Resources:
Make fields visible

create RUM and match event to RUM

map event to service

I. Make fields visible inside BRM

1. In Storable Class editor, click File - Generate Custom Fields Source to create a header file, a property file, and multiple java files for the class.

2. Copy the header file into the BRM server

3. Run: parse_custom_ops_fields.pl -L pcmc -I custom_flds.h -O nextai_fields.dat

   Verify that the output should be:
   Number of opcodes defined: 0,
   Total size of the opcodes: 0,
   Number of fields defined: 6,
   Total size of the fields: 107
   Number of masked fields defined: 0,
   Total size of the masked fields: 0
   file size: 267
   Done parsing custom ops fields with no error found.

4. Inside the pin.conf of the CM and testnap, and pin_rum .conf, add the following entry:
   - - ops_fields_extension_file /opt/app/brm/congero-XX/BRM/nextai_fields.dat

   *note: use absolute path

5. Place/copy the header file inside BRM/include

II. Create 2 new RUMs

1. Navigate to: sys/data/pricing/example and modify pin_rum
- Add 2 lines:

   /event/session/usage : Tokens :
   (PIN_FLD_NEXTAI.PIN_FLD_TOKENS_IN+PIN_FLD_NEXTAI.PIN_FLD_TOKENS_OUT)/1000 : none

   /event/session/usage : Prompt : 1 : none

- Meaning:
    + Inside the event type /event/session/usage, we declare a RUM name called Token, that is calculated by adding PIN_FLD_TOKENS_IN and PIN_FLD_TOKENS_OUT, and the unit of measure is token
    + Similarly, for prompt, the calculation is only 1 since each LOAD_SESSION call only consume 1 prompt

2. Save and close the file

3. Load the pin rum to the BRM
   load_pin_rum -d -v pin_rum

4. Restart CM

II. Match event to RUM

5. Go to pin_usage_map file in sys/data/pricing/example and add:
   # for Tokens event
   /event/session/usage : Tokens : 0: 0: 0: 0: 0: 0: 0: token_rate_plan

   # for Prompt event
   /event/session/usage : Prompt : 0: 0: 0: 0: 0: 0: 0: prompt_rate_plan

6. Save, exit, and load the file
   load_usage_map -d -v pin_usage_map

   The output should be:
   load_usage_map:open transaction
   load_usage_map:delete old /config/usage_map/system object(s)
   load_usage_map:create usage_map object
   load_usage_map:commit transaction

7. Restart CM

III. Match event to service

8. Go to BRM and cd to: sys/data/pricing/example/
   Modify the pin_event_map to map the event to our service class
   /service/nextai : /event/session/usage : nextai usage event

9. Load our event mapping and restart CM
   load_event_map -d -v pin_event_map

IV. Create a new resource id via BRM

10. Go to this file: BRM/sys/data/pricing/example/pin_beid and add the custom resource id:

    1000300 0 0 0.000000 0.000000 0.000000 Mu Mu * 0 6 1 MU vo dich 366 1

    For our use case, append:

    1000301 0 0 0.000000 0.000000 0.000000 Tu Tu * 0 6 1 Token Used 366 1
    1000302 0 0 0.000000 0.000000 0.000000 Pu Pu * 0 6 1 Prompt Used 366 1

11. Then run:
    load_pin_beid -d -v pin_beid

    The output should be:

    load_pin_beid: beid file parsed
    load_pin_beid: pcm_connection open
    load_pin_beid: transaction open
    load_pin_beid: old beid object pruned
    load_pin_beid: new beid object created
    load_pin_beid: transaction committed
    load_pin_beid: pcm_connection closed

12. Verify the changes in pricing center

IV. Save the class as an podl file

13. Inside the BRM, run:

```
pin_deploy class -mncp /event/session/usage >
event_session_usage.podl
pin_deploy class -mncp /service/nextai > service_nextai.podl
```

14. Transfer the .podl to the local machine