if>
28 +                             }
29 +                         }
30 +             
```

\ No newline at end of file

▼ src/main/resources/styles/inbound/Fnpdisconnect.xsl 0 → 100644 +32 -0 Viewed

```

1 + <?xml version="1.0"?>
```

```

4 +     <xsl:template match="fnpdisconnect">
5 +         {
6 +             "orderType": "DISCONNECT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_DISCONNECT",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +            "portingDetails": {
11 +                "recipientId": "<xsl:value-of select='recipientid' />",
12 +                "donorId": "<xsl:value-of select='donorid' />",
13 +                "idNumber": "<xsl:value-of select='idnumber' />",
14 +                "numberCount": "<xsl:value-of select='numbercount' />"
15 +                <xsl:if test="numberrange">
16 +                    , "numberRange": [
17 +                        <xsl:for-each select="numberrange">
18 +                            {
19 +                                "numberFrom": "<xsl:value-of select='numberfrom' />",
20 +                                "numberTo": "<xsl:value-of select='numberto' />"
21 +                                <xsl:if test="codedid">
22 +                                    , "codedId": "<xsl:value-of select='codedid' />"
23 +                                </xsl:if>
24 +                                }<xsl:if test="position() != last()">, </xsl:if>
25 +                            </xsl:for-each>
26 +                        ]
27 +                    ]
28 +                }
29 +            }
30 +        </xsl:template>
31 +

```

src/main/resources/styles/inbound/Fnnpdisconnect.xsl 0 → 100644

+32 -0 Viewed

```

32 +     </xsl:stylesheet>
|       \ No newline at end of file

```

src/main/resources/styles/inbound/Fnpexec.xsl 0 → 100644

+37 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpexec">
5 +         {
6 +             "orderType": "PORT_OUT",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_EXEC",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +            "portingDetails": {
11 +                "recipientId": "<xsl:value-of select='recipientid' />",
12 +                "donorId": "<xsl:value-of select='donorid' />",
13 +                "idNumber": "<xsl:value-of select='idnumber' />",
14 +                "numberCount": "<xsl:value-of select='numbercount' />"
15 +                <xsl:if test="portingdetails">
16 +                    , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
17 +                    "npDueDate": "<xsl:value-of select='portingdetails/n pdu date' />",
18 +                    "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />"
19 +                </xsl:if>
20 +                <xsl:if test="numberrange">
21 +                    , "numberRange": [
22 +                        <xsl:for-each select="numberrange">
23 +                            {

```

```

24 +             "numberFrom": "<xsl:value-of select='numberfrom' />",
25 +             "numberTo": "<xsl:value-of select='numberto' />"
26 +
27 +             "<xsl:if test='codedid'>
28 +                 , "codedId": "<xsl:value-of select='codedid' />""
29 +             }<xsl:if test='position() != last()'>, </xsl:if>
30 +             </xsl:for-each>
31 +         ]
32 +     </xsl:if>
33 +   }
34 + }
35 + </xsl:template>
36 +
37 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnphold.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnphold">
5 +         {
6 +             "orderType": "PORT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_HOLD",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />"}
15 +                 <xsl:if test="portingdetails">
16 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npdueDate' />"}
17 +                 </xsl:if>
18 +                 <xsl:if test="numberrange">
19 +                     , "numberRange": [
20 +                         <xsl:for-each select="numberrange">
21 +                             {
22 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",
23 +                                 "numberTo": "<xsl:value-of select='numberto' />"}
24 +                                 "<xsl:if test='codedid'>
25 +                                     , "codedId": "<xsl:value-of select='codedid' />""
26 +                                 </xsl:if>
27 +                                 }<xsl:if test='position() != last()'>, </xsl:if>
28 +                                 </xsl:for-each>
29 +                             ]
30 +                         </xsl:if>
31 +                     }
32 +                 }
33 +             </xsl:template>
34 +
35 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpnonrfs.xsl 0 → 100644 +42 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnponrfs">
5 +         {
6 +             "orderType": "PORT_OUT",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_NON_RFS",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +            "portingDetails": {
11 +                "recipientId": "<xsl:value-of select='recipientid' />",
12 +                "donorId": "<xsl:value-of select='donorid' />",
13 +                "idNumber": "<xsl:value-of select='idnumber' />",
14 +                "numberCount": "<xsl:value-of select='numbercount' />"
15 +            }
16 +            <xsl:if test="portingdetails">
17 +                , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />"
18 +            </xsl:if>
19 +            <xsl:if test="numberrange">
20 +                , "numberRange": [
21 +                    <xsl:for-each select="numberrange">
22 +                        {
23 +                            "numberFrom": "<xsl:value-of select='numberfrom' />",
24 +                            "numberTo": "<xsl:value-of select='numberto' />"
25 +                            <xsl:if test="rejectreason">
26 +                                , "rejectReason": [
27 +                                    <xsl:for-each select="rejectreason">
28 +                                        {
29 +                                            "rejectCode": "<xsl:value-of select='rejectcode' />",
30 +                                            "rejectString": "<xsl:value-of select='rejectstring' />"
31 +                                            }<xsl:if test="position() != last()" />, </xsl:if>
32 +                                        ]
33 +                                    </xsl:if>
34 +                                    }<xsl:if test="position() != last()" />, </xsl:if>
35 +                                </xsl:for-each>
36 +                                ]
37 +                            </xsl:if>
38 +                        }
39 +                    ]
40 +                </xsl:if>
41 +            ]
42 +        </xsl:template>
43 +
44 +    </xsl:stylesheet>

```

\ No newline at end of file

src/main/resources/styles/inbound/Fnpready.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpready">
5 +         {
6 +             "orderType": "PORT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_READY",
9 +             "originalMessage": "<xsl:value-of select='$input' />",

```

```

10 +         "portingDetails": {
11 +             "recipientId": "<xsl:value-of select='recipientid' />",
12 +             "donorId": "<xsl:value-of select='donorid' />",
13 +             "idNumber": "<xsl:value-of select='idnumber' />",
14 +             "numberCount": "<xsl:value-of select='numbercount' />"
15 +             <xsl:if test="portingdetails">
16 +                 , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />""
17 +             </xsl:if>
18 +             <xsl:if test="numberrange">
19 +                 , "numberRange": [
20 +                     <xsl:for-each select="numberrange">
21 +                         {
22 +                             "numberFrom": "<xsl:value-of select='numberfrom' />",
23 +                             "numberTo": "<xsl:value-of select='numberto' />""
24 +                             <xsl:if test="codedid">
25 +                                 , "codedId": "<xsl:value-of select='codedid' />""
26 +                             </xsl:if>
27 +                             }<xsl:if test="position() != last()">, </xsl:if>
28 +                         </xsl:for-each>
29 +                     ]
30 +                 }
31 +             }
32 +         }
33 +     </xsl:template>
34 +
35 + </xsl:stylesheet>
\ No newline at end of file

```

src/main/resources/styles/inbound/Fnpreject.xsl 0 → 100644

+46 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnpreject">
5 +         {
6 +             "orderType": "PORT_IN",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_REJECT",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />""
15 +                 <xsl:if test="portingdetails">
16 +                     , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
17 +                     , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />""
18 +                 </xsl:if>
19 +                 <xsl:if test="numberrange">
20 +                     , "numberRange": [
21 +                         <xsl:for-each select="numberrange">
22 +                             {
23 +                                 "numberFrom": "<xsl:value-of select='numberfrom' />",
24 +                                 "numberTo": "<xsl:value-of select='numberto' />""
25 +                                 <xsl:if test="codedid">
26 +                                     , "codedId": "<xsl:value-of select='codedid' />""
27 +                                 </xsl:if>

```

```

28 +             <xsl:if test="rejectreason">
29 +                 ,"rejectReason": [
30 +                     <xsl:for-each select="rejectreason">
31 +                         {
32 +                             "rejectCode": "<xsl:value-of select='rejectcode' />",
33 +                             "rejectString": "<xsl:value-of select='rejectstring' />"}
34 +                         }<xsl:if test="position() != last()">,</xsl:if>
35 +                     </xsl:for-each>
36 +                 ]
37 +             </xsl:if>
38 +             }<xsl:if test="position() != last()">,</xsl:if>
39 +         </xsl:for-each>
40 +     ]
41 +     </xsl:if>
42 +   }
43 + }
44 + </xsl:template>
45 +
46 + </xsl:stylesheet>

```

\ No newline at end of file

src/main/resources/styles/inbound/Fnprequest.xsl 0 → 100644 +57 -0 Viewed

```

1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +     <xsl:template match="fnprequest">
5 +         {
6 +             "orderType": "PORT_OUT",
7 +             "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +             "messageType": "FNP_REQUEST",
9 +             "originalMessage": "<xsl:value-of select='$input' />",
10 +             "portingDetails": {
11 +                 "recipientId": "<xsl:value-of select='recipientid' />",
12 +                 "donorId": "<xsl:value-of select='donorid' />",
13 +                 "idNumber": "<xsl:value-of select='idnumber' />",
14 +                 "numberCount": "<xsl:value-of select='numbercount' />"
15 +             }
16 +             <xsl:if test="portingdetails">
17 +                 , "complexityClass": "<xsl:value-of select='portingdetails/complexityclass' />",
18 +                 , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />",
19 +                 , "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />"
20 +                 <xsl:choose>
21 +                     <xsl:when test="portingdetails/business">
22 +                         , "businessDetails": {
23 +                             "vat": "<xsl:value-of select='portingdetails/business/vat' />",
24 +                             "accountNumber": "<xsl:value-of
25 +                                 select='portingdetails/business/accountnumber' />",
26 +                             "zipcode": "<xsl:value-of select='portingdetails/business/zipcode' />",
27 +                             "houseNumber": "<xsl:value-of select='portingdetails/business/housenumber' />",
28 +                             "street": "<xsl:value-of select='portingdetails/business/street' />"
29 +                         }
30 +                     </xsl:when>
31 +                     <xsl:when test="portingdetails/residential">

```

```

32 +             "accountNumber": "<xsl:value-of
33 +                 select='portingdetails/residential/accountnumber' />",
34 +             "zipcode": "<xsl:value-of select='portingdetails/residential/zipcode' />",
35 +             "houseNumber": "<xsl:value-of
36 +                 select='portingdetails/residential/housenumber' />",
37 +             "street": "<xsl:value-of select='portingdetails/residential/street' />"
38 +         }
39 +     
```

```

39 +   
```

```

40 +     </xsl:choose>
41 +   }
42 +   <xsl:if test="numberrange">
43 +     , "numberRange": [
44 +       <xsl:for-each select="numberrange">
45 +         {
46 +           "numberFrom": "<xsl:value-of select='numberfrom' />",
47 +           "numberTo": "<xsl:value-of select='numberto' />"
48 +           <xsl:if test="codedid">
49 +             , "codedId": "<xsl:value-of select='codedid' />"
50 +           </xsl:if>
51 +           }<xsl:if test="position() != last()" />, </xsl:if>
52 +         </xsl:for-each>
53 +       ]
54 +     </xsl:if>
55 +   }
56 + 
```

```

57 + </xsl:template>
58 + 
```

```

59 + </xsl:stylesheet>
```

▼ src/main/resources/styles/inbound/Fnprfsbroadcast.xsl 0 → 100644

+36 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +   <xsl:template match="fnprfsbroadcast">
5 +     {
6 +       "orderType": "PORT_ACROSS",
7 +       "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +       "messageType": "FNP_RFS_BROADCAST",
9 +       "originalMessage": "<xsl:value-of select='$input' />",
10 +      "portingDetails": {
11 +        "recipientId": "<xsl:value-of select='recipientid' />",
12 +        "donorId": "<xsl:value-of select='donorid' />",
13 +        "idNumber": "<xsl:value-of select='idnumber' />",
14 +        "numberCount": "<xsl:value-of select='numbercount' />"}
15 +      <xsl:if test="portingdetails">
16 +        , "npDueDate": "<xsl:value-of select='portingdetails/npduedate' />",
17 +        "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />"
18 +      </xsl:if>
19 +      <xsl:if test="numberrange">
20 +        , "numberRange": [
21 +          <xsl:for-each select="numberrange">
22 +            {
23 +              "numberFrom": "<xsl:value-of select='numberfrom' />",
24 +              "numberTo": "<xsl:value-of select='numberto' />"
25 +              <xsl:if test="codedid">
26 +                , "codedId": "<xsl:value-of select='codedid' />"
27 +              </xsl:if>
```

```

28 +             }<xsl:if test="position() != last()">,</xsl:if>
29 +         </xsl:for-each>
30 +     ]
31 +   </xsl:if>
32 + }
33 + }
34 + </xsl:template>
35 +
36 + </xsl:stylesheet>

```

\ No newline at end of file

src/main/resources/styles/inbound/Fnpupdate.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 +
4 +   <xsl:template match="fnupdate">
5 +     {
6 +       "orderType": "UPDATE",
7 +       "initiatedBy": "<xsl:value-of select='$initiatedBy' />",
8 +       "messageType": "FNP_UPDATE",
9 +       "originalMessage": "<xsl:value-of select='$input' />",
10 +      "portingDetails": {
11 +        "recipientId": "<xsl:value-of select='recipientid' />",
12 +        "donorId": "<xsl:value-of select='donorid' />",
13 +        "idNumber": "<xsl:value-of select='idnumber' />",
14 +        "numberCount": "<xsl:value-of select='numbercount' />"
15 +        <xsl:if test="portinadetails">
16 +          , "routingNumber": "<xsl:value-of select='portingdetails/routinginfo' />" 
17 +        </xsl:if>
18 +        <xsl:if test="numberrange">
19 +          , "numberRange": [
20 +            <xsl:for-each select="numberrange">
21 +              {
22 +                "numberFrom": "<xsl:value-of select='numberfrom' />",
23 +                "numberTo": "<xsl:value-of select='numberto' />" 
24 +                <xsl:if test="codedid">
25 +                  , "codedId": "<xsl:value-of select='codedid' />" 
26 +                </xsl:if>
27 +                  }<xsl:if test="position() != last()">,</xsl:if>
28 +                </xsl:for-each>
29 +              ]
30 +            </xsl:if>
31 +          }
32 +        }
33 +      </xsl:template>
34 +
35 + </xsl:stylesheet>

```

\ No newline at end of file

src/main/resources/styles/inbound/inbound.transform.xsl 0 → 100644 +35 -0 Viewed

```

1 + <?xml version="1.0"?>
2 + <xsl:stylesheet version="3.0"
3 +   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4 +

```

```

5 +     <xsl:import href="Fnpdisconnect.xsl"/>
6 +     <xsl:import href="Fnndeactbroadcast.xsl"/>
7 +     <xsl:import href="Fnndisdone.xsl"/>
8 +     <xsl:import href="Fnpaccept.xsl"/>
9 +     <xsl:import href="Fnpready.xsl"/>
10 +    <xsl:import href="Fnprfsbroadcast.xsl"/>
11 +    <xsl:import href="Fnpreject.xsl"/>
12 +    <xsl:import href="Fnphold.xsl"/>
13 +    <xsl:import href="Fnpcchangereject.xsl"/>
14 +    <xsl:import href="Fnpcchangeaccept.xsl"/>
15 +    <xsl:import href="Fnpcancelledbyportingauthority.xsl"/>
16 +    <xsl:import href="Fnabortionactivated.xsl"/>
17 +    <xsl:import href="Fnabortionactivatedbycrdc.xsl"/>
18 +    <xsl:import href="Fnprequest.xsl"/>
19 +    <xsl:import href="Fnexec.xsl"/>
20 +    <xsl:import href="Fncancel.xsl"/>
21 +    <xsl:import href="Fnchange.xsl"/>
22 +    <xsl:import href="Fnabort.xsl"/>
23 +    <xsl:import href="Fnnonrfs.xsl"/>
24 +    <xsl:import href="Fnupdate.xsl"/>
25 +
26 +    <xsl:output method="text" indent="yes"/>
27 +    <xsl:param name="input"/>
28 +    <xsl:param name="initiatedBy"/>
29 +    <xsl:template match="text()"/>
30 +
31 +    <xsl:template match="/">
32 +        <xsl:apply-templates/>
33 +    </xsl:template>
34 +
35 + </xsl:stylesheet>

```

\ No newline at end of file

	src/main/resources/logback.xml		0 → 100644	+33 -0	<input type="checkbox"/> Viewed		
--	--------------------------------	--	------------	--------	---------------------------------	--	--

```

1 +
2 + <configuration>
3 +     <property name="LOG_LEVEL" value="${LOG_LEVEL:-info}" />
4 +     <property name="LOG_LEVEL_ROOT" value="${LOG_LEVEL_ROOT:-error}" />
5 +
6 +     <!-- Console Appender -->
7 +     <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
8 +         <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
9 +     </appender>
10 +
11 +     <!-- Lambda Appender -->
12 +     <appender name="lambda" class="ch.qos.logback.core.ConsoleAppender">
13 +         <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
14 +         <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
15 +             <level>${LAMBDA_LOG_LEVEL:-info}</level>
16 +         </filter>
17 +     </appender>
18 +
19 +     <!-- Root Logger -->
20 +     <root level="${LOG_LEVEL_ROOT}">
21 +         <appender-ref ref="console"/>

```

```

22 +     </root>
23 +
24 +     <!-- Application Logger -->
25 +     <logger name="be.telenet" level="${LOG_LEVEL}" additivity="false">
26 +         <appender-ref ref="console"/>
27 +     </logger>
28 +
29 +     <!-- Lambda Logger -->
30 +     <logger name="com.amazonaws.services.lambda" level="${LOG_LEVEL:-info}" additivity="false">
31 +         <appender-ref ref="lambda"/>
32 +     </logger>
33 + </configuration>

```

\ No newline at end of file

src/main/resources/transformer.properties 0 → 100644 +6 -0 Viewed

```

1 + transformer.inbound.stylesheet=/styles/inbound/inbound.transform.xsl
2 + transformer.inbound.destination=fnp_inbound
3 + transformer.region=eu-central-1
4 +
5 + transformer.outbound.stylesheet=/styles.outbound/outbound.transform.xsl
6 + transformer.outbound.destination=fnp_crdc_transformed_outbound

```

src/test/java/be/telenet/message/inbound/FnpCrdcInboundMessageTransformer +23 -0 Viewed

Test.java 0 → 100644

```

1 + package be.telenet.message.inbound;
2 +
3 + import static org.junit.jupiter.api.Assertions.assertNotNull;
4 +
5 + import org.junit.jupiter.api.Test;
6 + import org.junit.jupiter.api.extension.ExtendWith;
7 + import org.mockito.junit.jupiter.MockitoExtension;
8 + import uk.org.webcompere.systemstubs.SystemStubs;
9 + import uk.org.webcompere.systemstubs.jupiter.SystemStubsExtension;
10 +
11 + @ExtendWith({MockitoExtension.class, SystemStubsExtension.class})
12 + class FnpCrdcInboundMessageTransformerTest {
13 +
14 +     @Test
15 +     void testConstructor() throws Exception {
16 +         SystemStubs.withEnvironmentVariable("OUTBOUND_QUEUE_URL", "dummy_url")
17 +             .and("AWS_REGION", "us-east-1")
18 +             .execute(() -> {
19 +                 FnpCrdcInboundMessageTransformer transformer = new
20 +                     FnpCrdcInboundMessageTransformer();
21 +                 assertNotNull(transformer);
22 +             });
23 +     }

```

src/test/resources/Fnprequest.xml 0 → 100644 +24 -0 Viewed

```
1 + <?xml version="1.0" encoding="UTF-8"?>
2 + <fnprequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 +   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4 +   <version xsi:type="xsd:string">01.00</version>
5 +   <datetime xsi:type="xsd:dateTime">2024-02-29T12:10:45</datetime>
6 +   <recipientid xsi:type="xsd:string">TELE</recipientid>
7 +   <donorid xsi:type="xsd:string">BGC</donorid>
8 +   <idnumber xsi:type="xsd:string">DUMMY</idnumber>
9 +   <portingdetails>
10 +     <complexityclass xsi:type="xsd:string">Simple</complexityclass>
11 +     <nptdueate xsi:type="xsd:dateTime">2024-03-06T08:00:00</nptdueate>
12 +     <routinginfo xsi:type="xsd:string">C3289</routinginfo>
13 +     <residential>
14 +       <zipcode xsi:type="xsd:integer">2018</zipcode>
15 +       <housenumber xsi:type="xsd:integer">69</housenumber>
16 +       <street xsi:type="xsd:string">Molenstraat [Antwerpen]</street>
17 +     </residential>
18 +   </portingdetails>
19 +   <numbercount xsi:type="xsd:integer">1</numbercount>
20 +   <numberrange>
21 +     <numberfrom xsi:type="xsd:integer">032160066</numberfrom>
22 +     <numberto xsi:type="xsd:integer">032160066</numberto>
23 +   </numberrange>
24 + </fnprequest>
\ No newline at end of file
```

Order Search Export button

For my final development story, I was tasked with creating an export button for the Orders section in the user interface. Although an export button already existed for the Tasks, the analytics team specifically requested a similar feature for Orders to enhance their reporting capabilities and improve data accessibility. This new functionality would allow users to extract order data directly from the system for further analysis. The project involved working with a UI built using both React and Vue.

To meet the requirements, I designed and implemented a custom export button with two core functionalities: print and download as CSV (convertible to an Excel file). The print feature enabled users to generate a printer-friendly view of the orders, while the CSV download allowed them to extract and save the data for external processing in tools like Microsoft Excel.

The screenshot shows the 'Search Orders' interface. At the top, there are dropdowns for 'Complexity', 'Order Type', and 'Order Status', and input fields for 'Msisdn', 'Customer Number', and 'NPR Number'. Below these are date pickers for 'From Date' (07/10/2024 03:55 AM) and 'To Date' (mm/dd/yyyy). There are 'Search' and 'Clear' buttons. The main area displays 'Order Search Results' with columns: Msisdn, Npr Number, Order Type, Order Status, Step Name, Order Creation Time, Customer Num, Complexity, and a row of icons for each order. An 'EXPORT' button is visible, with a tooltip showing 'Download as CSV' and 'Print'. At the bottom, there's a 'Print' dialog box with various settings like paper size, orientation, and page range, with 'Print' and 'Cancel' buttons.

id	msisdn	nprNumber	orderType	orderStatus	stepName	orderCreationTime	customerNumber	complexity
d21fb964-ec30-4cae-91f1-e2bea1ef4a3e	0460006546	0460006546	Port In	Open	Request Enriched	January 15, 2025 04:12:24.0409 AM	Post Paid Simple	Post Paid Simple
33b01c06-h2e1-4913-89f0-c0bc2289c28e	0460006623	9617199874166	Port Out	Open	Accept Feedback Tr	January 15, 2025 03:59:45.0870 AM	Post Paid Simple	Post Paid Simple

about:blank

Import favorites Telenet Group

<i>id</i>	<i>msisdn</i>	<i>npnNumber</i>	<i>orderType</i>	<i>orderStatus</i>	<i>stepName</i>	<i>orderCreationTime</i>	<i>customerNumber</i>	<i>complexity</i>
d21fb964-ac30-4cae-9tf-f-c2bea1df4a3e	0460006546	0460006546	Port In	Open	Accept To Antium	2025-01-15T04:12:24.409904854+01:00[Europe Brussels]		Post Paid Simple
33b01e06-b2e1-4913-89f9-cebe2289c29e	460006623	9617199874166	Port Out	Open	Accept Feedback To Antium	2025-01-15T03:59:45.87086321+01:00[Europe Brussels]		Post Paid Simple

src/api/orderSearch.ts

↑	@@ -85,7 +85,7 @@ export const searchOrders = async (
85	85 return {
86	86 orders: formattedData,
87	87 lastEvaluatedKey: data.lastEvaluatedKey,
88	- totalCount: data.totalCount formattedData.length, // Add this line
88	+ totalCount: data.totalCount formattedData.length,
89	89 };
90	90 } catch (error) {
91	91 console.error("There was a problem with the fetch operation during search order :", error);

src/api/orderSearchAll.ts

↑	0 → 100644
1	+ import {
2	+ ApiEndpointsConfig,
3	+ OrderData,
4	+ OrderFormValues,
5	+ } from "../constants/Cnp.types";
6	+ import { formatString } from "../utils/stringUtils";
7	+
8	+ const hostname = window.location.hostname;
9	+ let env = hostname.split(".")[2];
10	+
11	+ if (hostname === "localhost") {
12	+ env = "local";
13	+ }
14	+
15	+ const apiEndpoints: ApiEndpointsConfig = {
16	+ local: import.meta.env
17	+ .VITE_CNP_BFF_DEV_POST_ORDER_SEARCH_ALL_API_LOCAL_URL as string,
18	+ dev: import.meta.env.VITE_CNP_BFF_DEV_POST_ORDER_SEARCH_ALL_API_URL as string,
19	+ e2etest: import.meta.env
20	+ .VITE_CNP_BFF_E2ETEST_POST_ORDER_SEARCH_ALL_API_URL as string,
21	+ e2estab: import.meta.env
22	+ .VITE_CNP_BFF_E2ESTAB_POST_ORDER_SEARCH_ALL_API_URL as string,
23	+ stage: import.meta.env.VITE_CNP_BFF_STAGE_POST_ORDER_SEARCH_ALL_API_URL as string,
24	+ uat: import.meta.env.VITE_CNP_BFF_UAT_POST_ORDER_SEARCH_ALL_API_URL as string,
25	+ lat: import.meta.env.VITE_CNP_BFF_LAT_POST_ORDER_SEARCH_ALL_API_URL as string,
26	+ trg: import.meta.env.VITE_CNP_BFF_TRG_POST_ORDER_SEARCH_ALL_API_URL as string,
27	+ prod: import.meta.env.VITE_CNP_BFF_PROD_POST_ORDER_SEARCH_ALL_API_URL as string,
28	+ };
29	+
30	+ const endpoint = apiEndpoints[env];
31	+
32	+ export const searchOrdersAll = async (
33	+ orderSearchCriteria: OrderFormValues,
34	+ accessToken: string
35	+) => {
36	+ try {
37	+ const filteredCriteria = Object.entries(orderSearchCriteria).reduce(
38	+ (acc, [key, value]) => {
39	+ if (value !== "" && value !== null) {
40	+ acc[key as keyof OrderFormValues] = value;
41	+ }

```

42 +         return acc;
43 +     },
44 +     {} as OrderFormValues
45 + );
46 +
47 + const response = await fetch(endpoint, {
48 +   method: "POST",
49 +   headers: {
50 +     "Content-Type": "application/json",
51 +     Authorization: `Bearer ${accessToken}`,
52 +   },
53 +   body: JSON.stringify({ ...filteredCriteria }),
54 + });
55 +
56 + if (!response.ok) {
57 +   const errorBody = await response.text();
58 +   throw new Error(
59 +     `Network response was not ok: ${response.status} ${response.statusText}. Body:
60 + ${errorBody}`
61 +   );
62 +
63 +   const data = await response.json();
64 +
65 +   const formattedData = data.map((item: OrderData) => ({
66 +     id: item.npRequestId,
67 +     msisdn: item.msisdn && item.msisdn.length > 0 ? item.msisdn[0] : "",

68 +     nprNumber: item.nprNumber,
69 +     orderType: formatString(item.orderType),
70 +     orderStatus: formatString(item.portingStatus),
71 +     stepName: item.stepName && item.stepName.length > 0 ? formatString(item.stepName) : "",
72 +     orderCreationTime: item.creationDate,
73 +     customerNumber: item.customerNumber || "",
74 +     complexity: formatString(item.serviceClass || ""),
75 +   }));
76 +
77 +   return {
78 +     orders: formattedData,
79 +   };
80 + } catch (error) {
81 +   console.error("There was a problem with the fetch operation during search order :", error);
82 +   throw error;
83 + }
84 + };
85 +

```

\ No newline at end of file

src/pages/orders/AnimatedDataGrid.tsx

+220 -15 Viewed

```

1 - import React, { useContext } from "react";
1 + import React, { useContext, useState } from "react";
2 import {
3   DataGrid,
4   GridColDef,
5   GridPaginationModel,

```

```

 6   6 |     GridRowParams,
 7   7 | +   GridToolbarContainer,
 8   8 |     GridValidRowModel,
 9   9 |   } from "@mui/x-data-grid";
10 10 |   import { ThemeContext } from "../../context/ThemeContext";
11 11 | -   import { Typography, Grow, Fade } from "@mui/material";
12 12 | -   import { customColors } from "../../constants/Cnp.types";
13 13 | +   import { Typography, Grow, Fade, Box, Button, Menu, MenuItem } from "@mui/material";
14 14 | +   import { customColors, OrderFormValues } from "../../constants/Cnp.types";
15 15 | +   import { motion } from "framer-motion";
16 16 | +   import SaveAltIcon from "@mui/icons-material/SaveAlt";
17 17 | +   import { searchOrdersAll } from "../../api/orderSearchAll";
18 18 |
19 19 | interface AnimatedDataGridProps {
20 20 |   rows: GridValidRowModel[];
21 21 |   ⇩   @@ -19,8 +23,203 @@ interface AnimatedDataGridProps {
22 22 |     handleRowClick: (params: GridRowParams) => void;
23 23 |     selectedRowIndex: string | number;
24 24 |     CustomPagination: React.ComponentType<object>;
25 25 |     +   searchCriteria: OrderFormValues;
26 26 |     +   accessToken: string | null;
27 27 |   }
28 28 |
29 29 |   + const CustomToolbar: React.FC<{ generateFullData: () => Promise<any> }> = ({ generateFullData })
30 30 |   +> {
31 31 |     +   const [anchorEl, setAnchorEl] = useState<null | HTMLElement>(null);
32 32 |     +
33 33 |       const handleMenuClick = (event: React.MouseEvent<HTMLElement>) => {
34 34 |         +   setAnchorEl(event.currentTarget);
35 35 |       };
36 36 |
37 37 |       const handleMenuClose = () => {
38 38 |         +   setAnchorEl(null);
39 39 |       };
40 40 |
41 41 |       const handleExportFullData = async () => {
42 42 |         try {
43 43 |           const data = await generateFullData();
44 44 |
45 45 |             if (!data || !Array.isArray(data.orders)) {
46 46 |               console.error("Invalid full data format:", data);
47 47 |               return;
48 48 |             }
49 49 |
50 50 |             const csvContent = generateCSV(data.orders);
51 51 |             downloadCSV(
52 52 |               csvContent,
53 53 |               `CNP-Order-Search-Full-Data-${new Date().toISOString().split("T")[0]}.csv`
54 54 |             );
55 55 |             } catch (error) {
56 56 |               console.error("Error exporting full data:", error);
57 57 |             }
58 58 |           };

```

```

60 +     const handlePrint = async () => {
61 +       try {
62 +         const data = await generateFullData();
63 +
64 +         if (!data || !Array.isArray(data.orders)) {
65 +           console.error("Invalid full data format:", data);
66 +           return;
67 +         }
68 +
69 +         const printWindow = window.open("", "_blank");
70 +
71 +         const tableHtml = generateTable(data.orders);
72 +         printWindow.document.write(`
73 +           <html>
74 +             <head>
75 +               <title>Print Data</title>
76 +               <style>
77 +                 @media print {
78 +                   @page {
79 +                     size: landscape;
80 +                   }
81 +                 }
82 +                 table {
83 +                   width: 100%;
84 +                   border-collapse: collapse;
85 +                   font-size: 10px;
86 +                 }
87 +                 th, td {
88 +                   border: 1px solid #ddd;
89 +                   padding: 4px;
90 +                   text-align: left;
91 +                 }
92 +                 th {
93 +                   background-color: ${customColors.primary};
94 +                   color: white;
95 +                 }
96 +               </style>
97 +             </head>
98 +             <body>${tableHtml}</body>
99 +           </html>
100 +         `);
101 +         printWindow.document.close();
102 +         printWindow.print();
103 +       }
104 +     } catch (error) {
105 +       console.error("Error preparing print data:", error);
106 +     }
107 +   };
108 +
109 +   const generateCSV = (data: any[]) => {
110 +     if (data.length === 0) return "";
111 +
112 +     const delimiter = ";";
113 +     const headers = Object.keys(data[0]);
114 +
115 +     const rows = data.map((row) =>

```

```

116 +     headers
117 +       .map((header) => {
118 +         const value = row[header];
119 +         if (value === null || value === undefined) return "";
120 +         return `${String(value).replace(/\g, '')}`;
121 +       })
122 +       .join(delimiter)
123 +     );
124 +
125 +   return `${headers.join(delimiter)}\n${rows.join("\n")}`;
126 + };
127 +
128 + const generateTable = (data: any[]) => {
129 +   if (data.length === 0) return "<p>No data available</p>";
130 +
131 +   const headers = Object.keys(data[0]);
132 +   const rows = data.map(
133 +     (row) =>
134 +       `<tr>${headers}
135 +         .map((header) => `<td>${row[header] ?? ""}</td>`)
136 +         .join("")`</tr>`
137 +   );
138 +
139 +   return `
140 +     <table>
141 +       <thead>
142 +         ${headers.map((header) => `<th>${header}</th>`).join("")}</tr>
143 +           </thead>
144 +           <tbody>
145 +             ${rows.join("")}
146 +           </tbody>
147 +         </table>
148 +       `;
149 + };
150 +
151 + const downloadCSV = (csvContent: string, fileName: string) => {
152 +   const bom = "\uFEFF";
153 +   const blob = new Blob([bom + csvContent], { type: "text/csv;charset=utf-8;" });
154 +   const link = document.createElement("a");
155 +   const url = URL.createObjectURL(blob);
156 +
157 +   link.setAttribute("href", url);
158 +   link.setAttribute("download", fileName);
159 +   link.style.visibility = "hidden";
160 +
161 +   document.body.appendChild(link);
162 +   link.click();
163 +   document.body.removeChild(link);
164 + };
165 +
166 + return (
167 +   <Box sx={{ padding: 2, textAlign: "center", position: "relative" }}>
168 +     <motion.div initial="hidden" animate="visible" transition={{ delay: 0.5 }}>
169 +       <Typography variant="h6" sx={{ fontWeight: "bold", color: customColors.primary }}>
170 +         Order Search Results

```

```

171 +      </Typography>
172 +    </motion.div>
173 +    <Box
174 +      sx={{
175 +        position: "absolute",
176 +        right: 16,
177 +        top: "50%",
178 +        transform: "translateY(-50%)",
179 +      }}
180 +
181 +    >
182 +      <GridToolbarContainer>
183 +        <Button
184 +          onClick={handleMenuClick}
185 +          startIcon={<SaveAltIcon />}
186 +          sx={{
187 +            color: "black",
188 +            backgroundColor: customColors.secondary,
189 +            borderColor: customColors.primary,
190 +            backgroundImage: `linear-gradient(45deg, ${customColors.secondary},
191 +            ${customColors.primary})`,
192 +            fontWeight: "bold",
193 +            padding: "8px 16px",
194 +            borderRadius: "4px",
195 +            cursor: "pointer",
196 +            "&:hover": { backgroundColor: customColors.primary },
197 +          }}
198 +
199 +        >
200 +          <Button>
201 +            <Menu anchorEl={anchorEl} open={Boolean(anchorEl)} onClose={handleMenuClose}>
202 +              <MenuItem
203 +                onClick={() => {
204 +                  handleExportFullData();
205 +                  handleMenuClose();
206 +                }}
207 +              >
208 +                Download as CSV
209 +              </MenuItem>
210 +              <MenuItem
211 +                onClick={() => {
212 +                  handlePrint();
213 +                  handleMenuClose();
214 +                }}
215 +              >
216 +                Print
217 +              </MenuItem>
218 +            </Menu>
219 +          </GridToolbarContainer>
220 +
221 +
222 +
223 const AnimatedDataGrid: React.FC<AnimatedDataGridProps> = ({{
224   rows,

```

```

26  225 |     columns,
27  ↴ |     @@ -30,24 +229,27 @@ const AnimatedDataGrid: React.FC<AnimatedDataGridProps> = ({
28  229 |       handleRowClick,
29  230 |       selectedRowIndex,
30  231 |       CustomPagination,
31  232 |       + searchCriteria,
32  233 |       + accessToken,
33  234 |     }) => {
34  235 |       const { isDarkMode } = useContext(ThemeContext);
35  236 |
36  237 |       const generateFullData = async () => {
37  238 |         try {
38  239 |           const dataAll = await searchOrdersAll(
39  240 |             searchCriteria,
40  241 |             accessToken || ""
41  242 |           );
42  243 |           return dataAll;
43  244 |         } catch (error) {
44  245 |           console.error(error);
45  246 |           throw error;
46  247 |         }
47  248 |       };
48  249 |
49  250 |       return (
50  251 |         <Fade in={true} timeout={800}>
51  252 |           <div>
52  253 |             <Typography
53  254 |               variant="h6"
54  255 |               align="center"
55  256 |               gutterBottom
56  257 |               sx={{
57  258 |                 fontWeight: "bold",
58  259 |                 color: customColors.primary,
59  260 |                 marginTop: "1rem",
60  261 |               }}
61  262 |             >
62  263 |               Retrieved Orders
63  264 |             </Typography>
64  265 |             <Grow in={true} timeout={1000}>
65  266 |               <div>
66  267 |                 <DataGrid
67  ↴ |                 @@ -71,6 +273,7 @@ const AnimatedDataGrid: React.FC<AnimatedDataGridProps> = ({
68  273 |                   autoHeight
69  274 |                   slots={{
70  275 |                     pagination: CustomPagination,
71  276 |                     + toolbar: () => <CustomToolbar generateFullData={generateFullData} />,
72  277 |                   }
73  278 |                   sx={{
74  279 |                     height: "100%",
75  ↴ |                     @@ -107,6 +310,8 @@ const AnimatedDataGrid: React.FC<AnimatedDataGridProps> = ({
76  310 |                       "& .MuiDataGrid-columnHeaders": {
77  311 |                         borderBottom: "1px solid",
78  312 |                         borderColor: isDarkMode
79  |

```

```

110 315 ? "rgb(113,121,126)"
111 316 : "rgb(234, 179, 8)",
112 317 },
↓

```

src/pages/orders/OrderSearchForm.tsx

```

↑ @@ -194,6 +194,11 @@ const OrderSearchForm: React.FC = () => {
194 194   localStorage.removeItem("hasMoreResults");
195 195 }
196 196
197 197 + const searchCriteria: OrderFormValues = {
198 198   ...formValues,
199 199   operator: capitalizeAndAddUnderscores(selectedBrand || ""),
200 200 };
201 201 +
202 202   const performSearch = useCallback(
203 203     async (
204 204       isNewSearch: boolean = true,
↓ ↑ @@ -768,6 +773,8 @@ const OrderSearchForm: React.FC = () => {
768 773   handleRowClick={handleRowClick}
769 774   selectedRowIndex={selectedRowIndex}
770 775   CustomPagination={CustomPagination}
776 776 +   searchCriteria={searchCriteria}
777 777 +   accessToken={accessToken}
778 778   />
779 779   ) : (
780 780     <motion.div
↓

```

.env

```

↑ @@ -126,6 +126,25 @@ VITE_CNP_BFF_TRG_POST_ORDER_SEARCH_API_URL=https://bff.cnp.trg.numberportability
126 126 VITE_CNP_BFF_PROD_POST_ORDER_SEARCH_API_URL=https://bff.cnp.prod.numberportability-
prod.awsprod.internal.telenet.be/v1/api/search-orders
127 127
128 128
129 129 + VITE_CNP_BFF_DEV_POST_ORDER_SEARCH_ALL_API_LOCAL_URL=http://localhost:3697/v1/api/searchorders
130 130 +
131 131 + VITE_CNP_BFF_DEV_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.dev.numberportability-
dev.awsnprd.external.telenet.be/v1/api/searchorders
132 132 +
133 133 + VITE_CNP_BFF_E2ETEST_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.e2etest.numberportability-
dev.awsnprd.external.telenet.be/v1/api/searchorders
134 134 +
135 135 + VITE_CNP_BFF_E2ESTAB_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.e2estab.numberportability-
dev.awsnprd.external.telenet.be/v1/api/searchorders
136 136 +
137 137 + VITE_CNP_BFF_STAGE_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.stage.numberportability-
stage.awsnprd.external.telenet.be/v1/api/searchorders

```

```

138 +
139 + VITE_CNP_BFF_UAT_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.uat.numberportability-
  stage.awsnprd.internal.telenet.be/v1/api/searchorders
140 +
141 + VITE_CNP_BFF_LAT_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.lat.numberportability-
  stage.awsnprd.internal.telenet.be/v1/api/searchorders
142 +
143 + VITE_CNP_BFF_TRG_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.trg.numberportability-
  stage.awsnprd.internal.telenet.be/v1/api/searchorders
144 +
145 + VITE_CNP_BFF_PROD_POST_ORDER_SEARCH_ALL_API_URL=https://bff.cnp.prod.numberportability-
  prod.awspod.internal.telenet.be/v1/api/searchorders
146 +
147 +
129 148 VITE_CNP_BFF_DEV_GET_ORDER_DETAILS_API_LOCAL_URL=http://localhost:3697/v1/api/fetch-porting-
  request
130 149
131 150 VITE_CNP_BFF_DEV_GET_ORDER_DETAILS_API_URL=https://bff.cnp.dev.numberportability-
  dev.awsnprd.external.telenet.be/v1/api/fetch-porting-request

```



Summary

During my internship at Telenet, I played an active role in the development of their modernized number porting systems, contributing to both backend and frontend solutions within the TOF squad of the Number Porting Tribe. This work was centered on replacing outdated infrastructure with a robust system built using AWS, ensuring improved performance and reliability.

On the backend, I worked with Spring Boot to implement APIs, debugged issues, and improved code quality. I also developed key components for the fixed number porting system, such as message transformers to convert XML to JSON and AWS Lambda functions to streamline message processing. Terraform was used to enhance the infrastructure's maintainability and scalability.

On the frontend, I contributed to features like a CSV export function, using Vue.js and React to deliver user-friendly solutions. Throughout the development process, I collaborated closely with team members, participated in agile sprints, attended stand-ups, and adhered to best practices such as conducting code reviews.

This experience provided me with valuable technical and collaborative skills, deepening my knowledge of modern software development practices and tools while contributing to a high-impact project.