

Command
+ const SUCCESS + const FAILURE + const INVALID # static \$defaultName # static \$defaultDescription
+ __construct(string \$name=null) + ignoreValidationErrors() + setApplication(Application \$application=null) + setHelperSet(HelperSet \$helperSet) + getHelperSet() + getApplication() + isEnabled() + run(InputInterface \$input, OutputInterface \$output) + complete(CompletionInput \$input, CompletionSuggestions \$suggestions) + setCode(callable \$code) et 22 de plus... + static getDefaultName() + static getDefaultDescription() # configure() # execute(InputInterface \$input, OutputInterface \$output) # interact(InputInterface \$input, OutputInterface \$output) # initialize(InputInterface \$input, OutputInterface \$output)

GlipiCommandInterface
+ mustCheckMandatoryRequirements() + requiresUpToDateDb()

AbstractCommand
\$db # \$input # \$output # \$requires_db # \$requires_db_up_to_date # \$progress_bar
+ mustCheckMandatoryRequirements() + requiresUpToDateDb() # initialize(InputInterface \$input, OutputInterface \$output) # writelnOutputWithProgress Bar(\$messages, ProgressBar \$progress_bar, \$verbosity =OutputInterface::VERBOSITY_NORMAL) # outputSessionBufferedMessages (\$levels_to_output=[INFO, WARNING, ERROR]) # outputWarningOnMissingOptionnal Requirements() # askForConfirmation (bool \$default_to_yes=true) # warnAboutExecutionTime() # iterate(iterable \$iterable, ?callable \$message_callback=null, ?int \$count=null) # outputMessage(string \$message, int \$verbosity =OutputInterface::VERBOSITY_NORMAL)

SynchronizeUsersCommand
+ const ERROR_LDAP_CONNECTION_FAILED + const ERROR_LDAP_LIMIT_EXCEEDED

