Code review guideline

Applicability	EBX Add-ons	QMS Level	3
Purpose	Code review guidelines	**Page Status**	PUBLISHED
Standards & Compliance	ISO 27001: A.14.2.5	**Publish Date**	09 Sep 2020
Owner/Author Email	Minh Tran Quang	**Version**	1.0
Classification	For internal use only - TIBCO Confidential		
Page Editing	This page limits editing; suggestions and changes are welcome and can be made via the page comments or contact the owner(s) of this page.		

Handle string

- "ABC".equals(abc) not abc.equals(“ABC”) avoid NPE
- Use StringBuilder not “+” improve performance and memory
    - Don’t use StringBuffer unless you know what are you doing!!!
- Use abc.isEmpty() not abc.equals(“”) gain 60% performance
- Use abc.equalsIgnoreCase(def) not abc.toLowerCase().equals(def.toLowerCase())
- Use abc.regionMatches(true,0,def,0,length) not abc.startsWith(def.toLowerCase());
- If a string (or an object) never be changed, convert it to a constant

For loop

- Avoid *forEach* if possible
- Iterate over *Map<>* example

```
Set<Entry<Path, TableDescriptor>> entrySet = this.outgoingSchema.entrySet();
Iterator<Entry<Path, TableDescriptor>> iterator = entrySet.iterator();
int size = entrySet.size();
for (int i = 0; i < size; i++)
{
   Entry<Path, TableDescriptor> item = iterator.next();
   if (foreignTable.equals(item.getValue()))
   {
      results.add(item.getKey());
   }
}
```

Condition control statement

Use *switch* if there are more than 3 values to be checked

Switch block always have default branch, it has to throw *IllegalArgumentException*

An *if* condition doesn't have more than 3 operators

Handle NULL

- Avoid return NULL when have to return a collection
  - Use *Collections.empty*()*
- Always check NULL before consume an object.

Handle Exception

If there are multiple exceptions to be catched, start with the most detail one

Don’t catch an exception if don’t have enough information to handle it correctly

Never catch Exception then write useless log

```
catch (Exception ex)
{
   if (TeseLogger.getLoggingCategory().isDebug())
   {                                                DON'T DO IT
      TeseLogger.getLoggingCategory().debug(ex.getMessage());
   }
}
```

Too complex lambda

Meaningful name of methods and variables

- Name of a method has to have a verb or verb phrase

- Name of a variable has to have a noun or noun phrase
- Use pronounceable names
- Use searchable names
- Length of name (just suggestion, don’t strictly follow)

- 1 char for loop counters
- 1 word for condition/loop variables
- 2-3 words for methods
- 2-4 words for classes
- 3-5 words for globals

Data structures

- *LinkedList* can be used in those scenarios:
  - Frequently insert/remove at head
  - Frequently access at head
  - Size of list can variable from a few to hundreds of thousand
- *ArrayList* can be used in those scenarios:
  - Need to random access item by index
  - Size of list is predictable
  - Frequently insert/remove at tail
- *HashSet* can be used in those scenarios:
  - Frequently check contains an item
  - Don't want to store duplicated items
- *HashMap* can be used in those scenarios:
  - Need to store an object related with a key

Don't Repeat Yourself

Duplicated code must extract as a method

Sign-off History

Action	Name	Date
Prepared by	Thi Viet Phuong Luu	21 Feb 2020
Approved	Minh Tran Quang	09 Sep 2020

Revision History

Version	Date	Authors	Description
1.0	09 Sep 2020	Minh Tran Quang	Initial version