

```

/* Sample of C function to create SO file and use Postgres SQL API
 * used only as helloworld example
 */
#include "postgres.h"
#include "postgres_ext.h"
#include "executor/spi.h" /* this is what you need to work with SPI */
#include "commands/trigger.h" /* TriggerData */
#include "access/tupdesc.h"
#include "utils/rel.h"
#include "nodes/bitmapset.h"
#include "catalog/pg_attribute.h"

#include "fmgr.h"
#include "utils/builtins.h"
#include "utils/guc.h" // GetConfigOptionByName
#include "utils/elog.h"
#include <string.h>
#include <math.h>

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>

#include <time.h>

PG_MODULE_MAGIC;

PG_FUNCTION_INFO_V1(test1);

Datum
test1(PG_FUNCTION_ARGS)
{
    TriggerData * trigdata = (TriggerData *) fcinfo->context;
    TupleDesc tupdesc = trigdata->tg_relation->rd_att;
    HeapTuple rettuple;

    /* tuple to return to executor */
    if (TRIGGER_FIRED_BY_UPDATE(trigdata->tg_event)) {
        rettuple = trigdata->tg_newtuple;
    } else {
        rettuple = trigdata->tg_trigtuple;
    }
    // = SPI_getnspname(trigdata->tg_relation)); // schema name
    // = SPI_getrelname(trigdata->tg_relation)); // relation name

    // config_file = GetConfigOptionByName("rust.file_path", NULL, true);
    // file path set as
    // sql> set rust.file_path = '/home/vk/git/rust_test/data';

    // NEEDS to get all PK(s),
    // ! SPI_execute is NOT permitted
    // something like this:
    // trigdata->tg_relation->rd_pkindex

    // total_attr = tupdesc->natts;
    // # for a_id from 0 to total_attr-1 do
    // SPI_fname(tupdesc, a_id);
    // SPI_gettype(tupdesc, a_id);
    // SPI_getvalue(rettuple, tupdesc, a_id);

```

```
    elog(INFO, "[test1 called]");  
    return PointerGetDatum(rettuple);  
}
```