

LiveDataTestUtil
(com.cleanup.todos.utils)
+getValueLiveData : LiveData<T> : T

MainActivityInstrumentedTest
(com.cleanup.todos)
-mDatabase : TodocDatabase
+rule : ActivityTestRule<MainActivity> = new ActivityTestRule<>(MainActivity.class)
+initDb() : void
+closeDb() : void
+addAndRemoveTask() : void
+sortTasks() : void

ProjectDaoInstrumentedTest
(com.cleanup.todos)
-mDatabase : TodocDatabase
-PROJECT_ID : long = 1
-SECOND_PROJECT_ID : long = 2
-PROJECT_DEMO : Project = new Project(PROJECT_ID, "TestProject", 0xFFEADAD1)
-SECOND_PROJECT_DEMO : Project = new Project(SECOND_PROJECT_ID, "SecondTestProject", 0xFFEADAD1)
+mInstantTaskExecutorRule : InstantTaskExecutorRule = new InstantTaskExecutorRule()
+initDb() : void
+closeDb() : void
+getAllProjects() : void
+getProject() : void
+createProject() : void
+deleteProject() : void

TodocDatabase
(com.cleanup.todos.database)
-INSTANCE : TodocDatabase
-mDatabase : TodocDatabase
+mTaskDao() : TaskDao
+mProjectDao() : ProjectDao
+getInstance(context : Context) : TodocDatabase
+populateDatabase() : Callback
+prepopulateDatabase() : Callback

TaskDaoInstrumentedTest
(com.cleanup.todos)
-mDatabase : TodocDatabase
-PROJECT_ID : long = 1
-PROJECT_DEMO : Project = new Project(PROJECT_ID, "TestProject", 0xFFEADAD1)
-TASK_DEMO : Task = new Task(PROJECT_ID, "TestTask", 8)
-SECOND_TASK_DEMO : Task = new Task(PROJECT_ID, "SecondTestTask", 9)
+mInstantTaskExecutorRule : InstantTaskExecutorRule = new InstantTaskExecutorRule()
+initDb() : void
+closeDb() : void
+getAllTasks() : void
+getTask() : void
+insertAndUpdateTask() : void
+insertAndDeleteTask() : void

Project
(com.cleanup.todos.model)
-id : long
-name : String
-color : int
+Project(id : long, name : String, color : int)
+getProjects() : List<Project>
+getProjectById(id : long) : Project
+toString() : String

ProjectRepositoryUnitTest
(com.cleanup.todos)
-mProjectDao : ProjectDao = mock(ProjectDao.class)
-mProjectDataRepository : ProjectDataRepository
-PROJECT_DEMO : Project = new Project(1, "TestProject", 0xFFEADAD1)
+setUp() : void
+getProjects() : void
+getProject() : void
+createProject() : void
+deleteProject() : void

SortMethod
(com.cleanup.todos.ui.MainActivity)
+ALPHABETICAL
+ALPHABETICAL_INVERTED
+RECENT_FIRST
+OLD_FIRST
+NONE

ProjectDao
(com.cleanup.todos.database.dao)
+getProjects() : LiveData<List<Project>>
+getProject(projectId : long) : LiveData<Project>
+createProject(project : Project) : void
+deleteProject(projectId : long) : void

ProjectDataRepository
(com.cleanup.todos.repositories)
-mProjectDao : ProjectDao
+getProjects() : LiveData<List<Project>>
+getProject(projectId : long) : LiveData<Project>
+createProject(project : Project) : void
+deleteProject(projectId : long) : void

ViewModelFactory
(com.cleanup.todos.injections)
-mTaskDataRepository : TaskDataRepository
-mProjectDataRepository : ProjectDataRepository
-mExecutor : Executor
+Factory : ViewModelFactory
+getInstance(context : Context) : ViewModelFactory
+ViewModelFactory(context : Context)
+create(modelClass : Class<T>) : T

TaskDao
(com.cleanup.todos.database.dao)
+getTasks() : LiveData<List<Task>>
+getTask(taskId : long) : LiveData<Task>
+createTask(task : Task) : long
+updateTask(task : Task) : void
+deleteTask(taskId : long) : void

TaskDataRepository
(com.cleanup.todos.repositories)
-mTaskDao : TaskDao
+TaskDataRepository(taskDao : TaskDao)
+getTasks() : LiveData<List<Task>>
+getTask(taskId : long) : LiveData<Task>
+createTask(task : Task) : void
+updateTask(task : Task) : void
+deleteTask(taskId : long) : void

TaskViewModel
(com.cleanup.todos.view)
-mTaskDataRepository : TaskDataRepository
-mProjectDataRepository : ProjectDataRepository
-mExecutor : Executor
+TaskViewModel(taskDataRepository : TaskDataRepository, projectDataRepository : ProjectDataRepository, executor : Executor)
+getProjects() : LiveData<List<Project>>
+getProject(projectId : long) : LiveData<Project>
+createProject(project : Project) : void
+deleteProject(projectId : long) : void
+getTasks() : LiveData<List<Task>>
+getTask(taskId : long) : LiveData<Task>
+createTask(task : Task) : void
+updateTask(task : Task) : void
+deleteTask(taskId : long) : void

TaskRepositoryUnitTest
(com.cleanup.todos)
-mTaskDao : TaskDao = mock(TaskDao.class)
-mTaskDataRepository : TaskDataRepository
-TASK_DEMO : Task = new Task(1, "TestTask", 8)
+setUp() : void
+getTasks() : void
+getTask() : void
+createTask() : void
+updateTask() : void
+deleteTask() : void

TaskViewModelUnitTest
(com.cleanup.todos)
-mProjectDataRepository : ProjectDataRepository = mock(ProjectDataRepository.class)
-mTaskDataRepository : TaskDataRepository = mock(TaskDataRepository.class)
-mExecutor : Executor = spy(Runnable -> Runnable.run())
-mTaskViewModel : TaskViewModel
-PROJECT_DEMO : Project = new Project(1, "TestProject", 0xFFEADAD1)
-TASK_DEMO : Task = new Task(1, "TestTask", 8)
+setUp() : void
+getProjects() : void
+getProject() : void
+createProject() : void
+deleteProject() : void
+getTasks() : void
+getTask() : void
+createTask() : void
+updateTask() : void
+deleteTask() : void

MainActivity
(com.cleanup.todos.ui)
-mTaskViewModel : TaskViewModel
-mProjects : Project
-mTasks : Task
-sortMethod : SortMethod = SortMethod.NONE
+dialog : AlertDialog = null
+dialogEditText : EditText = null
+dialogSpinner : Spinner = null
+listTasks : RecyclerView
+bNoTasks : TextView
+onCreate(savedInstanceState : Bundle) : void
+configureViewModel() : void
+initUI() : void
+initProjects() : void
+updateProjects/projects : List<Project> : void
+initTasks() : void
+updateTasks/tasks : List<Task> : void
+onPostContributorClick(dialogInterface : DialogInterface) : void
+showAddTaskDialog() : void
+getAddTaskDialog() : AlertDialog
+populateDialogSpinner() : void
+onDeleteTask(task : Task) : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : boolean

DeleteTaskListener
(com.cleanup.todos.ui.TasksAdapter)
+onDeleteTask(task : Task) : void
+deleteTaskListener

TaskViewHolder
(com.cleanup.todos.ui.TasksAdapter)
+imgProject : AppCompatImageView
+bTaskName : TextView
+bProjectName : TextView
+imgDelete : AppCompatImageView
+deleteTaskListener : DeleteTaskListener
+TaskViewHolder(itemView : View, deleteTaskListener : DeleteTaskListener)
+bind(task : Task) : void
+TaskViewHolder(itemView : View, deleteTaskListener : DeleteTaskListener)
+TaskViewHolder(itemView : View, deleteTaskListener : DeleteTaskListener)
+TaskViewHolder(itemView : View, deleteTaskListener : DeleteTaskListener)

TasksAdapter
(com.cleanup.todos.ui)
-mTasks : Task
-mDeleteTaskListener : DeleteTaskListener
+TasksAdapter(tasks : List<Task>, deleteTaskListener : DeleteTaskListener)
+onCreateViewHolder(viewGroup : ViewGroup, viewType : int) : TaskViewHolder
+onBindViewHolder(taskViewHolder : TaskViewHolder, position : int) : void
+getItemCount() : int

ActionOnItemViewAtPositionViewAction
(com.cleanup.todos.utils.TestUtils)
-position : int
-viewAction : ViewAction
-viewId : int
+ActionOnItemViewAtPositionViewAction(position : int, viewId : int, viewAction : ViewAction)
+getConstraints() : Matcher<View>
+getDescription() : String
+perform(uController : UIController, view : View) : void

TestUtils
(com.cleanup.todos.utils)
+ActionOnItemViewAtPosition(position : int, viewId : int, viewAction : ViewAction) : ViewAction
+withRecyclerView(recyclerViewId : int) : RecyclerViewMatcher

ScrollToPositionViewAction
(com.cleanup.todos.utils.TestUtils)
-position : int
+ScrollToPositionViewAction(position : int)
+getConstraints() : Matcher<View>
+getDescription() : String
+perform(uController : UIController, view : View) : void

RecyclerViewMatcher
(com.cleanup.todos.utils)
-recyclerView : int
+RecyclerViewMatcher(recyclerView : int)
+withPosition(position : int) : Matcher<View>
+withPositionOnView(position : int, targetViewId : int) : Matcher<View>

TaskUnitTest
(com.cleanup.todos)
+Test_projects() : void