

*My***ETS**

**Getting millions of config records from
MySQL to ETS**

AGENDA

- 1. How we use configuration data?**
- 2. Legacy solution**
- 3. New solution**
- 4. What is MyETS?**
- 5. How to use it**
- 6. Benefits**

CONFIGURATION MGMT

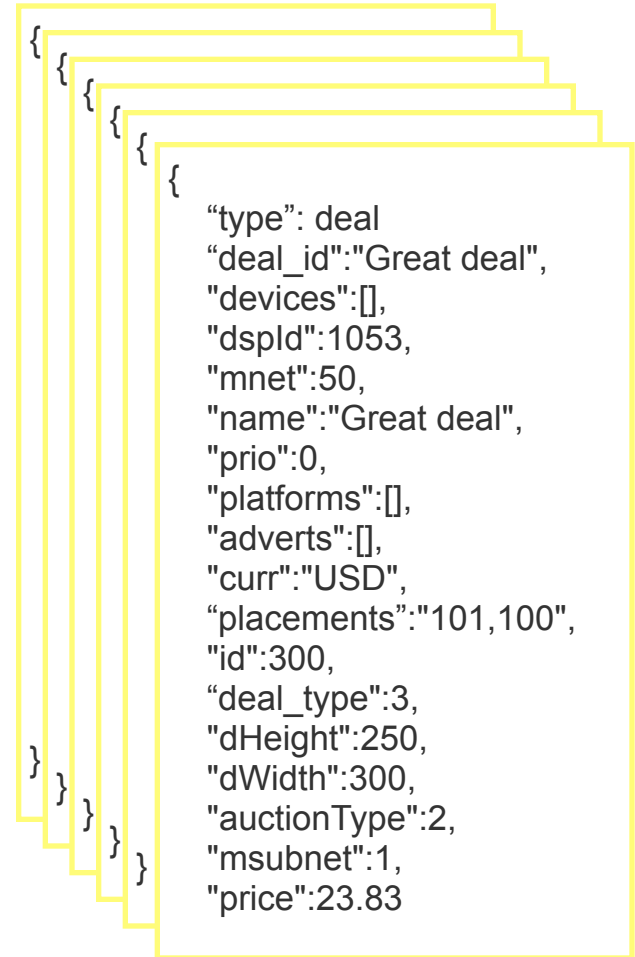
- **JSON** documents
- Several types
- Each doc has simple, **single id**
- Each doc has **type** field
- Documents are **independent**
- **No deleting**

```
{  
  "type": "deal,"  
  deal_id": "123SA",  
  operation: "update",  
  "devices": [],  
  "dspld": 1053,  
  "mnet": 50,  
  "name": "Great deal",  
  "prio": 0,  
  "platforms": [],  
  "adverts": [],  
  "curr": "USD",  
  "placements": "101,100",  
  "id": 300,  
  "deal_type": 3,  
  "dHeight": 250,  
  "dWidth": 300,  
  "auctionType": 2,  
  "msubnet": 1,  
}
```

CONFIGURATION MGMT

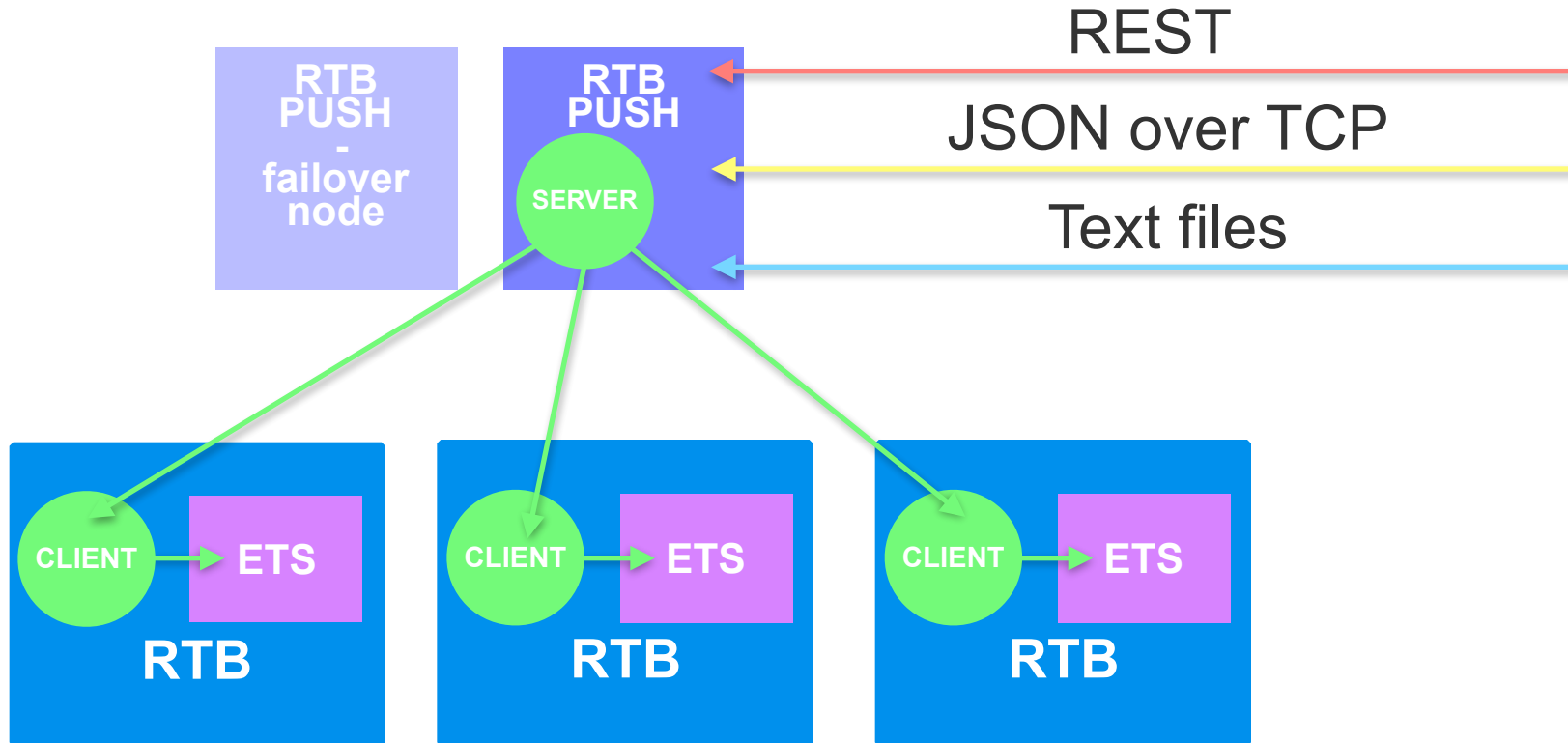
CHALLENGES

- **Millions** of documents
- **Hundreds** of machines
- Change in config should be delivered **instantly** to **all** nodes



CONFIGURATION MGMT

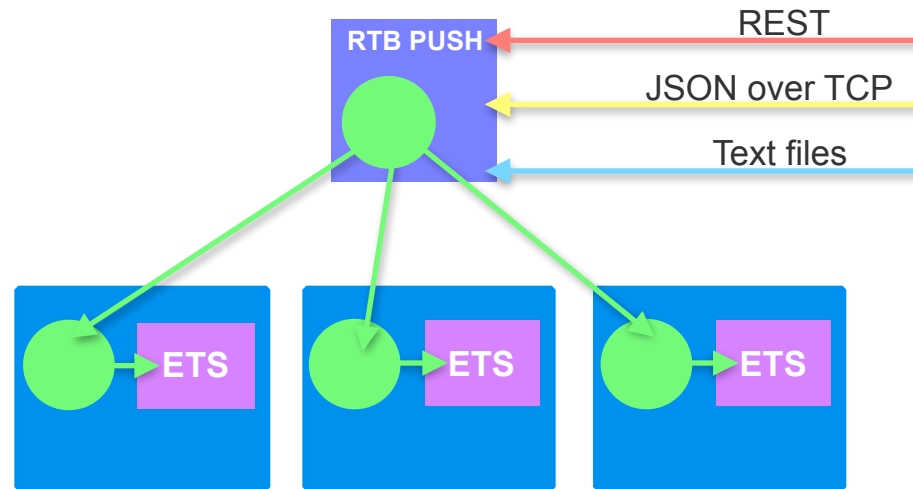
FIRST APPROACH



CONFIGURATION MGMT

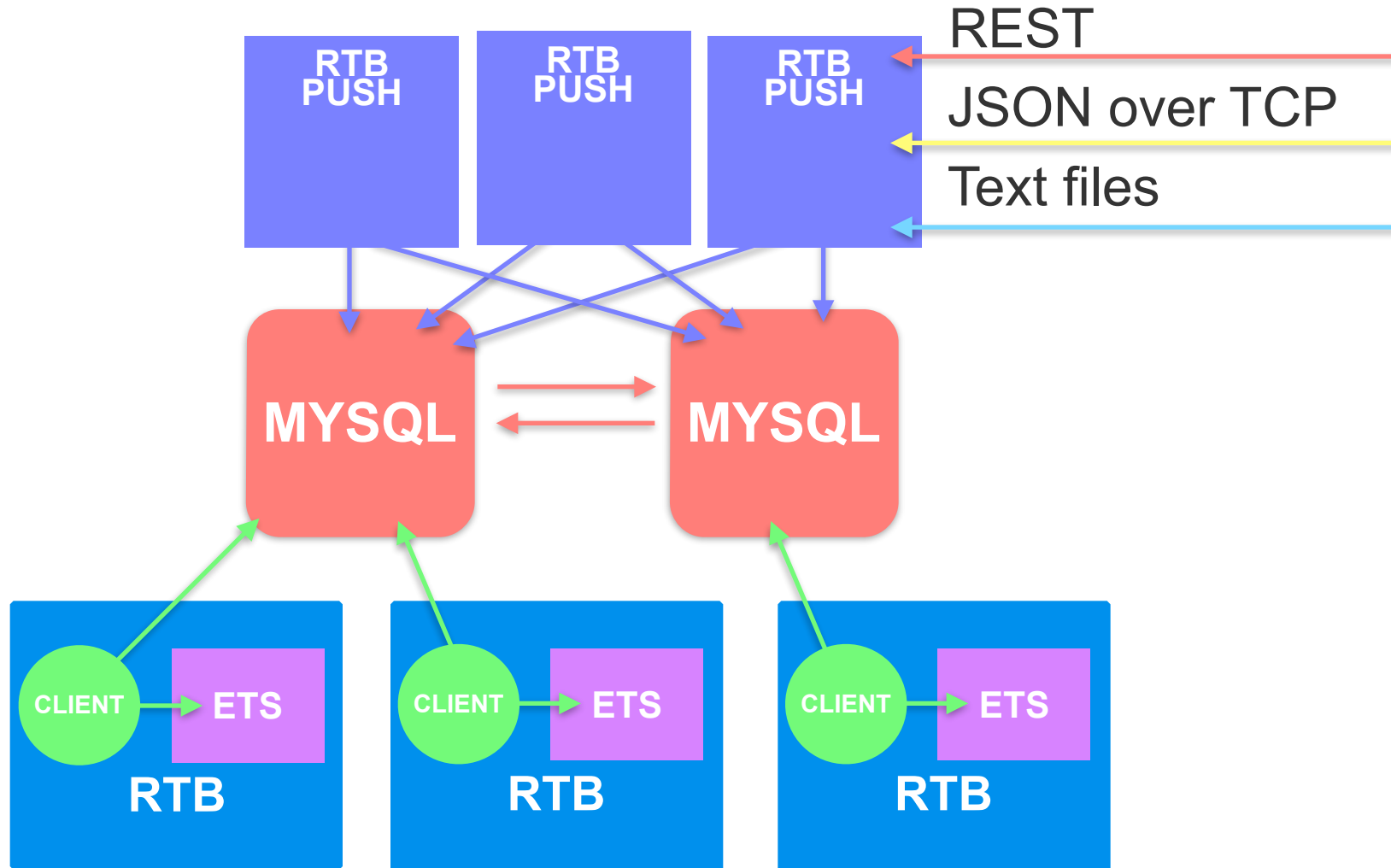
FIRST APPROACH - CONS

- Didn't scale well
- Server was a **single** process working **sequentially**
- **Failover** was causing clients to reconnect and refetch **all docs**
- Lack of **tools**



CONFIGURATION MGMT

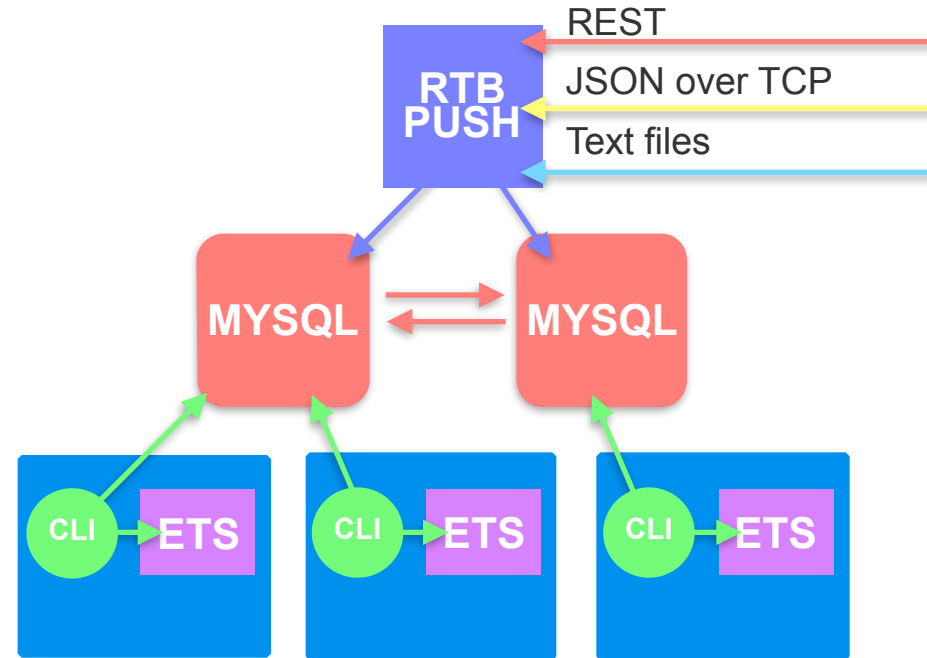
NEW APPROACH



CONFIGURATION MGMT

MyETS app

- **Fault tolerance**
- Alarms
- WebUI
- Load balances queries across different mysql nodes



MyETS processes

- Server storage
 - owns ETS tables
- Stats server
- Table handler sup
- Configuration driven

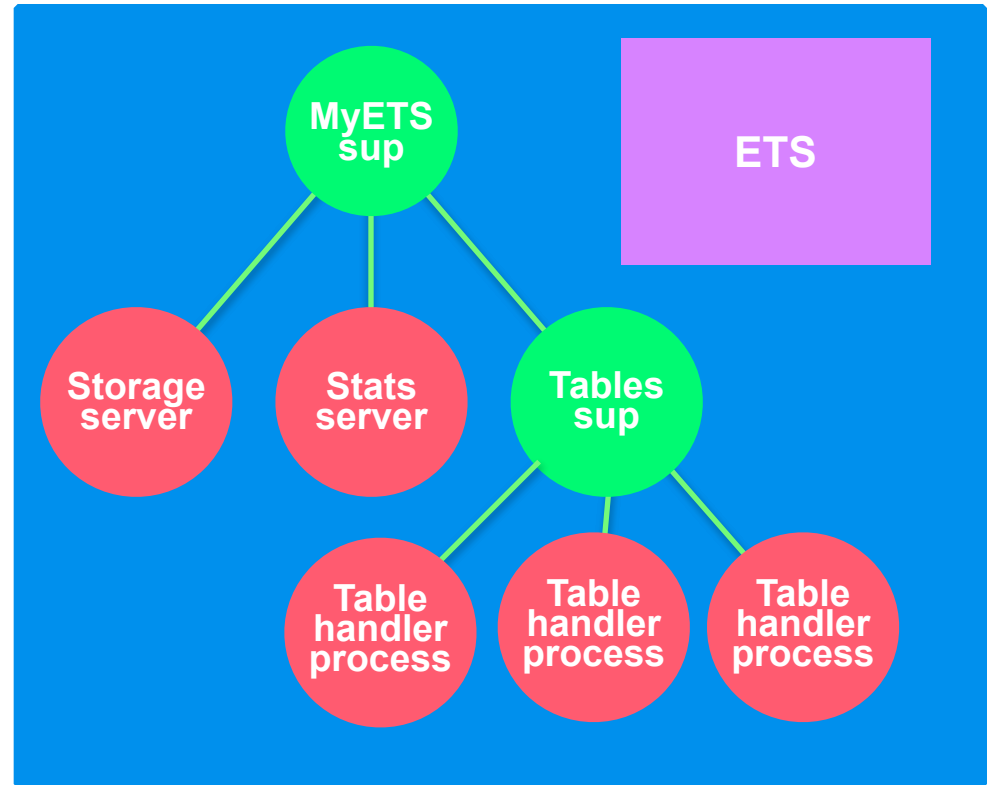


TABLE HANDLERS

- long running *gen_servers*
- isolated
- inits ETS table and conn pool
- polls db for data using functions from callback module

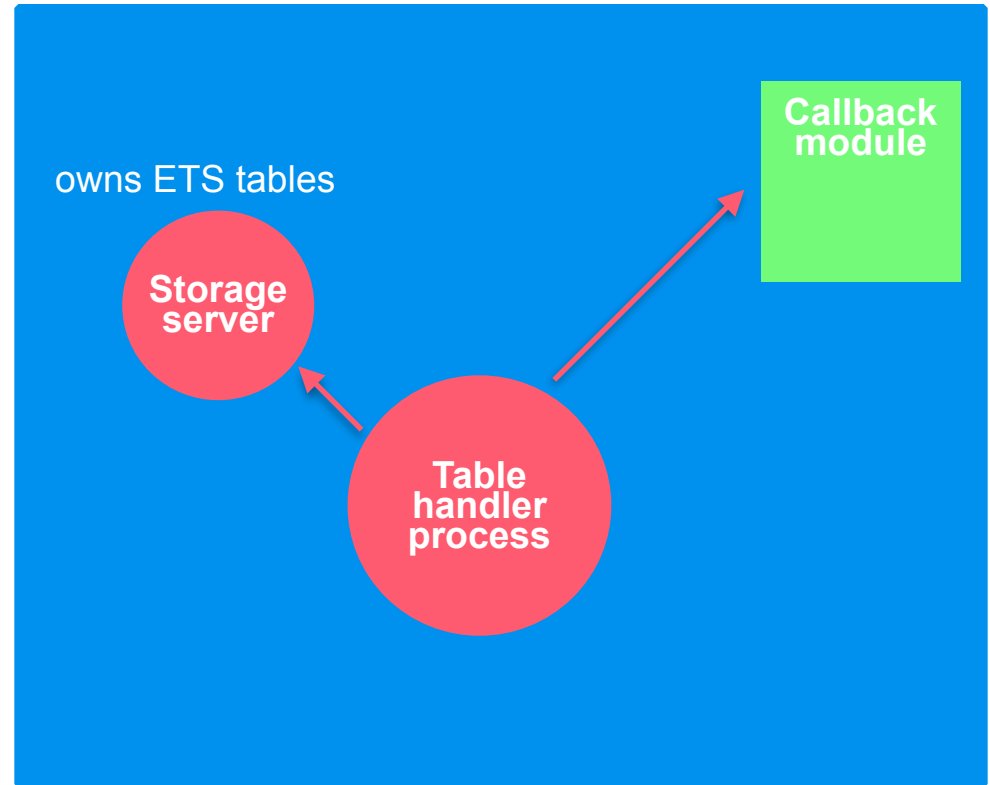
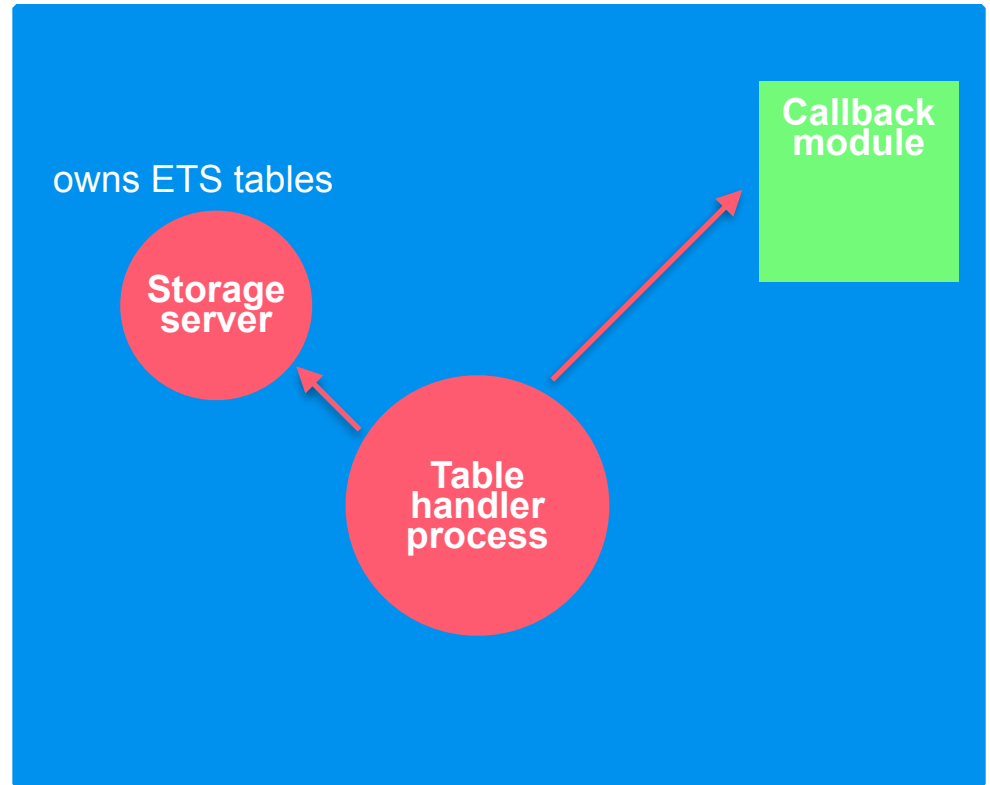


TABLE HANDLERS

- **init_storage()**
 - ran in storage server
- **reset()**
- **get_query(State)**
 - generates query
 - defines range
- **process_doc(State)**
 - processes results
- **get_next_state()**
 - deals with errors



EXAMPLE

`init_storage()` ->

```
ets:new(my_tab, [public, named_table]);  
#state{counter=0}.
```

`reset()` ->

```
Max = ets:foldl(fun get_max/2, -1, my_tab),  
#state{counter=Max}.
```

EXAMPLE

```
get_query(#state{counter=Cnt}) ->
```

```
“SELECT id, docid, doc FROM docs WHERE id >”
```

```
++ integer_to_list(Cnt) ++ “ORDER BY id”
```

```
“LIMIT 5”.
```

```
process_doc([Id, DocId, Body], State) ->
```

```
ets:insert(my_tab, #my_doc{id=DocId,  
sth=Body}),
```

```
State#state{counter=Id}.
```

EXAMPLE

```
{myets, [  
  {hosts, [{rtb_push_db,  
    [{user, "root"}, {password, "testtest"},  
    {host, "127.0.0.1"}, {port, 3306},  
    {database, "push"}]}]},  
  {db_config, [{rtb_push_db,  
    myets_table_handler_incremental,  
    [my_tab_cb]}]}].
```

BENEFITS ERROR HANDLING

- Don't need to handle **crashes** in your code
 - MyETS reconnects to MySQL
 - MyETS restarts sync
- Each table handler is **isolated** and has it's own **pool**
 - **errors** in one handler doesn't **affect others**
 - **long running** queries doesn't block others
- No need to worry about **ETS table ownership**

BENEFITS ERROR HANDLING

- Stats page shows sync status and **measures sync time**
- Stat server monitors syncing and raises alerts
- All timeouts are configurable

BENEFITS LOAD BALANCING

Resolves MySQL host name.

If there is more than **one IP address** tries to pick one ip using **simple hashing function**.

In case of connection failure tries to connect to next ip from the list

```
hash_name(Node, NumMySQLHosts) ->  
    Sum = lists:sum(Node),  
    Sum rem NumMySQLHosts.
```

QUESTIONS

Coming soon to <http://aol.github.io/>

@skamander