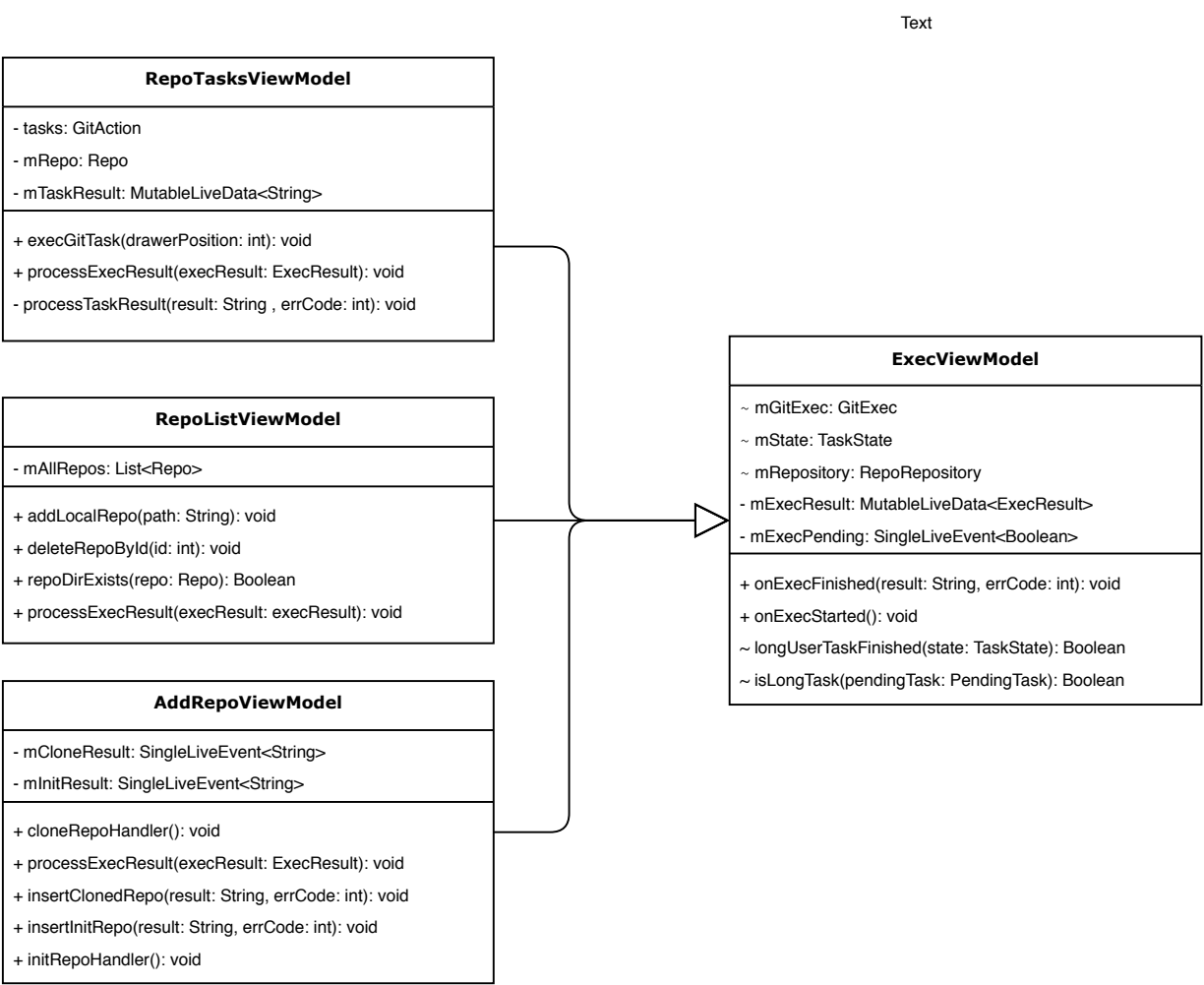
Text
RepoTasksViewModel
- tasks: GitAction
- mRepo: Repo
- mTaskResult: MutableLiveData<String>
+ execGitTask(drawerPosition: int): void
+ processExecResult(execResult: ExecResult): void
- processTaskResult(result: String , errCode: int): void
RepoListViewModel
- mAllRepos: List<Repo>
+ addLocalRepo(path: String): void
+ deleteRepoById(id: int): void
+ repoDirExists(repo: Repo): Boolean
+ processExecResult(execResult: execResult): void
AddRepoViewModel
- mCloneResult: SingleLiveEvent<String>
- mInitResult: SingleLiveEvent<String>
+ cloneRepoHandler(): void
+ processExecResult(execResult: ExecResult): void
+ insertClonedRepo(result: String, errCode: int): void
+ insertInitRepo(result: String, errCode: int): void
+ initRepoHandler(): void
ExecViewModel
~ mGitExec: GitExec
~ mState: TaskState
~ mRepository: RepoRepository
- mExecResult: MutableLiveData<ExecResult>
- mExecPending: SingleLiveEvent<Boolean>
+ onExecFinished(result: String, errCode: int): void
+ onExecStarted(): void
~ longUserTaskFinished(state: TaskState): Boolean
~ isLongTask(pendingTask: PendingTask): Boolean