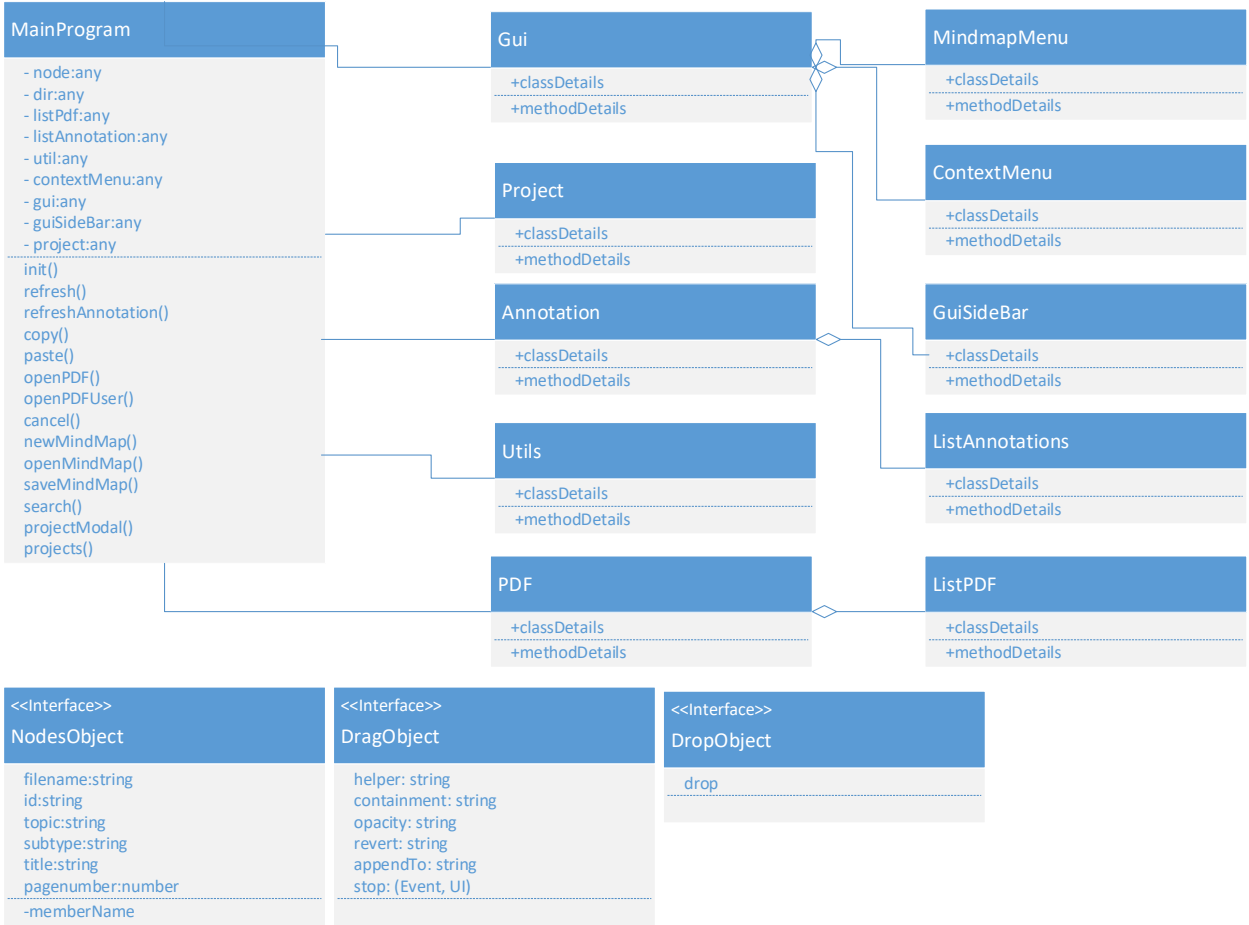


High Level Class Diagram



PDF Class Detail

PDF

- pdfName:string
- pdfLocation:string
- pdfLastModifiedDate:string

+ setPdfName(param:string)
+ getPdfName()
+ setPdfLocation(param:string)
+ getPdfLocation()
+ setPdfLastModifiedDate(param:string)
+ getPdfLastModifiedDate()

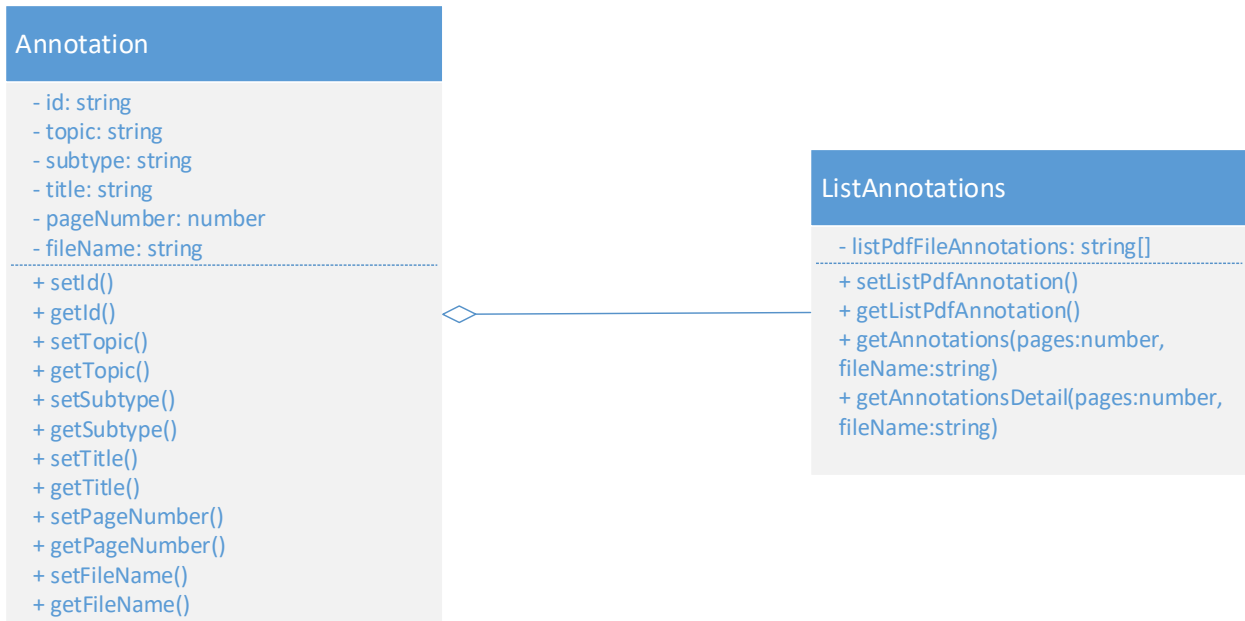


ListPdf

- listPdfFiles: string[]
- directory: string
- lastModDatePdfFiles: string[]

+ setListPdfFile(param:string[])
+ getListPdfFile(index:number)->string
+ getListPdf()->string[]
+ setLastModDate(list:string[])
+ getLastMod()-> string[]
+ getLastModDate(index:number)
+ getModDateFs(util:any, location:string, listFile:string[])
+ getListCount()->number
+ getPdfFromFs()->any
- readPdfInDirectory(param:string)->any
+ getPdfPage(path:string, filename:string)->any
- getPdfFilesPage(filePath:string, fileName:string)
+ setDirectory(dir:string)
+ getDirectory()->string
+ chkDateChange(util:any, list:string[]) -> any
+ clrListPdfState()

Annotation Class Detail



GUI Class Detail

Gui

```
- jsMindProjectNameState: string[]
- jsMindProjectsSavedState: string[]
- jsMindSavedState: string[]
- jsMindStatusState: string[]
- jsMindCurrentlyOpenedProject:
  number
- savedStateChildElement: any
+ setJsMindProjectNameState(input:string, idx:
  number)
+ getJsMindProjectNameState(name:string) ->
  number
+ getJsMindProjectName(idx:number) -> string
+ getCountProjectStatesNumber() -> number
+ setJsMindProjectState(input:string, idx:number)
+ getJsMindProjectState(idx:number) -> string
+ setJsMindStatusState(input:string, idx:number)
+ getJsMindStatusState(idx:number) -> string
+ setJsMindOpenProject(input:number)
+ getJsMindOpenProject() -> number
+ setChildElement(input:any)
+ getChildElement() -> any
+ setJsMindSavedState(input:string, idx:number)
+ getJsMindSavedState(idx:number) -> string
+ setReCreateState(list:any)
+ setGuiInitialize() -> any
+ windowConfirmation(msg:string) -> boolean
+ windowAlert(msg:string)
+ windowResizeListener()
+ windowNewProject()
+ windowOpenProject()
+ loadPdfButton(contents:any)
+ setPdfButton(nodeid:string, pdfid:string,
  pagenumber:string, node:any)
+ setJsMindListener()
+ setContextMenu()
+ setEffectShaking()
+ setElementVisibility()
- setShowHide(elementName:string, mode:string)
+ loadRecentProjects()
```

MindmapMenu

```
- jsMindObject: string
- fileTypeSave: string
+ setFileTypeSave(input:string)
+ getFileTypeSave() -> string
+ newMap(project: any, rootName: string)
+ loadFile(fileType:string) -> string
+ saveFile(fileType:string, mind_str: string) ->
  string
+ getJsMindData() -> any
+ selectFile()
+ setOpenFileListener()
+ loadFileJsMind(content:any, type:string,
  freemind_name: any) -> any
```

ContextMenu

```
- tempBoard: string
- clipBoard: string
+ setTempBoard(input:string)
+ setClipboard()
+ getClipboard() -> string
+ actionCopy()
+ actionPaste(project:any)
+ actionOpenPdf(directory:string)
+ actionOpenPdfUserDefine(dir: string)
+ actionCancel()
+ actionDelete
```

GuiSideBar

```
- basicHtmlTitle: any
- basicHtmlContent: any
+ setGuiInit(data:string, directory: string)
+ seGuiOnAppend(object:any, directory:string)
+ resetSideBarAnnotation()
+ checkSideBar() -> boolean
+ searchAnnotation(directory: string)
+ setTreeView(projectName: string, projectLoc:
  string, listFiles: string[])
+ openTreeView(input:string)
+ setTreeViewListener()
+ resetTreeView()
+ setReCreateTreeView() -> any
- setDynamicHtmlTitle(util, id: string, fileName:
  string) -> string
- setDynamicHtml(objekt: NodesObject, mode:
  string, directory: string)
- setDynamicHtmlContent(node, appendToName:
  string, className: string, mode: string, directory:
  string)
+ setAnnotationTitleVisibility()
- checkJsMindNode(id:string) -> boolean
+ checkAnnotationExistInSidebar(id: string)->
  boolean
+ findGuiAnnotationByAttribute(name: string, attr:
  any, val: string)-> any
```

Project Class Detail

Project

```
- projectName:string
- projectLocation:string
- projectPdfList:string[]
- projectStatus:string
- projectSavedFileLocation:string
- projectSavedPDFLocation: string
- projectTreeView: string
- projectPromises: any[]
+ setProjectName(input:string)
+ getProjectName()->string
+ setProjectLocation(input:string)
+ getProjectLocation()->string
+ setProjectPdfList(input:string[])
+ getProjectPdfList()->string[]
+ setProjectStatus(input:string)
+ getProjectStatus()->string
+ setProjectSavedFileLocation(input:string)
+ getProjectSavedFileLocation->string
+ setProjectTreeView(input:string)
+ getProjectTreeView() -> string
+ setProjectPromises(input: any[])
+ getProjectPromises() -> any[]
+ checkNewProject()->boolean
+ setNewProjectModal()
- setNewProjectListener()
+ createNewProject()
+ setSaveProject()->string
- saveProject(util:any, projectName:string,
projectLoc:string, listFile:string[],
projectSavedFile:string)
- setArrayAnnotation(childnode:any, nodes:any, i:
any) -> any
+ setTempSaveProject(param:string)
+ addToRecentProjects(projectName: string,
projectFilePath: string)
+ openRecentProject(projectFilePath: string)
+ setOpenProjectModal()
- setOpenProjectListener(self: any)
+ openProject(data: string, self: any, mode: string,
fileContent: string)
+ openProjectFromTreeView(input: string)
+ setCloseProject(status:string)
```

Utils Class Detail

Utils

- jsonFile: string
- jsonObject:NodesObject

- + setJsonFile(input:any[])->string
- + setJsonString(input:string)->string
- + parseJsonFile(input:string)->NodesObject
- + concatJsonFile(input:string)->string
- + getJsonFile()->string
- + getJsonObject()->NodesObject
- + getHashFunction(input:string)-> string
- + getLastMod(url:string)->any
- + getLastModFs(location:string, url:string)-> any
- + getComparedLastMod(param1:string, param2:string)-> boolean
- + writeAnyTypeFile(path:string, input:string, name:string, type:string)
- + readAnyTypeFile(path:string, readingtype:string, type:string)->any