Pavitha P

Candidate

E-mail:
pavithasakthi1@gmail.com

Session

ID: B9EZR3-8D3
Time limit: 60 min.
Report recipients: No one
Accessed from: 157.49.169.224, 157.49.169.224
Invited by: arun.k@uplers.in

Status: completed

Created on: 2022-12-16 10:36 UTC
Started on: 2022-12-16 10:37 UTC
Finished on: 2022-12-16 11:50 UTC

Notes:
N/A

Similarity Check

Status: not found
No similar solutions have been detected.

Test score

0%

Tasks in test	Score
1 AsyncJobsScheduler	N/A

Tasks Details

Medium

1. AsyncJobsScheduler

Handle timeouts and cancellations in coroutine calls.

Task Score	Correctness	Performance
N/A	Not assessed	Not assessed

Task description

You are required to implement a user data aggregator using coroutines by implementing an `aggregateDataForCurrentUser` function.

Requirements

Your task is to resolve and aggregate user details in an `aggregateDataForCurrentUser` function.

You have APIs for resolving user details (`UserEntity`), user comments (`List<CommentEntity>`) and friends suggestions (`List<FriendEntity>`).

To resolve current user details, call the `resolveCurrentUser` function. To resolve user comments, call `fetchUserComments` with `id` resolved from `UserEntity.id`. To resolve user friends suggestions, call `fetchSuggestedFriends` with `id` resolved from `UserEntity.id`.

All of these functions are suspend functions; they need to be invoked in some coroutine context.

Implement functions in `AggregateUserDataUseCase`, where `aggregateDataForCurrentUser` must satisfy the following requirements:

1. When `resolveCurrentUser` throws an error, forward the exception.
2. When `resolveCurrentUser` returns `UserEntity`, call `fetchUserComments` and `fetchSuggestedFriends` and then aggregate the results into `AggregatedData`.
3. When `fetchUserComments` call takes longer than 2000ms, return an empty comments list.
4. When `fetchUserComments` throws an exception, return an empty comments list.
5. When `fetchSuggestedFriends` call takes longer than 2000ms, return an empty friends suggestions list.
6. When `fetchSuggestedFriends` throws an exception, return an empty friends suggestions list.

And for `AggregateUserDataUseCase.close`:

- When `close` is invoked, cancel all ongoing coroutine calls.

Hints

- You may create additional `CoroutineScope` inside `AggregateUserDataUseCase` to handle cancellation when `AggregateUserDataUseCase.close` is invoked.

Classes accompanying `AggregateUserDataUseCase`

```
package coroutines

data class AggregatedData(
    val user: UserEntity,
    val comments: List<CommentEntity>,
    val suggestedFriends: List<FriendEntity>
)

typealias UserId = String

data class UserEntity(val id: UserId, val name: String)

data class CommentEntity(val id: String, val content: String)

data class FriendEntity(val id: String, val name: String)
```

Available packages/libraries

- 'org.jetbrains.kotlin:kotlin-stdlib:1.4.20'
- 'org.jetbrains.kotlinx:kotlinx-coroutines-jdk8:1.4.2'

Solution

See Live Version

Programming language used: (None)

No final submissions.