package com.twitter.tweetypie

package util

import com.twitter.tweetypie.thriftscala.\_

object ConversationControls {

object Create {

def byInvitation(

inviteViaMention: Option[Boolean] = None

): TweetCreateConversationControl.ByInvitation = TweetCreateConversationControl.ByInvitation(

TweetCreateConversationControlByInvitation(inviteViaMention = inviteViaMention)

)

def community(

inviteViaMention: Option[Boolean] = None

): TweetCreateConversationControl.Community = TweetCreateConversationControl.Community(

TweetCreateConversationControlCommunity(inviteViaMention = inviteViaMention)

)

def followers(

inviteViaMention: Option[Boolean] = None

): TweetCreateConversationControl.Followers = TweetCreateConversationControl.Followers(

TweetCreateConversationControlFollowers(inviteViaMention = inviteViaMention)

)

}

object Scenario {

case class CommonScenario(

createConversationControl: TweetCreateConversationControl,

descriptionSuffix: String,

expectedConversationControl: (UserId, Seq[UserId]) => ConversationControl,

inviteViaMention: Option[Boolean])

def mkCommunityScenario(inviteViaMention: Option[Boolean]): CommonScenario =

CommonScenario(

Create.community(inviteViaMention = inviteViaMention),

"community",

expectedConversationControl = (authorId, userIds) => {

community(userIds, authorId, inviteViaMention)

},

inviteViaMention

)

def mkByInvitationScenario(inviteViaMention: Option[Boolean]): CommonScenario =

CommonScenario(

Create.byInvitation(inviteViaMention = inviteViaMention),

"invited users",

expectedConversationControl = (authorId, userIds) => {

byInvitation(userIds, authorId, inviteViaMention)

},

inviteViaMention

)

def mkFollowersScenario(inviteViaMention: Option[Boolean]): CommonScenario =

CommonScenario(

Create.followers(inviteViaMention = inviteViaMention),

"followers",

expectedConversationControl = (authorId, userIds) => {

followers(userIds, authorId, inviteViaMention)

},

inviteViaMention

)

val communityScenario = mkCommunityScenario(None)

val communityInviteViaMentionScenario = mkCommunityScenario(Some(true))

val byInvitationScenario = mkByInvitationScenario(None)

val byInvitationInviteViaMentionScenario = mkByInvitationScenario(Some(true))

val followersScenario = mkFollowersScenario(None)

val followersInviteViaMentionScenario = mkFollowersScenario(Some(true))

}

def byInvitation(

invitedUserIds: Seq[UserId],

conversationTweetAuthorId: UserId,

inviteViaMention: Option[Boolean] = None

): ConversationControl =

ConversationControl.ByInvitation(

ConversationControlByInvitation(

conversationTweetAuthorId = conversationTweetAuthorId,

invitedUserIds = invitedUserIds,

inviteViaMention = inviteViaMention

)

)

def community(

invitedUserIds: Seq[UserId],

conversationTweetAuthorId: UserId,

inviteViaMention: Option[Boolean] = None

): ConversationControl =

ConversationControl.Community(

ConversationControlCommunity(

conversationTweetAuthorId = conversationTweetAuthorId,

invitedUserIds = invitedUserIds,

inviteViaMention = inviteViaMention

)

)

def followers(

invitedUserIds: Seq[UserId],

conversationTweetAuthorId: UserId,

inviteViaMention: Option[Boolean] = None

): ConversationControl =

ConversationControl.Followers(

ConversationControlFollowers(

conversationTweetAuthorId = conversationTweetAuthorId,

invitedUserIds = invitedUserIds,

inviteViaMention = inviteViaMention

)

)

}