package com.twitter.search.earlybird.util;

import com.google.common.annotations.VisibleForTesting;

import com.twitter.common.base.ExceptionalFunction;

public interface CoordinatedEarlybirdActionInterface {

/\*\*

\* Executes the provided Function associated with the given segment.

\* @param description a name for the action to be exected.

\* @param function the function to call in a coordinated manner.

\* As input, the function will receive a flag indicating whether or not it is being

\* called in a coordinated fashion. true if it is, and false otherwise.

\* @return true iff the function was executed, and function.apply() returned true;

\* throws CoordinatedEarlybirdActionLockFailed if function is not executed (because lock

\* aquisition failed).

\*/

<E extends Exception> boolean execute(

String description,

ExceptionalFunction<Boolean, Boolean, E> function)

throws E, CoordinatedEarlybirdActionLockFailed;

/\*\*

\* Set whether this action should be synchronized.

\* If not, the action is directly applied. If yes, Earlybirds will coordinate executing the

\* action via ZooKeeperTryLocks.

\*/

boolean setShouldSynchronize(boolean shouldSynchronizeParam);

/\*\*

\* Number of times this coordinated actions has been executed.

\* @return

\*/

@VisibleForTesting

long getNumCoordinatedFunctionCalls();

/\*\*

\* Number of times we have left the serverset.

\* @return

\*/

@VisibleForTesting

long getNumCoordinatedLeaveServersetCalls();

/\*\*

\* Retry until we can run an action on a single instance in the serverset.

\* @param description Text description of the action.

\* @param action A runnable to be ran.

\*/

void retryActionUntilRan(String description, Runnable action);

}