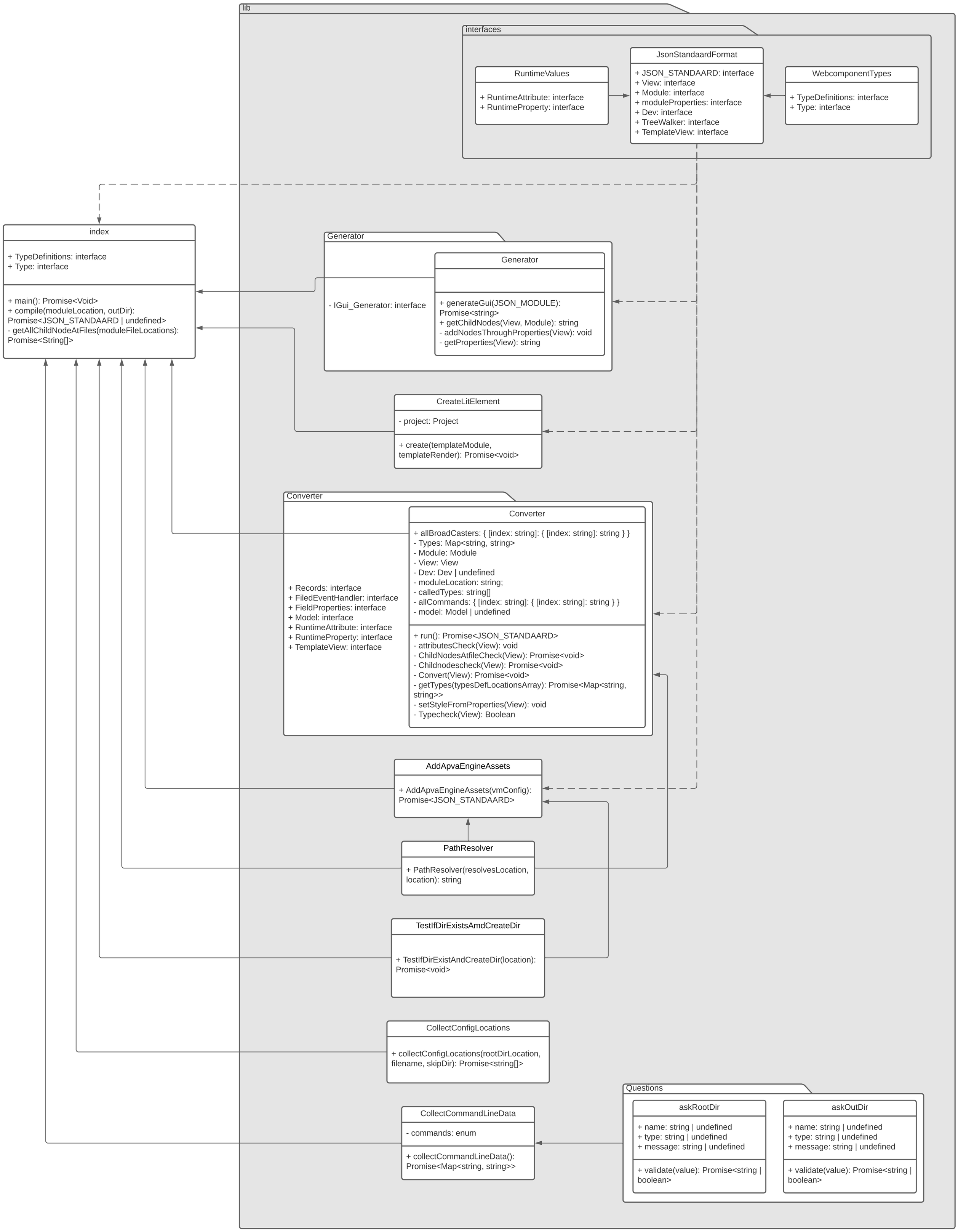
lib
interfaces
RuntimeValues
+ RuntimeAttribute: interface
+ RuntimeProperty: interface
JsonStandaardFormat
+ JSON_STANDAARD: interface
+ View: interface
+ Module: interface
+ moduleProperties: interface
+ Dev: interface
+ TreeWalker: interface
+ TemplateView: interface
WebcomponentTypes
+ TypeDefinitions: interface
+ Type: interface
index
+ TypeDefinitions: interface
+ Type: interface
+ main(): Promise<Void>
+ compile(moduleLocation, outDir): Promise<JSON_STANDAARD | undefined>
- getAllChildNodeAtFiles(moduleFileLocations): Promise<String[]>
Generator
- IGui_Generator: interface
Generator
+ generateGui(JSON_MODULE): Promise<string>
+ getChildNodes(View, Module): string
- addNodesThroughProperties(View): void
- getProperties(View): string
CreateLitElement
- project: Project
+ create(templateModule, templateRender): Promise<void>
Converter
+ Records: interface
+ FiledEventHandler: interface
+ FieldProperties: interface
+ Model: interface
+ RuntimeAttribute: interface
+ RuntimeProperty: interface
+ TemplateView: interface
Converter
+ allBroadCasters: { [index: string]: { [index: string]: string } }
- Types: Map<string, string>
- Module: Module
- View: View
- Dev: Dev | undefined
- moduleLocation: string;
- calledTypes: string[]
- allCommands: { [index: string]: { [index: string]: string } }
- model: Model | undefined
+ run(): Promise<JSON_STANDAARD>
- attributesCheck(View): void
- ChildNodesAtfileCheck(View): Promise<void>
- Childnodescheck(View): Promise<void>
- Convert(View): Promise<void>
- getTypes(typesDefLocationsArray): Promise<Map<string, string>>
- setStyleFromProperties(View): void
- Typecheck(View): Boolean
AddApvaEngineAssets
+ AddApvaEngineAssets(vmConfig): Promise<JSON_STANDAARD>
PathResolver
+ PathResolver(resolvesLocation, location): string
TestIfDirExistsAmdCreateDir
+ TestIfDirExistAndCreateDir(location): Promise<void>
CollectConfigLocations
+ collectConfigLocations(rootDirLocation, filename, skipDir): Promise<string[]>
CollectCommandLineData
- commands: enum
+ collectCommandLineData(): Promise<Map<string, string>>
Questions
askRootDir
+ name: string | undefined
+ type: string | undefined
+ message: string | undefined
+ validate(value): Promise<string | boolean>
askOutDir
+ name: string | undefined
+ type: string | undefined
+ message: string | undefined
+ validate(value): Promise<string | boolean>