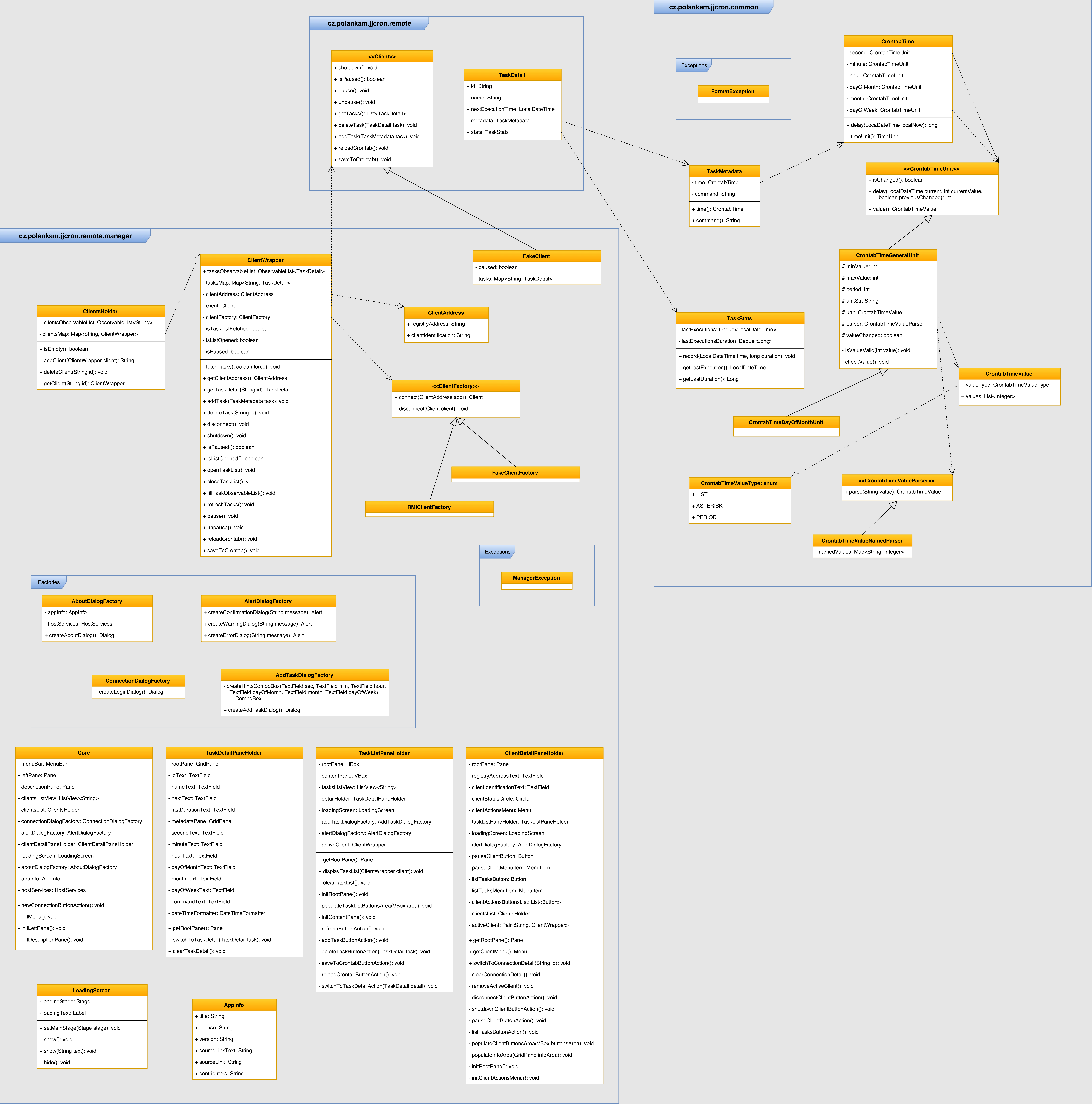
cz.polankam.jjcron.remote
<<Client>>
+ shutdown(): void
+ isPaused(): boolean
+ pause(): void
+ unpause(): void
+ getTasks(): List<TaskDetail>
+ deleteTask(TaskDetail task): void
+ addTask(TaskMetadata task): void
+ reloadCrontab(): void
+ saveToCrontab(): void
TaskDetail
+ id: String
+ name: String
+ nextExecutionTime: LocalDateTime
+ metadata: TaskMetadata
+ stats: TaskStats
cz.polankam.jjcron.common
Exceptions
FormatException
CrontabTime
- second: CrontabTimeUnit
- minute: CrontabTimeUnit
- hour: CrontabTimeUnit
- dayOfMonth: CrontabTimeUnit
- month: CrontabTimeUnit
- dayOfWeek: CrontabTimeUnit
+ delay(LocaDateTime localNow): long
+ timeUnit(): TimeUnit
TaskMetadata
- time: CrontabTime
- command: String
+ time(): CrontabTime
+ command(): String
<<CrontabTimeUnit>>
+ isChanged(): boolean
+ delay(LocalDateTime current, int currentValue, boolean previousChanged): int
+ value(): CrontabTimeValue
CrontabTimeGeneralUnit
# minValue: int
# maxValue: int
# period: int
# unitStr: String
# unit: CrontabTimeValue
# parser: CrontabTimeValueParser
# valueChanged: boolean
- isValueValid(int value): void
- checkValue(): void
TaskStats
- lastExecutions: Deque<LocalDateTime>
- lastExecutionsDuration: Deque<Long>
+ record(LocalDateTime time, long duration): void
+ getLastExecution(): LocalDateTime
+ getLastDuration(): Long
CrontabTimeValue
+ valueType: CrontabTimeValueType
+ values: List<Integer>
CrontabTimeDayOfMonthUnit
CrontabTimeValueType: enum
+ LIST
+ ASTERISK
+ PERIOD
<<CrontabTimeValueParser>>
+ parse(String value): CrontabTimeValue
CrontabTimeValueNamedParser
- namedValues: Map<String, Integer>
cz.polankam.jjcron.remote.manager
ClientWrapper
+ tasksObservableList: ObservableList<TaskDetail>
- tasksMap: Map<String, TaskDetail>
- clientAddress: ClientAddress
- client: Client
- clientFactory: ClientFactory
- isTaskListFetched: boolean
- isListOpened: boolean
- isPaused: boolean
- fetchTasks(boolean force): void
+ getClientAddress(): ClientAddress
+ getTaskDetail(String id): TaskDetail
+ addTask(TaskMetadata task): void
+ deleteTask(String id): void
+ disconnect(): void
+ shutdown(): void
+ isPaused(): boolean
+ isListOpened(): boolean
+ openTaskList(): void
+ closeTaskList(): void
+ fillTaskObservableList(): void
+ refreshTasks(): void
+ pause(): void
+ unpause(): void
+ reloadCrontab(): void
+ saveToCrontab(): void
FakeClient
- paused: boolean
- tasks: Map<String, TaskDetail>
ClientsHolder
+ clientsObservableList: ObservableList<String>
- clientsMap: Map<String, ClientWrapper>
+ isEmpty(): boolean
+ addClient(ClientWrapper client): String
+ deleteClient(String id): void
+ getClient(String id): ClientWrapper
ClientAddress
+ registryAddress: String
+ clientIdentification: String
<<ClientFactory>>
+ connect(ClientAddress addr): Client
+ disconnect(Client client): void
FakeClientFactory
RMIClientFactory
Exceptions
ManagerException
Factories
AboutDialogFactory
- appInfo: AppInfo
- hostServices: HostServices
+ createAboutDialog(): Dialog
AlertDialogFactory
+ createConfirmationDialog(String message): Alert
+ createWarningDialog(String message): Alert
+ createErrorDialog(String message): Alert
ConnectionDialogFactory
+ createLoginDialog(): Dialog
AddTaskDialogFactory
- createHintsComboBox(TextField sec, TextField min, TextField hour, TextField dayOfMonth, TextField month, TextField dayOfWeek): ComboBox
+ createAddTaskDialog(): Dialog
Core
- menuBar: MenuBar
- leftPane: Pane
- descriptionPane: Pane
- clientsListView: ListView<String>
- clientsList: ClientsHolder
- connectionDialogFactory: ConnectionDialogFactory
- alertDialogFactory: AlertDialogFactory
- clientDetailPaneHolder: ClientDetailPaneHolder
- loadingScreen: LoadingScreen
- aboutDialogFactory: AboutDialogFactory
- appInfo: AppInfo
- hostServices: HostServices
- newConnectionButtonAction(): void
- initMenu(): void
- initLeftPane(): void
- initDescriptionPane(): void
TaskDetailPaneHolder
- rootPane: GridPane
- idText: TextField
- nameText: TextField
- nextText: TextField
- lastDurationText: TextField
- metadataPane: GridPane
- secondText: TextField
- minuteText: TextField
- hourText: TextField
- dayOfMonthText: TextField
- monthText: TextField
- dayOfWeekText: TextField
- commandText: TextField
- dateTimeFormatter: DateTimeFormatter
+ getRootPane(): Pane
+ switchToTaskDetail(TaskDetail task): void
+ clearTaskDetail(): void
TaskListPaneHolder
- rootPane: HBox
- contentPane: VBox
- tasksListView: ListView<String>
- detailHolder: TaskDetailPaneHolder
- loadingScreen: LoadingScreen
- addTaskDialogFactory: AddTaskDialogFactory
- alertDialogFactory: AlertDialogFactory
- activeClient: ClientWrapper
+ getRootPane(): Pane
+ displayTaskList(ClientWrapper client): void
+ clearTaskList(): void
- initRootPane(): void
- populateTaskListButtonsArea(VBox area): void
- initContentPane(): void
- refreshButtonAction(): void
- addTaskButtonAction(): void
- deleteTaskButtonAction(TaskDetail task): void
- saveToCrontabButtonAction(): void
- reloadCrontabButtonAction(): void
- switchToTaskDetailAction(TaskDetail detail): void
ClientDetailPaneHolder
- rootPane: Pane
- registryAddressText: TextField
- clientIdentificationText: TextField
- clientStatusCircle: Circle
- clientActionsMenu: Menu
- taskListPaneHolder: TaskListPaneHolder
- loadingScreen: LoadingScreen
- alertDialogFactory: AlertDialogFactory
- pauseClientButton: Button
- pauseClientMenuItem: MenuItem
- listTasksButton: Button
- listTasksMenuItem: MenuItem
- clientActionsButtonsList: List<Button>
- clientsList: ClientsHolder
- activeClient: Pair<String, ClientWrapper>
+ getRootPane(): Pane
+ getClientMenu(): Menu
+ switchToConnectionDetail(String id): void
- clearConnectionDetail(): void
- removeActiveClient(): void
- disconnectClientButtonAction(): void
- shutdownClientButtonAction(): void
- pauseClientButtonAction(): void
- listTasksButtonAction(): void
- populateClientButtonsArea(VBox buttonsArea): void
- populateInfoArea(GridPane infoArea): void
- initRootPane(): void
- initClientActionsMenu(): void
LoadingScreen
- loadingStage: Stage
- loadingText: Label
+ setMainStage(Stage stage): void
+ show(): void
+ show(String text): void
+ hide(): void
AppInfo
+ title: String
+ license: String
+ version: String
+ sourceLinkText: String
+ sourceLink: String
+ contributors: String