package com.twitter.search.earlybird\_root.quota;

import java.util.Optional;

import javax.inject.Inject;

import com.google.common.base.Preconditions;

import com.twitter.search.common.dark.ServerSetResolver.SelfServerSetResolver;

/\*\*

\* A config based implementation of the {@code ClientIdQuotaManager} interface.

\* It uses a ConfigBasedQuotaConfig object to load the contents of the config.

\*/

public class ConfigRepoBasedQuotaManager implements ClientIdQuotaManager {

public static final String COMMON\_POOL\_CLIENT\_ID = "common\_pool";

private final ConfigBasedQuotaConfig quotaConfig;

private final SelfServerSetResolver serverSetResolver;

/\*\* Creates a new ConfigRepoBasedQuotaManager instance. \*/

@Inject

public ConfigRepoBasedQuotaManager(

SelfServerSetResolver serverSetResolver,

ConfigBasedQuotaConfig quotaConfig) {

Preconditions.checkNotNull(quotaConfig);

this.quotaConfig = quotaConfig;

this.serverSetResolver = serverSetResolver;

}

@Override

public Optional<QuotaInfo> getQuotaForClient(String clientId) {

Optional<QuotaInfo> quotaForClient = quotaConfig.getQuotaForClient(clientId);

if (!quotaForClient.isPresent()) {

return Optional.empty();

}

QuotaInfo quota = quotaForClient.get();

int quotaValue = quota.getQuota();

int rootInstanceCount = serverSetResolver.getServerSetSize();

if (rootInstanceCount > 0) {

quotaValue = (int) Math.ceil((double) quotaValue / rootInstanceCount);

}

return Optional.of(

new QuotaInfo(

quota.getQuotaClientId(),

quota.getQuotaEmail(),

quotaValue,

quota.shouldEnforceQuota(),

quota.getClientTier(),

quota.hasArchiveAccess()));

}

@Override

public QuotaInfo getCommonPoolQuota() {

Optional<QuotaInfo> commonPoolQuota = getQuotaForClient(COMMON\_POOL\_CLIENT\_ID);

Preconditions.checkState(commonPoolQuota.isPresent());

return commonPoolQuota.get();

}

}