Technical Articles

Sujin Appukuttan

August 6, 2020 | 4 minute read

Odata $batch processing part 2

3 3 2,796

This is the 2nd part of this Odata $batch processing chapter. Click here for Odata $batch processing part 1

So far we have seen :

1) The default behavior of a change set within a $batch .

2) How the change set can contain only one operation and where each change set is treated as single LUW

3) In order to have more than one operation in a change set, SAP GW framework provides a defer mode. This will change the default processing mode. This is done by flagging defer mode = X in changes et begin method.

For defer mode:

1) Method CHANGESET_BEGIN is redefined with cv_defer_mode = abap_true

2) Method CHANGESET_PROCESS is redefined to have logic to create material + extend material to the plant

3) All operations in a changeset is treated as one LUW

$batch Use case 8: Change set in defer mode

Below $batch request has 2 change sets:

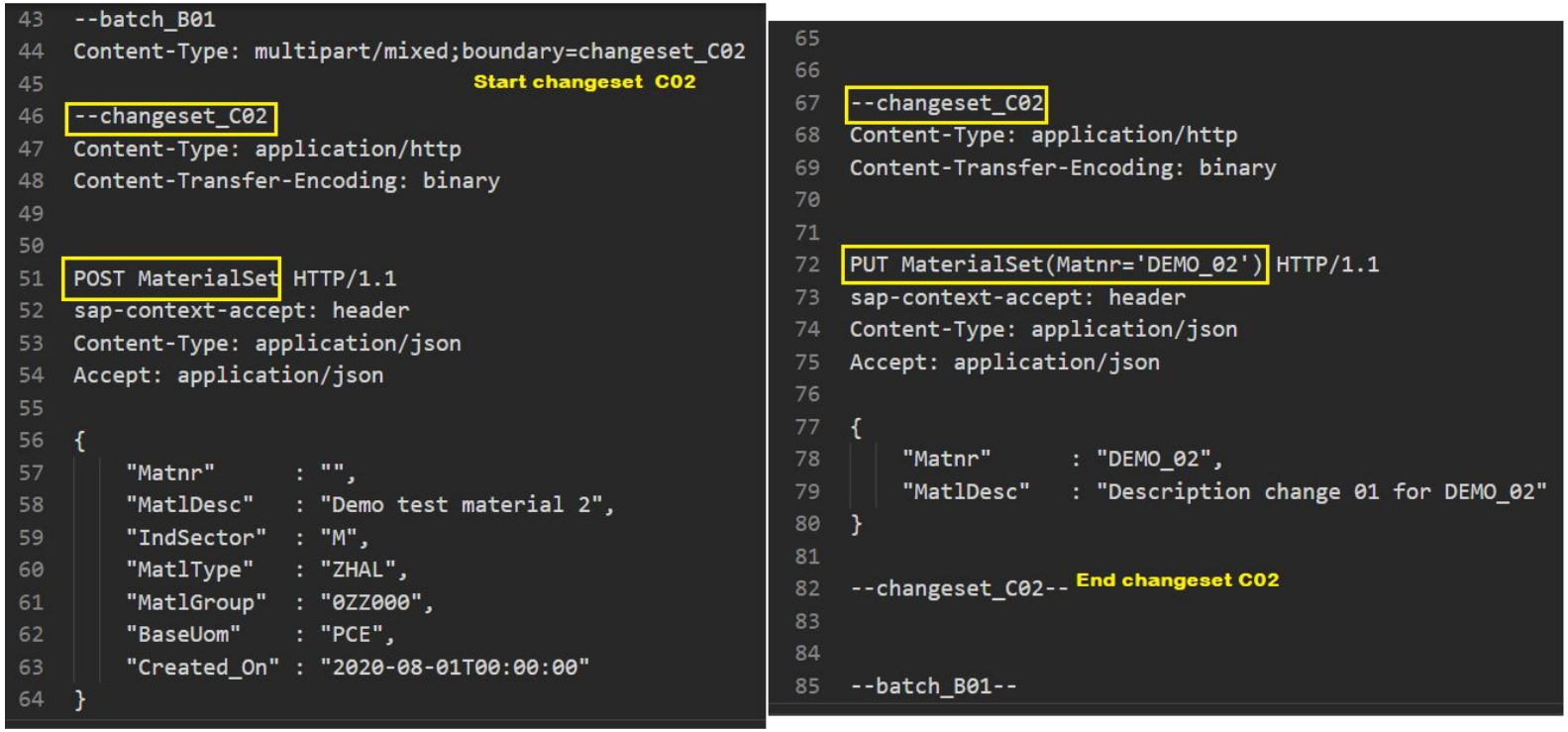
Change set C01 – 1 POST (for material creation) and 1 PUT (material update)

Change set C02 – 1 POST and 1 PUT

```
--batch_B01
Content-Type: multipart/mixed;boundary=changeset_C01
                                        Start changeset C01
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary


POST MaterialSet HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "Matnr"      : "",
    "MatlDesc"   : "Demo test material 1",
    "IndSector"  : "M",
    "MatlType"   : "ZHAL",
    "MatlGroup"  : "0ZZ000",
    "BaseUom"    : "PCE",
    "Created_On" : "2020-08-01T00:00:00"
}
```

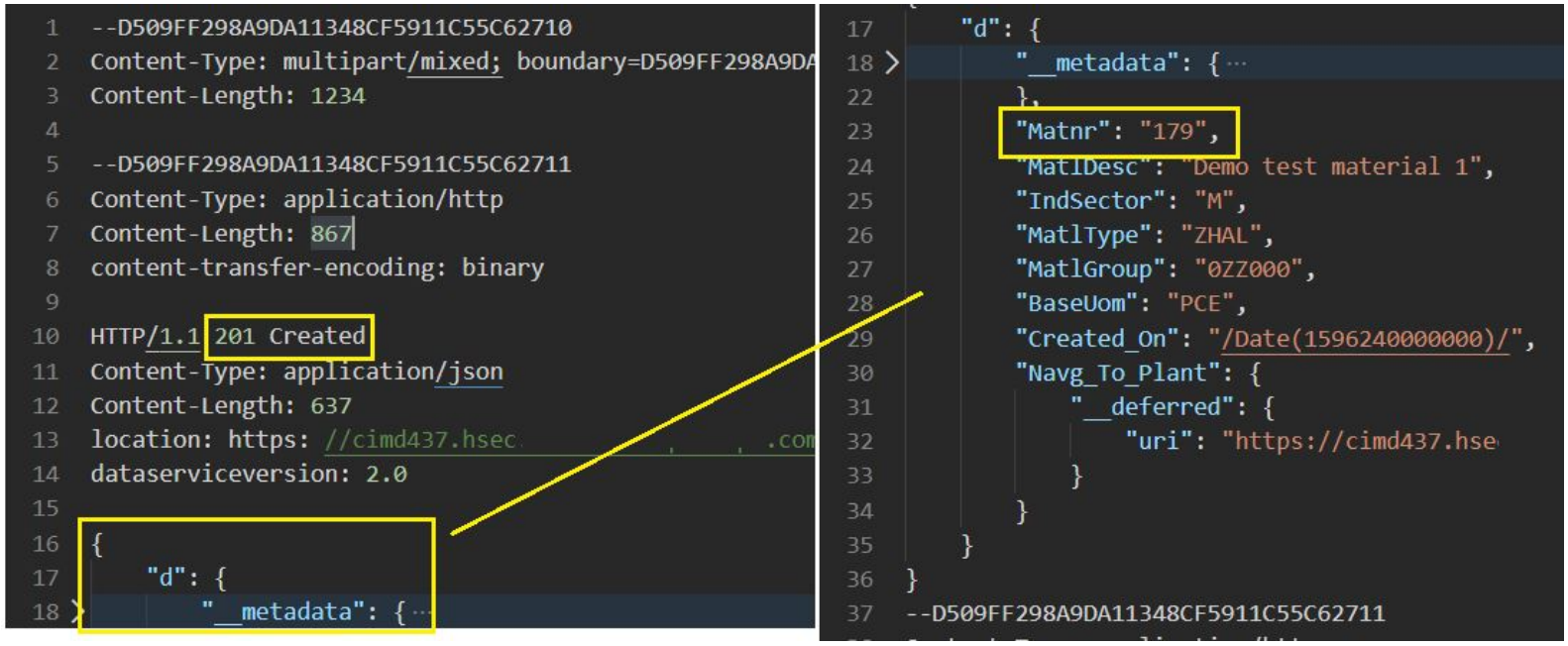
```

--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary


PUT MaterialSet(Matnr='DEMO_01') HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "Matnr"      : "DEMO_01",
    "MatlDesc"   : "Description change 01 for DEMO_01"
}

--changeset_C01--   End changeset C01

```

Change set C01

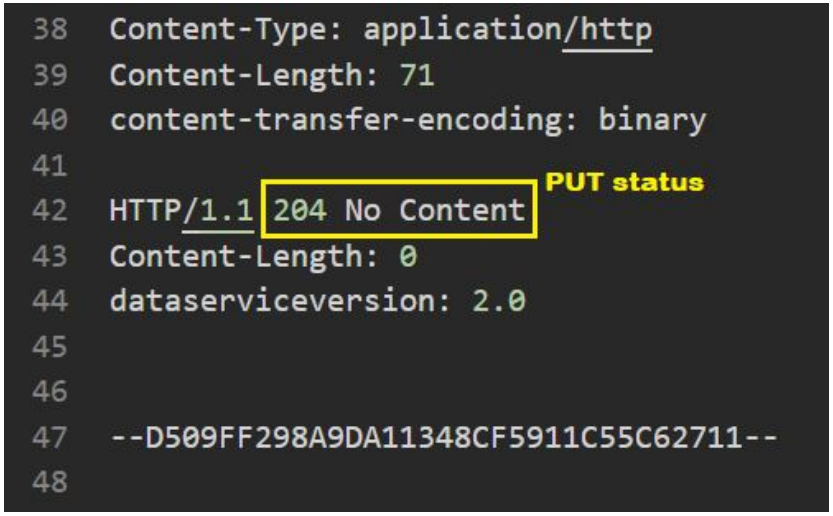


Change set C02

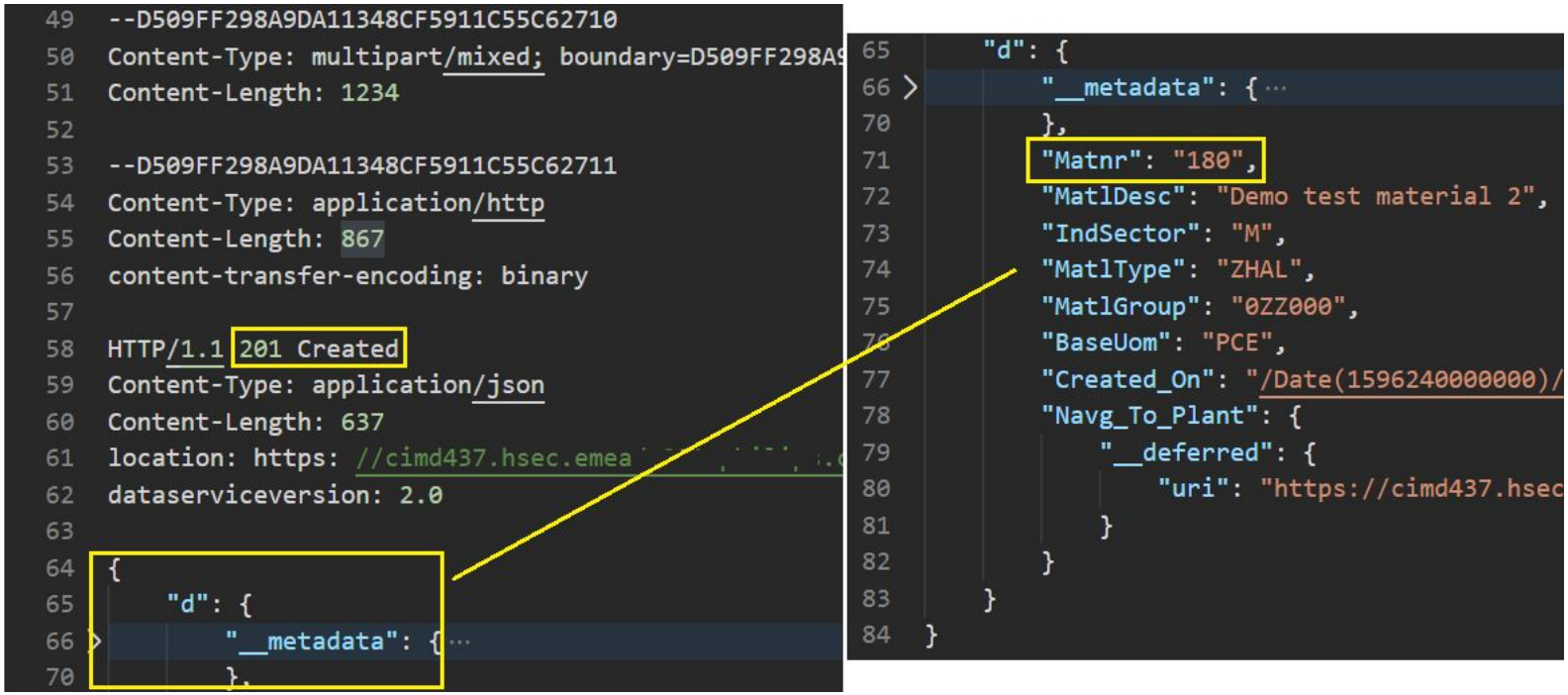
Response:



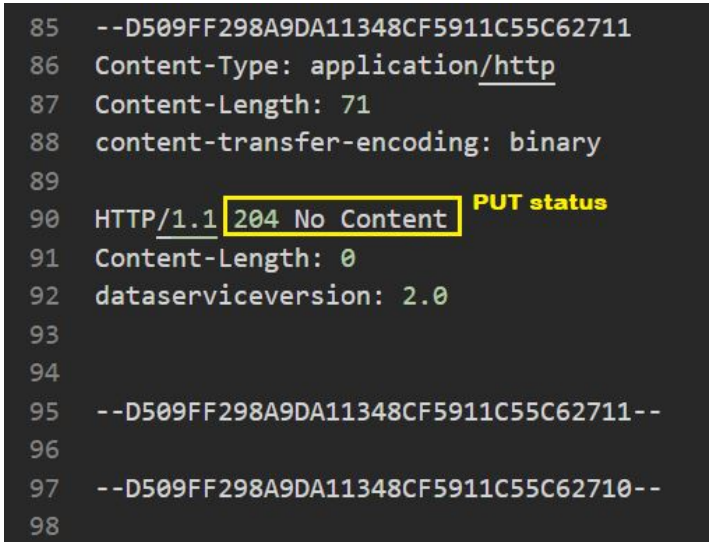
C01 POST response



C01 PUT response



C02 POST response



C02 PUT response

$batch Use case 9: Default processing vs Deferred processing

Deferred processing is meant to achieve performance gain. Allows more than one operation in a change set (single LUW). Let's compare the time between default & deferred processing.

To check the time consumed in postman, add header parameter "sap-statistics = true"

Below request for default processing mode contains:

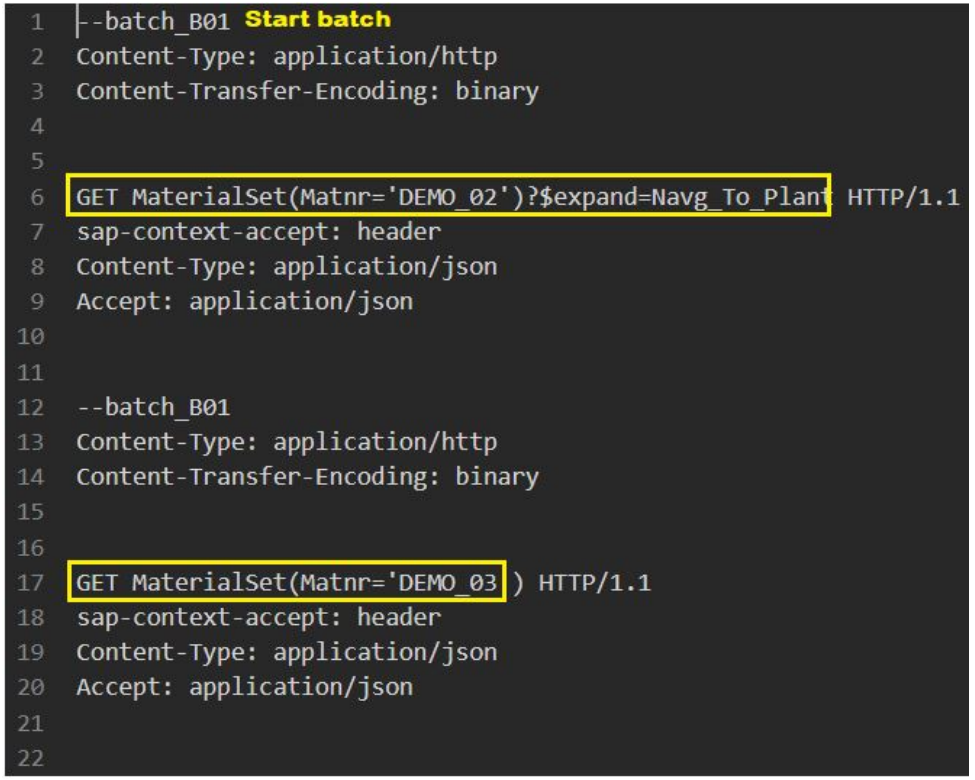
1) Two retrieve operations and

2) Two change sets as below:

C01 contain 1 POST operation

C02 contain 1 PUT operation

Request:



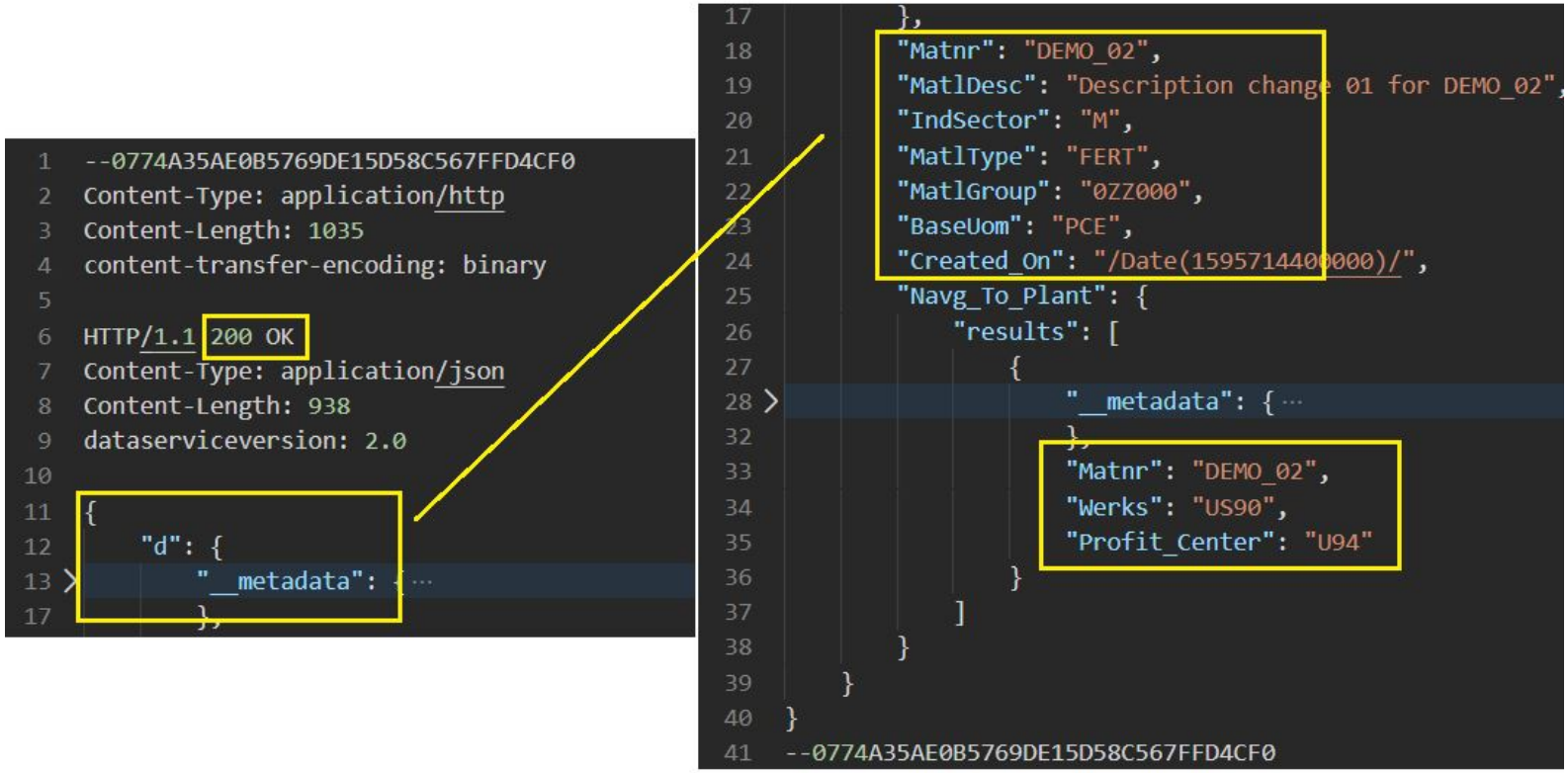
C01 GET request

```
--batch_B01
Content-Type: multipart/mixed;boundary=changeset_C01
                                    Start changeset C01
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary


POST MaterialSet HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "Matnr"      : "",
    "MatlDesc"   : "Demo test material",
    "IndSector"  : "M",
    "MatlType"   : "ZHAL",
    "MatlGroup"  : "0ZZ000",
    "BaseUom"    : "PCE",
    "Created_On" : "2020-08-01T00:00:00"
}


--changeset_C01-- End changeset C01
```

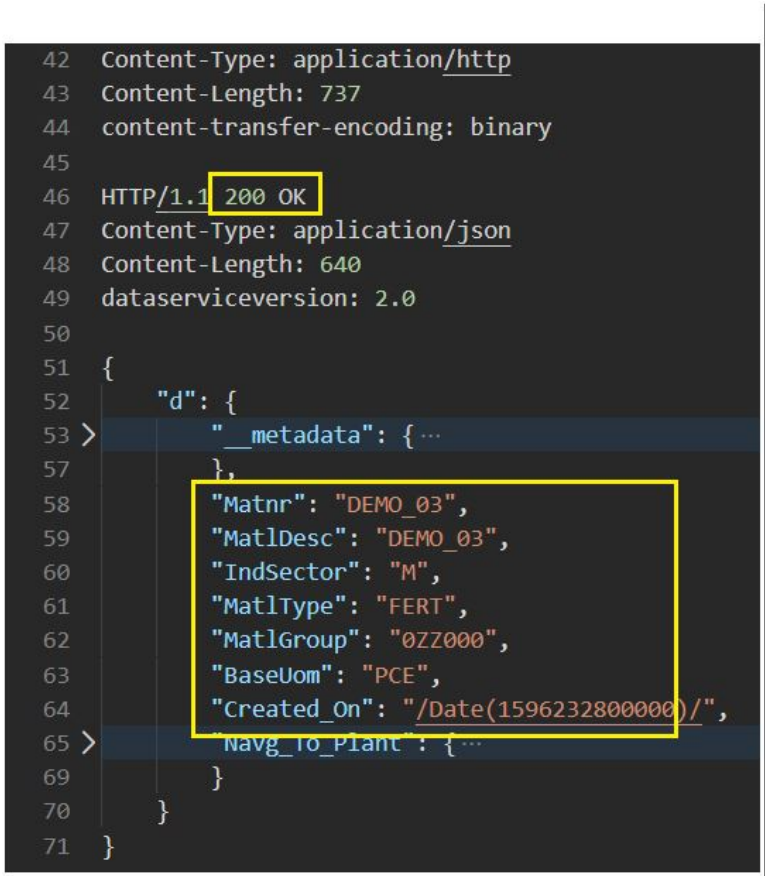
```


--batch_B01
Content-Type: multipart/mixed;boundary=changeset_C02
                                    Start changeset C02
--changeset_C02
Content-Type: application/http
Content-Transfer-Encoding: binary


PUT MaterialSet(Matnr='DEMO_04') HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "MatlDesc"   : "Change description DEMO_04"
}

--changeset_C02-- End of changeset C02


--batch_B01-- End batch
```

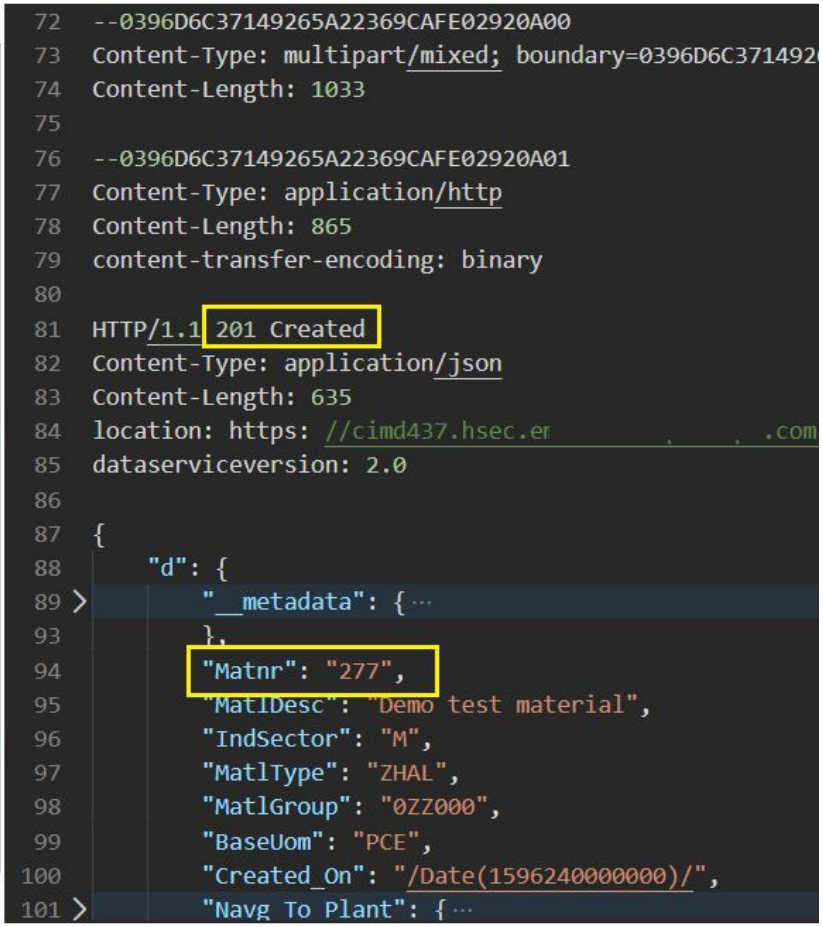
C01 POST & C02 PUT request

Response is as below:

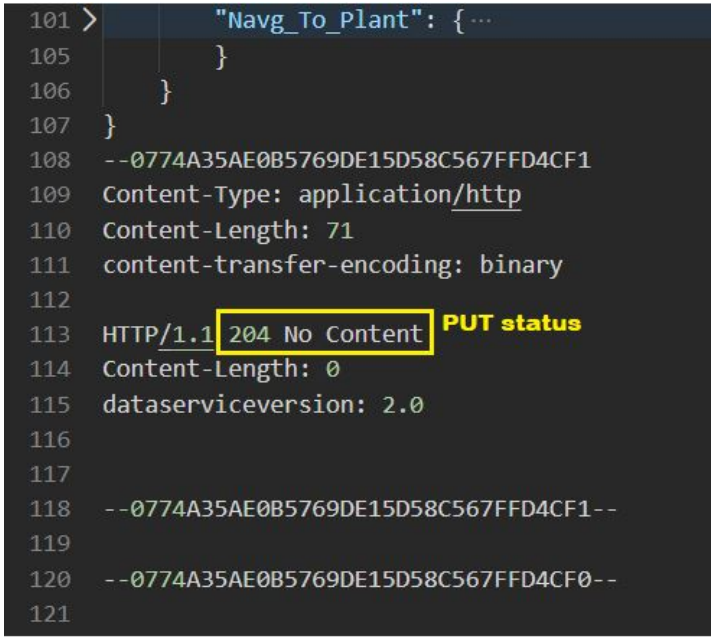


GET response



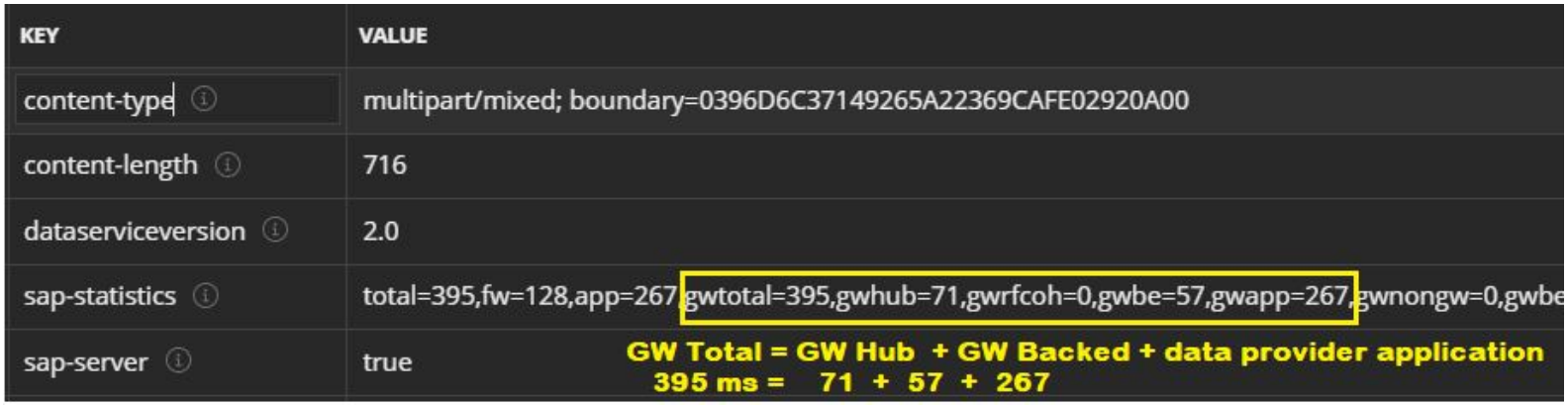


C01 POST response



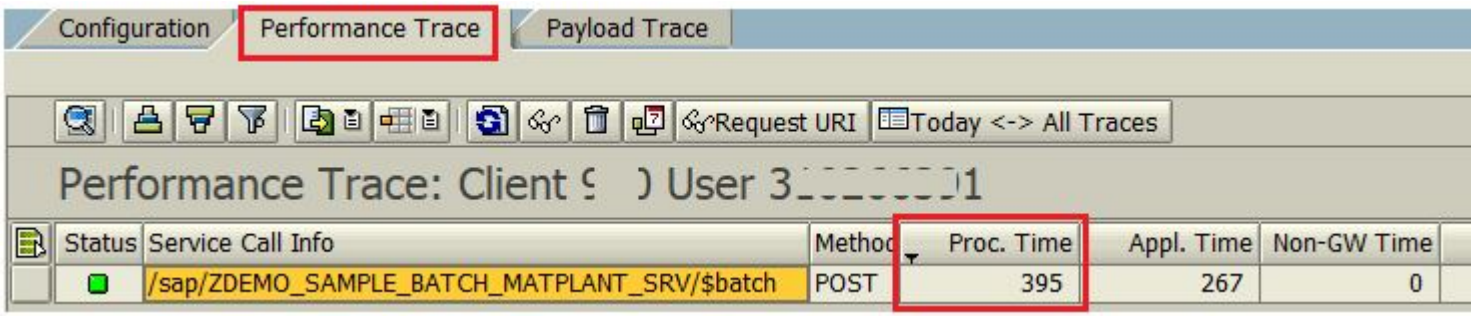
C01 PUT response

The time taken is :

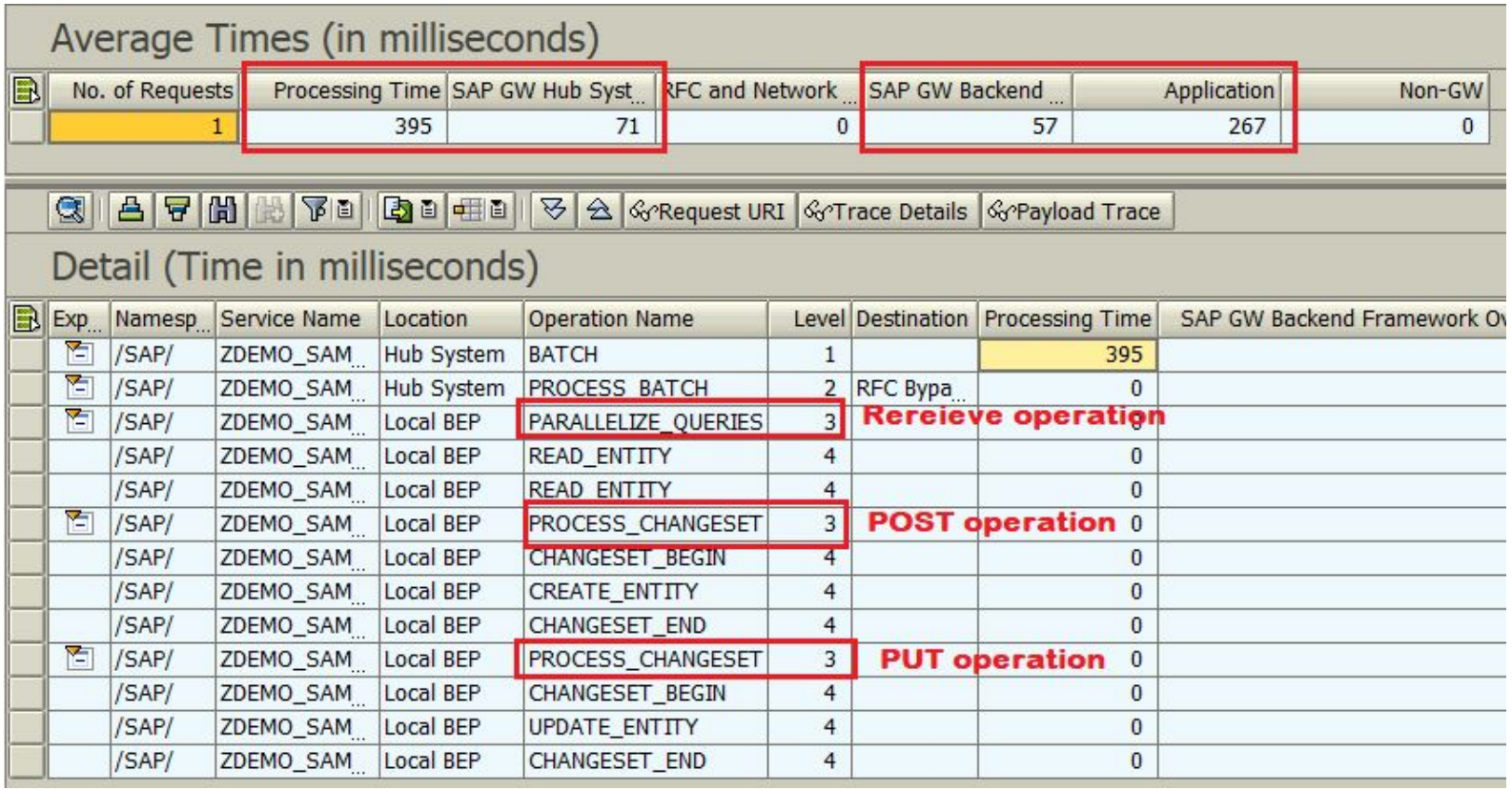


Time taken for default processing mode

Looking at tcode /IWFND/TRACES, we see the GW time taken is 395 milliseconds (ms)

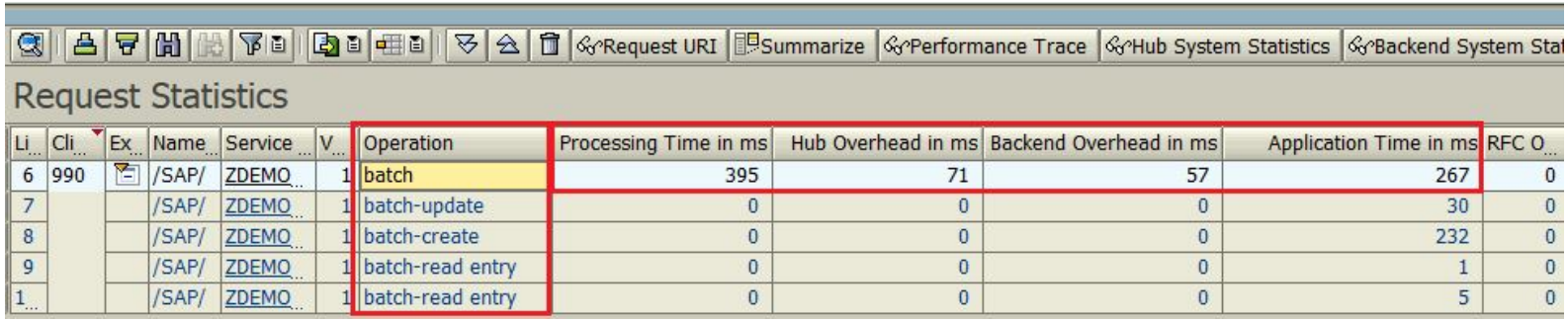


Frontend trace



Frontend trace details

Another good Gateway statistics tool (tcode /IWFND/STATS) will show the overall results as below:



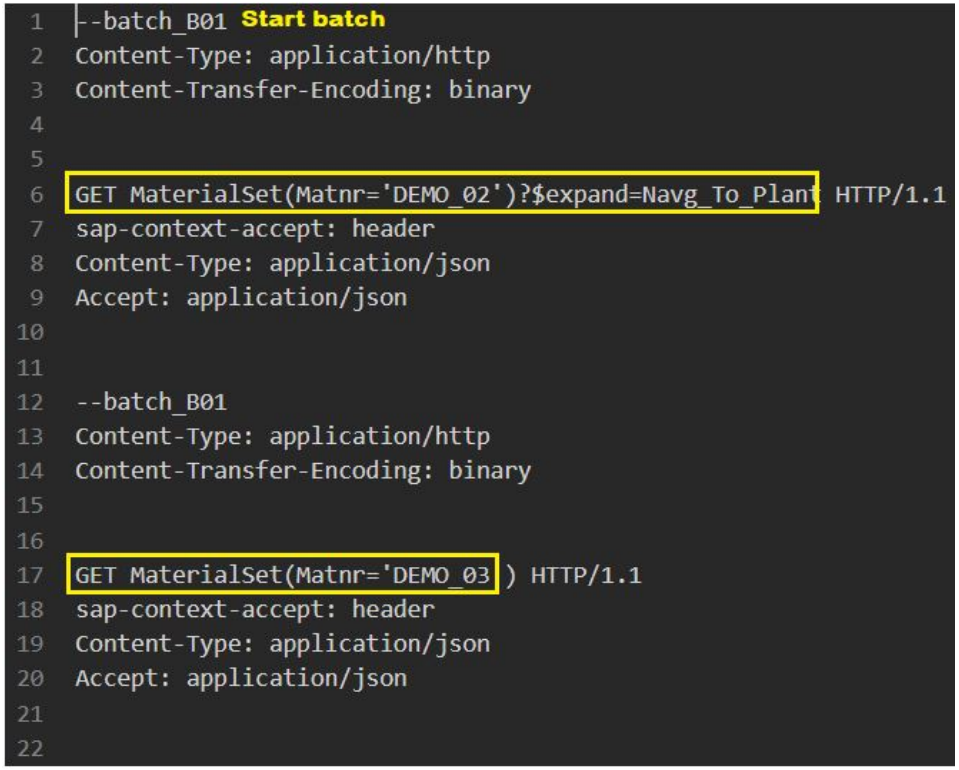
Li...	Cli...	Ex...	Name...	Service...	V...	Operation	Processing Time in ms	Hub Overhead in ms	Backend Overhead in ms	Application Time in ms	RFC O...
6	990		/SAP/	ZDEMO...	1	batch	395	71	57	267	0
7			/SAP/	ZDEMO...	1	batch-update	0	0	0	30	0
8			/SAP/	ZDEMO...	1	batch-create	0	0	0	232	0
9			/SAP/	ZDEMO...	1	batch-read entry	0	0	0	1	0
1...			/SAP/	ZDEMO...	1	batch-read entry	0	0	0	5	0

Gateway Statistics tool

More details on GW performance & measuring time can be found here.

Moving to the defer mode processing. The change set begin and change set process methods are implemented as mentioned before. Let's run earlier operations now in defer mode.

Request:


--batch_B01 Start batch
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MaterialSet(Matnr='DEMO_02')?$expand=Navg_To_Plant HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json
--batch_B01
Content-Type: application/http
Content-Transfer-Encoding: binary
GET MaterialSet(Matnr='DEMO_03') HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

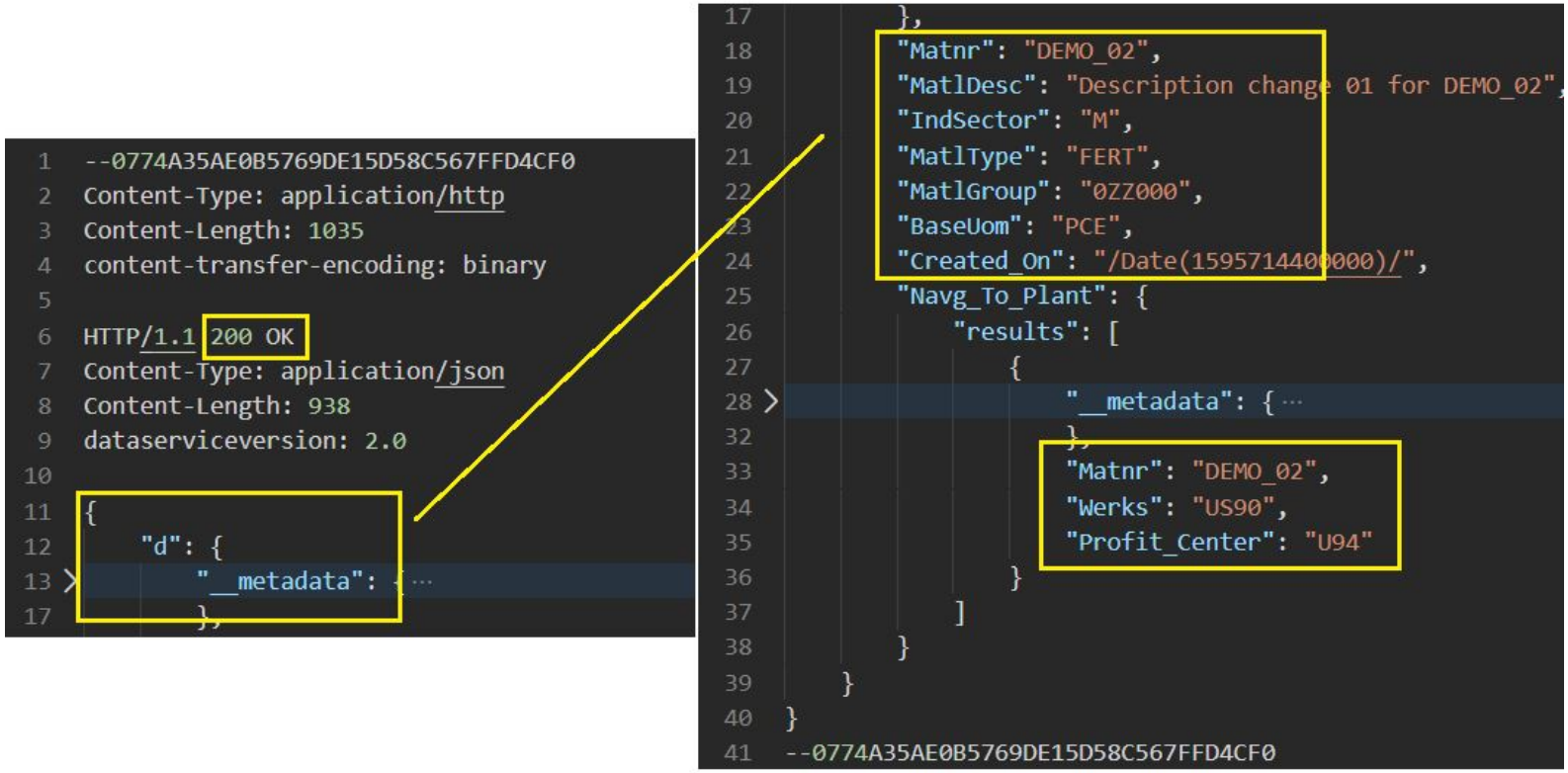
GET request

```
--batch_B01
Content-Type: multipart/mixed;boundary=changeset_C01
                                    Start changeset C01
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary


POST MaterialSet HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "Matnr"      : "",
    "MatlDesc"   : "Demo test material",
    "IndSector"  : "M",
    "MatlType"   : "ZHAL",
    "MatlGroup"  : "0ZZ000",
    "BaseUom"    : "PCE",
    "Created_On" : "2020-08-01T00:00:00"
}
```

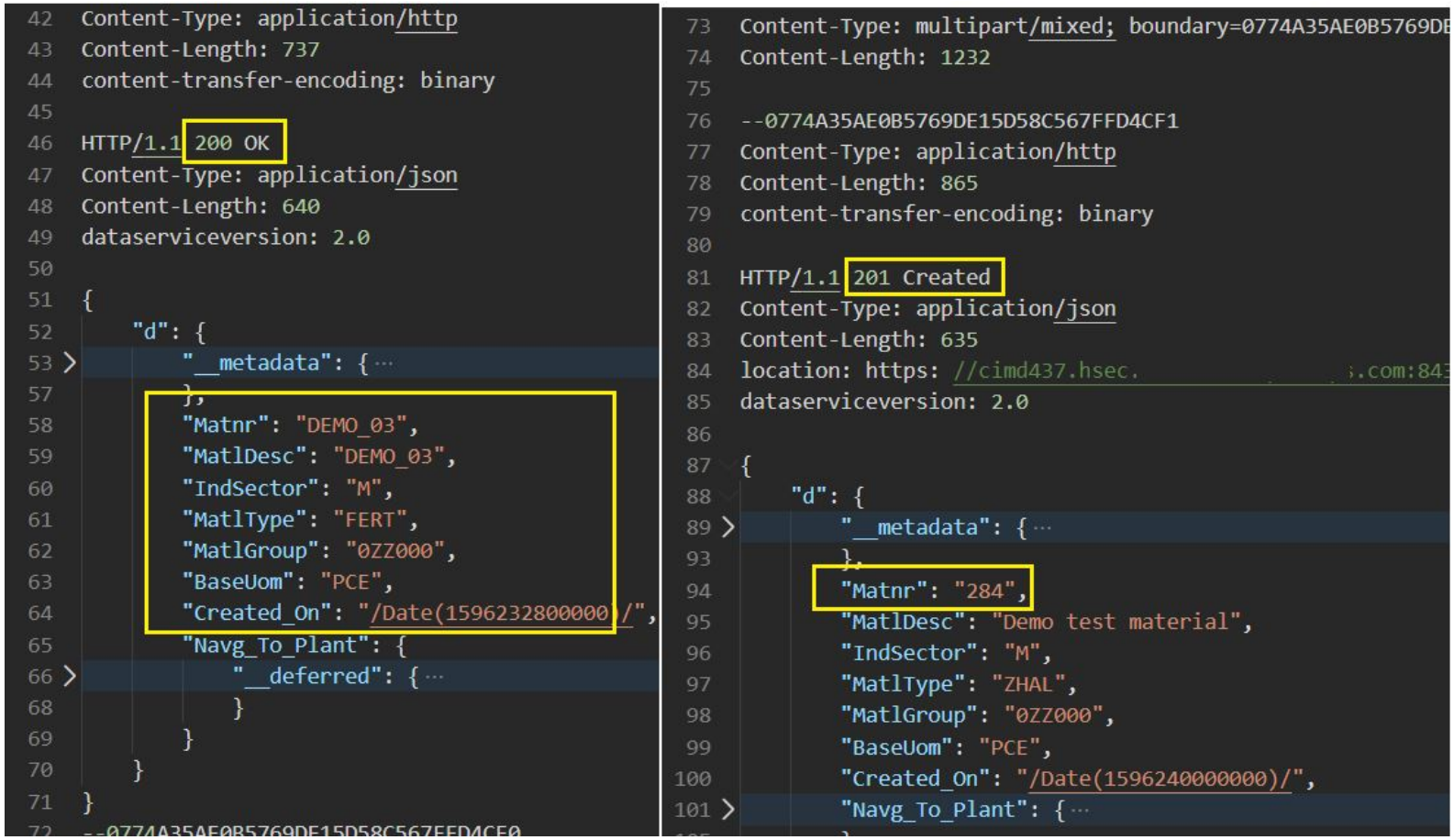
```
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary


PUT MaterialSet(Matnr='DEMO_04') HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "MatlDesc"   : "Change description 1 DEMO_04"
}

--changeset_C01--  End of changeset C01


--batch_B01--  End batch
```

C01 POST & PUT request

Response :



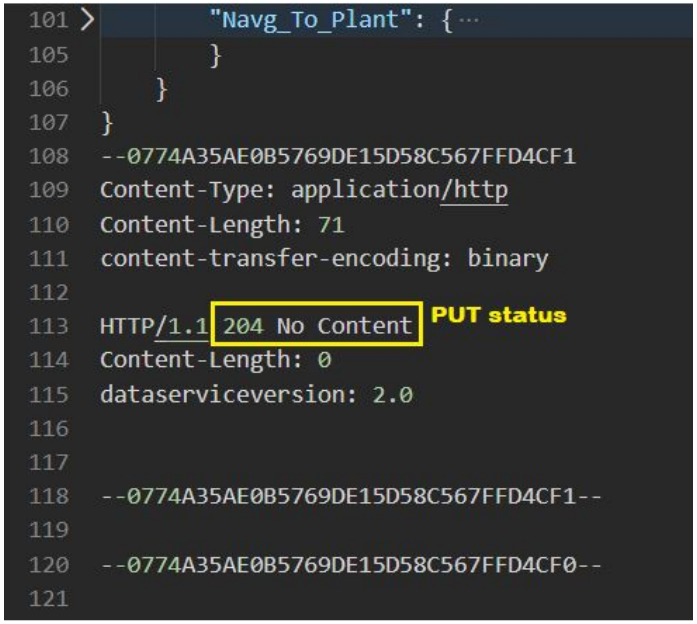
GET response



```
Content-Type: application/http
Content-Length: 737
content-transfer-encoding: binary

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 640
dataserviceversion: 2.0

{
    "d": {
        "__metadata": {…
        },
        "Matnr": "DEMO_03",
        "MatlDesc": "DEMO_03",
        "IndSector": "M",
        "MatlType": "FERT",
        "MatlGroup": "0ZZ000",
        "BaseUom": "PCE",
        "Created_On": "/Date(1596232800000)/",
        "Navg_To_Plant": {
            "__deferred": {…
            }
        }
    }
}
```

```
Content-Type: multipart/mixed; boundary=0774A35AE0B5769DE
Content-Length: 1232

--0774A35AE0B5769DE15D58C567FFD4CF1
Content-Type: application/http
Content-Length: 865
content-transfer-encoding: binary

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 635
location: https: //cimd437.hsec.                    .com:843
dataserviceversion: 2.0

{
    "d": {
        "__metadata": {…
        },
        "Matnr": "284",
        "MatlDesc": "Demo test material",
        "IndSector": "M",
        "MatlType": "ZHAL",
        "MatlGroup": "0ZZ000",
        "BaseUom": "PCE",
        "Created_On": "/Date(1596240000000)/",
        "Navg_To_Plant": {…
```

GET & C01 POST response



C01 PUT response

The time taken is:

KEY	VALUE
content-type	multipart/mixed; boundary=0774A35AE0B5769DE15D58C567FFD4CF0
content-length	709
dataserviceversion	2.0
sap-statistics	total=249,fw=52,app=197,gwtotal=249,gwhub=30,gwrfcoh=0,gwbe=22,gwapp=197,gwnongw=0,gwbewa
sap-server	true

GW Total = GW Hub + GW Backend + data provider application
249 ms = 30 + 22 + 197

Time taken for defer mode

Default behavior processing took 395 ms.

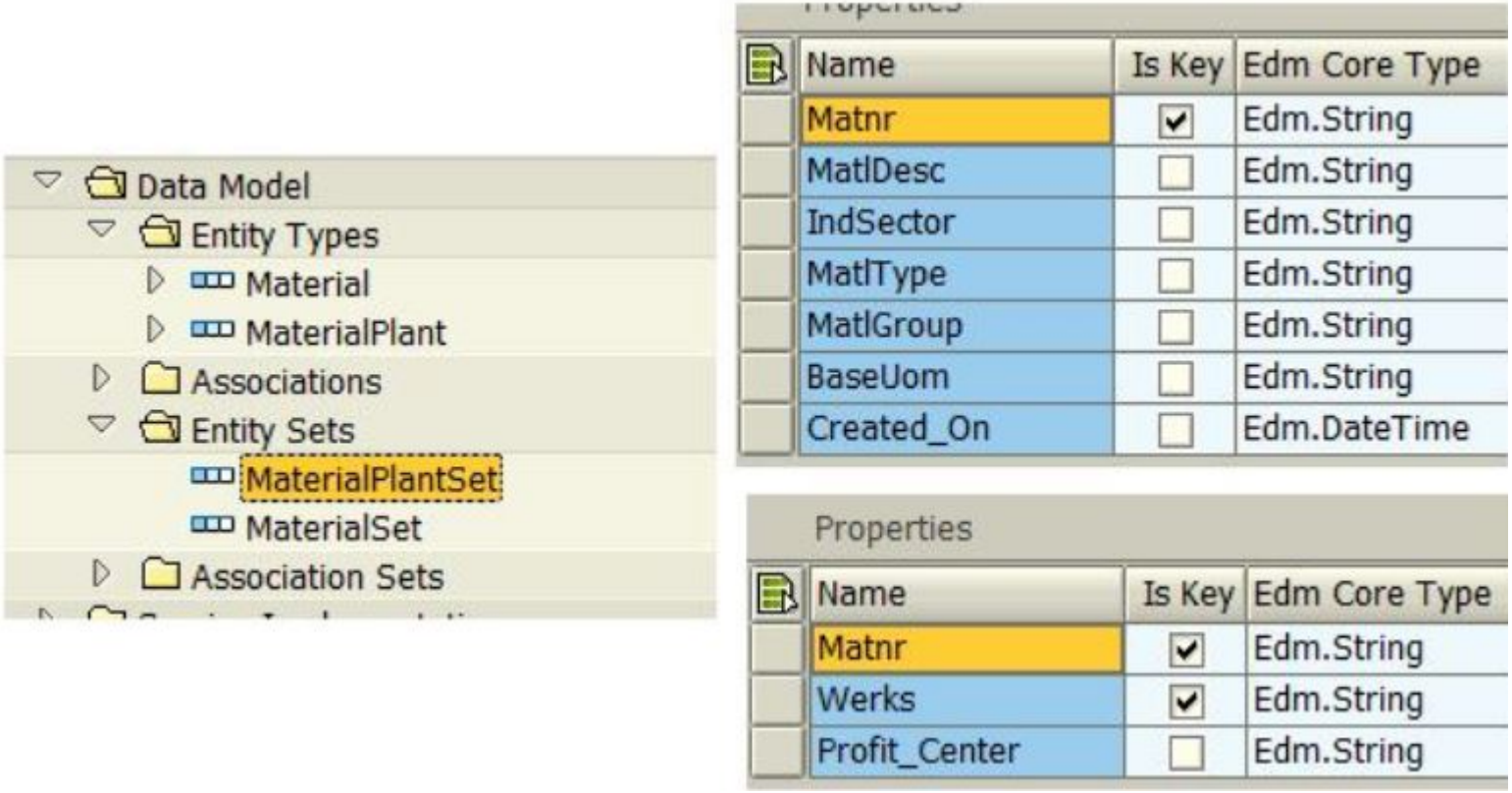
Deferred processing mode took 249 ms.

We see a performance gain factor of around 36%. The performance also depend on other factors like : system + network resources. For more number of operations, the deferred processing will have good performance due to its single LUW handling for a change set

$batch Use case 10: Content-ID usage

If there are multiple operations, one operation can refer to another using Content-ID reference instead of using entity key which may not be known at run time.

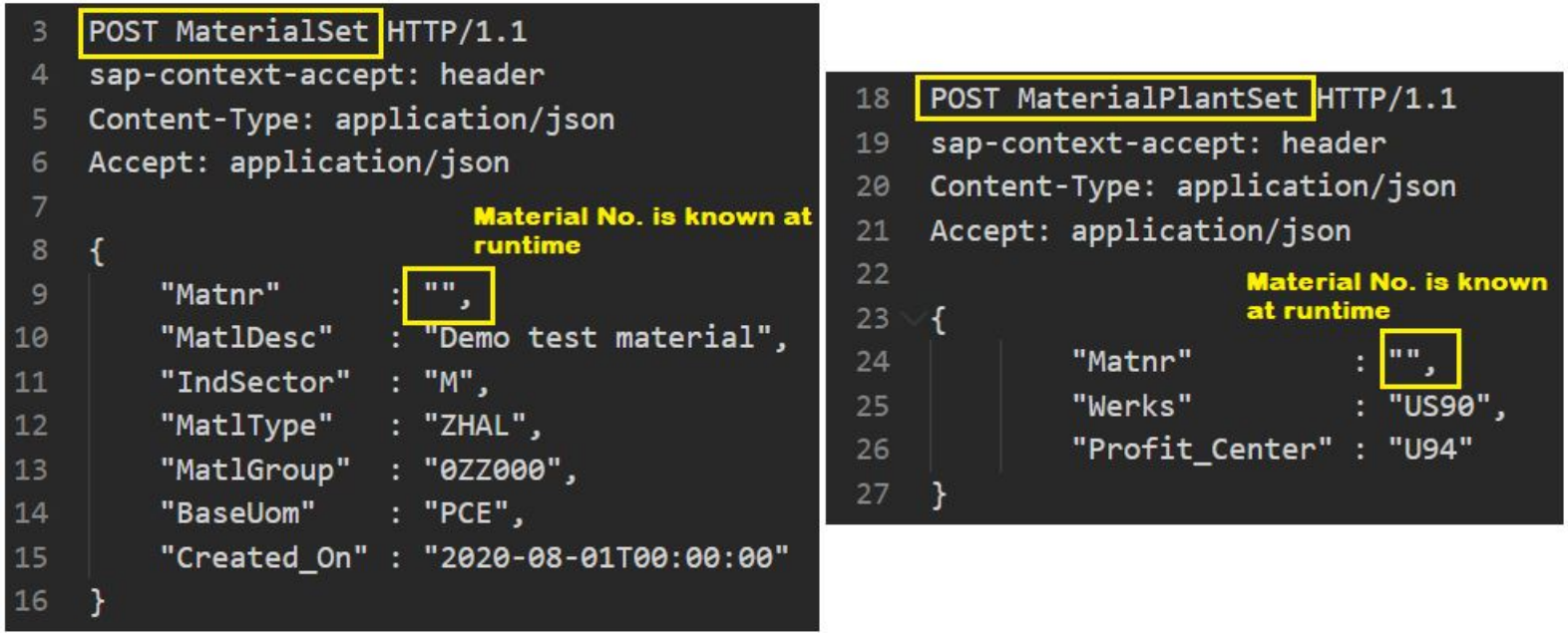
Say for example in our model, we have been using MaterialSet and MaterialPlantSet.



Odata model

Now if MaterialSet has a POST operation. Say MAT1 is newly created out of it. If I we were to run a POST on MaterialPlantSet for the newly generated MAT1, the MaterialPlantSet need to know about the newly created material at

runtime. That is achieved through Content-ID referencing.



Runtime relation between operation

Using Content-ID the 2nd operation can refer to the material generated in the 1st operation. Let's run the example.

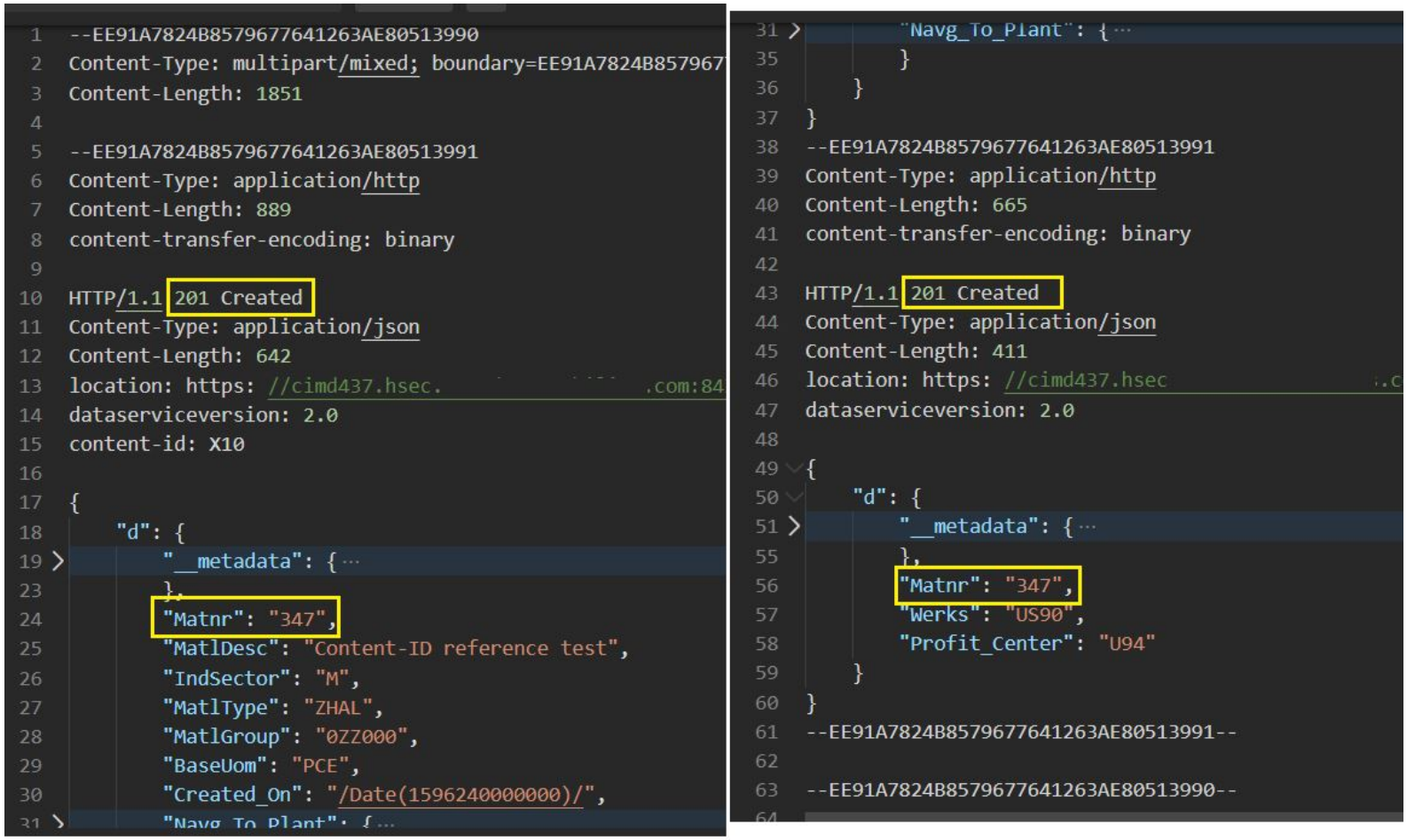
Content-ID can be used only in defer mode ie where more than one operation are present within a change set.

```
--batch_B01
Content-Type: multipart/mixed;boundary=changeset_C01

                    Start changeset C01
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary

POST MaterialSet HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json
Content-ID: X10

{
    "Matnr"      : "",
    "MatlDesc"   : "Content-ID reference test",
    "IndSector"  : "M",
    "MatlType"   : "ZHAL",
    "MatlGroup"  : "0ZZ000",
    "BaseUom"    : "PCE",
    "Created_On" : "2020-08-01T00:00:00"
}
```

```
--changeset_C01
Content-Type: application/http
Content-Transfer-Encoding: binary

POST $X10/Navg_To_Plant HTTP/1.1
sap-context-accept: header
Content-Type: application/json
Accept: application/json

{
    "Matnr"         : "",
    "Werks"         : "US90",
    "Profit_Center" : "U94"
}

--changeset_C01-- End changeset C01

--batch_B01--
```

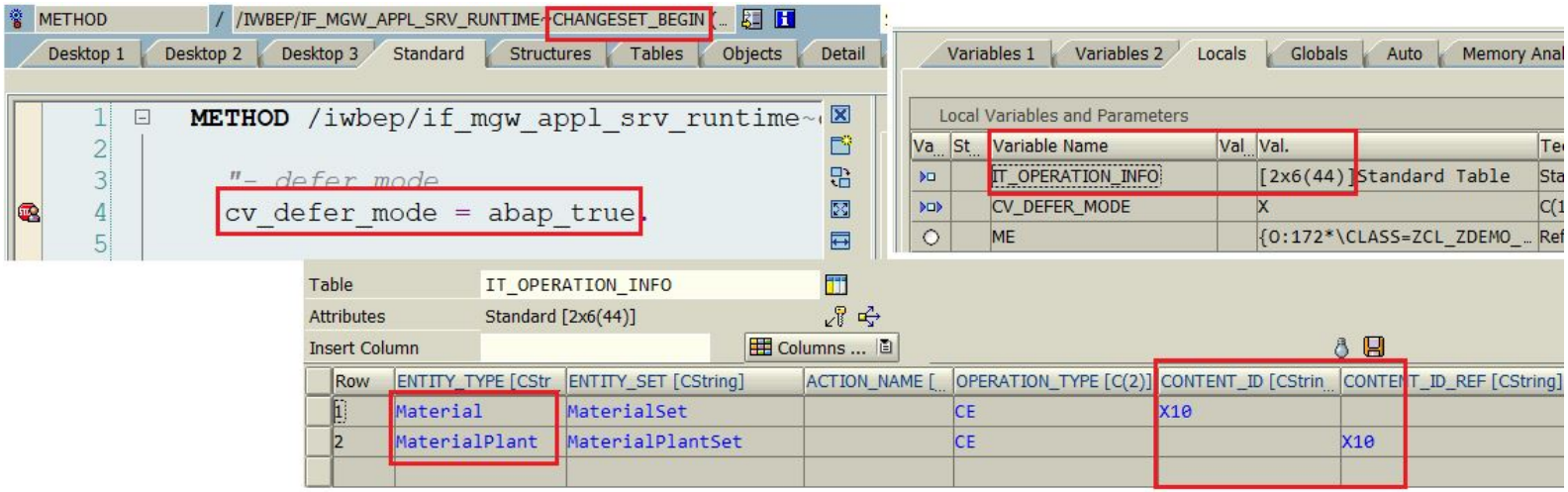
Content-ID request

Response :



Content-ID response

When the framework calls CHANGESET_BEGIN, we see it has collected two change set operations along with the Content-ID references



Debug – CHANGESET_BEGIN method

Alert Moderator

Further when CHANGESET_PROCESS is called, the two operations are executed.

Assigned tags

NW ABAP Gateway (OData)

OData

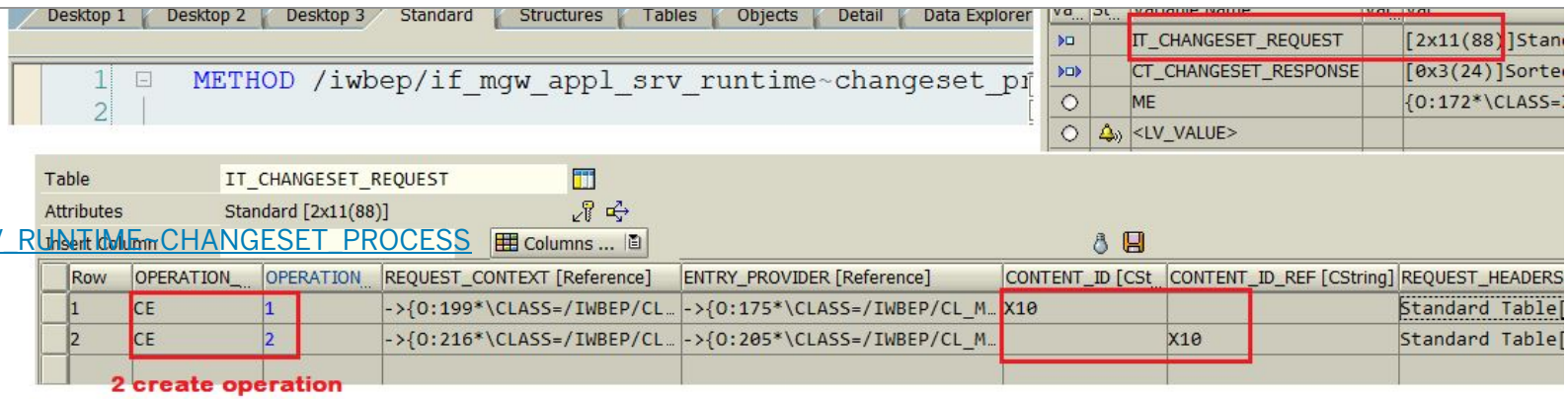
SAP Gateway

/IWBEP/IF_MGW_APPL_SRV_RUNTIME~CHANGESET_PROCESS

$batch

$batch changeset process

CRUD operations using BATCH in gateway



Debug – CHANGESET_PROCESS method

View more... 〉

Content-ID use will useful with scenarios like Sales Order header/Item or more like Purchase order/Item.

Similar Blog Posts

Odata $batch processing part1

So we pretty much covered the basics & advanced features of $batch requests using json.

By Sujin Appukuttan Aug 05, 2020

We did see the difference between default vs deferred processing. That will be all with the $batch request experiments. Hope the concepts will be clear and will prove beneficial to those using $batch requests. Thank you.

From OData Modelling To Service Execution using GWPA(Eclipse):PART-2

By Vishnu Pankajakshan Panicker Dec 11, 2014

$BATCH request in SAP GATEWAY

By Manikandan Rajasekaran May 06, 2018

Related Questions

Deferred mode versus Non-deferred mode batch request processing? in SAP NW gateway Odata

By Avinash D M Feb 07, 2019

SAP ODATA MULTIPLE UPDATE

By Ankit Shukla Sep 19, 2019

ABAP: Odata batch calls with json string and attachment content

By Rajesh Kumar Jul 17, 2019

3 Comments

You must be Logged on to comment or reply to a post.

Pavel Lazhbanau

September 16, 2020 at 2:41 pm

Hi Sujin

Good overview!

Like 0 | Share

S. Appukuttan

February 3, 2021 at 11:22 am

Thank you

Like 0 | Share

Sooraj Manjery

August 25, 2021 at 9:09 am

Hi,

Can you please have a look at the below query?

https://answers.sap.com/questions/13464045/odata-delete-triggering-separately-in-batch-operat.html

Thanks,

Sooraj

Like 0 | Share

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support