



index

- globalConfig: GlobalConfig

+ main(): Promise<void>
- getConfig(pathToOwnConfig): Promise<Config>
- compile(moduleFiles, outDir): Promise<void>
- prepareGenerate(moduleFiles, outDir): Promise<void>
- generate(moduleConfig, outDir, relDirs): Promise<void>

interfaces

RuntimeValues

- + RuntimeAttribute: interface
- + RuntimeProperty: interface

Config

- + Config: interface
- + ConfigFile: interface
- + ConfigModules: interface
- + InitialLumberRootConfig: interface
- + LumberConfig: interface
- + LumberRootConfig: interface
- + Module: interface
- + RelativeDirs: interface
- + RootConfigFile: interface
- + View: interface
- InitialModuleProperties: interface
- InitialRootModuleProperties: interface
- ModuleProperties: interface
- RootModule: interface
- RootModuleProperties: interface

Model

- + FieldEventHandler: interface
- + Model: interface

Converter

- Module: Module
- ModuleProps: ModuleProperties
- View: View
- configFile: ConfigFile
- configModules: ConfigModules
- model: Model | undefined
- globalConfig: GlobalConfig

- + run(): LumberConfig
- ChildNodesAtFileCheck(View, ChildNodeFiles): void
- ChildNodesCheck(View): void
- Convert(View): void
- convertAttributes(View): void
- convertExceptions(View): void
- convertTypes(View): void
- modelCheck(View): void
- setStyleFromProperties(View): void

CreateLitElement

- globalConfig: GlobalConfig
- project: Project

- + create(templateModule, templateRender): Promise<void>

GenerateActions

- + generateActions(moduleConfig, relDirs): Promise<void>
- + generateDefaultActions(moduleConfigs): Promise<void>

GenerateReducers

- + generateReducers(moduleConfig, relDirs): Promise<void>
- + generateDefaultReducers(moduleConfigs): Promise<void>

GenerateRuntimeFiles

- + generateRuntimeFiles(moduleConfigs): Promise<void>

GenerateStore

- + generateStore(moduleConfigs): Promise<void>
- + generateModuleStore(moduleConfig): Promise<void>

GenerateStyles

- + generateStyles(moduleConfig): Promise<void>

Generator

- + IGui_Generator: interface
- + Observer: interface

Generator

- + generateGui(JSON_MODULE): object
- + getChildNodes(View, Module): object
- addNodesThroughProperties(View): void
- getProperties(View): string

GlobalConfig

- instance: GlobalConfig
- + rootModule: string
- + debug: boolean

- + getInstance: GlobalConfig

Logger

- + indexLogger: winston.Logger
- + analyserLogger: winston.Logger
- + converterLogger: winston.Logger
- + fileGenerator: winston.Logger
- + litElementGenerator: winston.Logger
- container: Container
- getFormat: Function<Format>
- defaultTransports: object[]

LumberConfig

- + model: Model[]
- + store: object
- + runtimeFiles: object[]
- + styles: object[]
- + convertTypesAtFiles: Map<string, string>
- + importTypesDefAtFile: object[]
- + config: LumberConfig
- initialConfig: InitialLumberConfig
- location: string
- isReady: Promise<void>

- + loadConfigFromFile(filepath, inheritProps?): Promise<LumberModule | LumberRootModule>
- loadExternalFiles(): Promise<string[]>

- inheritableProps: type

LumberModule

- + model: Model[]
- + store: object
- + runtimeFiles: object[]
- + styles: object[]
- + convertTypesAtFiles: Map<string, string>
- + importTypesDefAtFile: object[]
- + config: LumberConfig
- initialConfig: InitialLumberConfig
- location: string
- isReady: Promise<void>

- + loadConfigFromFile(filepath, inheritProps?): Promise<LumberModule | LumberRootModule>
- loadExternalFiles(): Promise<string[]>

LumberRootConfig

- + initialConfig: InitialLumberRootConfig
- + config: LumberRootConfig

- + analyseModules(): Promise<ConfigModules>
- getChildNodesAtFiles(): Promise<string[]>

Utils

- + ismorphTextFormat: object
- + ismorphCompilerOptions: object
- + PathResolver(resolvesLocation, location): Function<string>
- + RelativePath(fromLocation, toLocation): Function<string>
- + ImportablePath(path): Function<string>
- + IsMeaningfulString(str): Function<boolean>
- + CamelCase(str): Function<string>
- + TestIfDirExistAndCreateDir(location): Promise<void>
- + serializeCommandObjs(commandObj): Function<object>

Validator

- + ajv: Ajv
- + lumberConfigId: string
- + lumberRootConfigId: string
- + lumberDefsid: string
- lumberConfigSchema: JSONSchemaType
- lumberRootConfigSchema: JSONSchemaType
- lumberDefsSchema: object