
Authentication Library*Functions:*

- ❖ `rigAuthActivate(id[, code])`
- ❖ `rigAuthChangePassword(identity, oldPassword, newPassword)`
- ❖ `rigAuthClearForgottenPasswordCode(code)`
- ❖ `rigAuthClearLoginAttempts(identity[, exoirePeriod])`
- ❖ `rigAuthClearOTP([id])`
- ❖ `rigAuthDeactivate(id)`
- ❖ `rigAuthDeleteUser(id)`
- ❖ `rigAuthEmailCheck(email)`
- ❖ `rigAuthErrors([prefix][, suffix])`
- ❖ `rigAuthFetchConfigItem(item[, index])`
- ❖ `rigAuthForgottenPassword(identity [, extraData])`
- ❖ `rigAuthForgottenPasswordCheck(code)`
- ❖ `rigAuthGenerateQR([account])`
- ❖ `rigAuthGetAttemptsNum(identity)`
- ❖ `rigAuthGroup(id)`
- ❖ `rigAuthGroups()`
- ❖ `rigAuthIdentityCheck(identity)`
- ❖ `rigAuthIncreaseLoginAttempts(identity)`
- ❖ `rigAuthInGroup(group[, id])`
- ❖ `rigAuthIsAdmin([id])`
- ❖ `rigAuthIsMaxLoginAttemptsExceeded(ident)`
- ❖ `rigAuthLoggedIn()`
- ❖ `rigAuthLogin(identity, password[, remember])`



- ❖ `rigAuthMessages([prefix][, suffix])`
- ❖ `rigAuthOTPkeysMatch(challenge)`
- ❖ `rigAuthRegister(username, password, email[, extraData][, groupName])`
- ❖ `rigAuthRemoveFromGroup(groupIDs, userID)`
- ❖ `rigAuthResetPassword(identity, newPassword)`
- ❖ `rigAuthSetErrorDelimiters(startDelimiter, endDelimiter)`
- ❖ `rigAuthUpdate(id, data)`
- ❖ `rigAuthUser([id])`
- ❖ `rigAuthUserGroups([id])`
- ❖ `rigAuthUserHasOTP([id])`
- ❖ `rigAuthUsers([groupIDs])`
- ❖ `rigAuthUsernameCheck(username)`

Commands:

- ❖ `rigAuthSetErrorDelimiters startDelimiter, endDelimiter`
 - ❖ `rigAuthLogout`
 - ❖ `rigAuthSetMessageDelimiters startDelimiter, endDelimiter`
-

Benchmark Library*Functions:*

- ❖ `rigElapsedTime(startMarker, endMarker)`

Commands:

- ❖ `rigTimeMark marker`
-

Captcha Library*Functions:*

- ❖ `rigCaptchaStyle([nonce])`
- ❖ `rigGetCaptchaHash()`
- ❖ `rigGetCaptchaNum()`
- ❖ `rigCaptcha()`
- ❖ `rigCaptchaMathProblem([numOperators])`
- ❖ `rigValidCaptchaUserInput(userInput, hiddenHashField)`

Commands:

- ❖ `rigInitCaptcha config`
-

Calendaring Library

Functions:

- ❖ `rigCalGenerate([year, month][, cellDataArray])`
-

Config Library

Functions:

- ❖ `rigFetchConfigItem(configArrayIndex[, customConfigArray])`
- ❖ `rigLoadConfigFile(fileName[, storeAsConfigIndex][, suppressErrors])`
- ❖ `rigSiteURL()`
- ❖ `rigSystemURL()`

Commands:

- ❖ `rigSetConfigItem name, value[, configName][, moduleName]`
-

Database Library Active Record

Functions:

- ❖ `rigDbCountAllResults([table])`
- ❖ `rigDbDelete([table][, whereClause])`
- ❖ `rigDbEmptyTable([table])`

- ❖ `rigDbGet([table][, limit][, offset][, subquery][, query])`
- ❖ `rigDbGetWhere([table], whereClause[, limit][, offset])`
- ❖ `rigDbInsert(table, dataArray, lastInsertID)`
- ❖ `rigDbPrefix(table)`
- ❖ `rigDbTruncate([table])`
- ❖ `rigDbUpdate(table, data[, whereClause])`
- ❖ `rigDbValuesSet({field, value | fieldValueArray}[, escape])`

Commands:

- ❖ `rigDbCloseParenthesis`
- ❖ `rigDbDistinct`
- ❖ `rigDbFlushCache`
- ❖ `rigDbFrom table`
- ❖ `rigDbGroupBy {field | fieldArray}`
- ❖ `rigDbHaving {field, value | havingClause | fieldValueArray} [, escape]`
- ❖ `rigDbJoin table, condition[, option][,escape]`
- ❖ `rigDbLike field, match[, option]`
- ❖ `rigDbLimit limit[, offset]`
- ❖ `rigDbNotLike field, match[, option]`
- ❖ `rigDbOpenParenthesis`
- ❖ `rigDbOrderBy {field, option | field option, field option ...}`
- ❖ `rigDbOrHaving {field, value | havingClause | fieldValueArray} [, escape]`
- ❖ `rigDbOrLike field, match[, option]`
- ❖ `rigDbOrNotLike field, match[, option]`

- ❖ `rigDbOrWhere {field, value | fieldValueArray | customClause}`
 - ❖ `rigDbOrWhereIn field, valueArray`
 - ❖ `rigDbOrWhereNotIn field, valueArray`
 - ❖ `rigDbSelect fieldsList[, protect]`
 - ❖ `rigDbStartCache`
 - ❖ `rigDbStopCache`
 - ❖ `rigDbWhere {field, value | fieldValueArray | customClause}`
 - ❖ `rigDbWhereIn field, valueArray`
 - ❖ `rigDbWhereNotIn field, valueArray`
 - ❖ `rigSelectAvg field`
 - ❖ `rigSelectMax field[, newFieldName]`
 - ❖ `rigSelectMin field[, newFieldName]`
 - ❖ `rigSelectSum field[, newFieldName]`
-

Database Library Driver

Functions:

- ❖ `rigDbAffectedRows()`
- ❖ `rigDbCacheOff()`
- ❖ `rigDbCacheOn()`
- ❖ `rigDbEscape(string)`
- ❖ `rigDbFieldData(table)`
- ❖ `rigDbFieldExists(field, table)`
- ❖ `rigDbInsertString(table, dataArray)`
- ❖ `rigDbLastQuery()`

- ❖ rigDbListFields(table)
- ❖ rigDbListTables()
- ❖ rigDbPlatform()
- ❖ rigDbQuery(query)
- ❖ rigDbSimpleQuery(query)
- ❖ rigDbTableExists(table)
- ❖ rigDbTransStatus()
- ❖ rigDbUpdateString(table, dataArray, whereClause)
- ❖ rigDbVersion()
- ❖ rigEscapeLikeStr(string)
- ❖ rigProtectIdentifiers(table[, prefix])
- ❖ rigSqlDateTimeToIntDate(string)
- ❖ rigTransComplete()
- ❖ rigDbGetPragma(pragma)

Commands:

- ❖ rigDbCacheDelete segmentsList
- ❖ rigDbCacheDeleteAll fileNameToExclude
- ❖ rigDbTransOff
- ❖ rigDbTransStart
- ❖ rigDbTransStrict {TRUE | FALSE}
- ❖ rigDbSetPragma pragma, value

Database Library Result

Functions:

- ❖ rigDbCurrentRow()

- ❖ rigDbFirstRow()
- ❖ rigDbLastRow()
- ❖ rigDbNextRow()
- ❖ rigDbPreviousRow()
- ❖ rigDbResult([changeFieldsListDelimiter])
- ❖ rigDbRow([rowNumber])

Email Library

Functions:

- ❖ rigMailSend()
- ❖ rigPrintMailDebugger()

Commands:

- ❖ rigClearMail [clearAttachments]
- ❖ rigInitializeMail configArray
- ❖ rigMailAttach filePath
- ❖ rigMailBcc {email |emailList |emailArray}
- ❖ rigMailCC {email |emailList |emailArray}
- ❖ rigMailFrom email, name
- ❖ rigMailMessage message
- ❖ rigMailSubject subject
- ❖ rigMailTo {email |emailList |emailArray}
- ❖ rigReplyTo email, name
- ❖ rigSet_altMessage altMessage

Encrypt Library

Functions:

- ❖ rigDecode(string)
- ❖ rigEncode(string[, encryptionKey])
- ❖ rigSetCipher name, keyLength

File Uploading Library

Functions:

- ❖ rigDisplayUploadErrors([openTag, closeTag])
- ❖ rigUploadData()

Commands:

- ❖ rigDoUpload {fieldname}

Formvalidation Library

Functions:

- ❖ rigFormError(fieldName)
- ❖ rigFormValidRun([validationGroup])
- ❖ rigSetCheckbox(name, value[, bool])
- ❖ rigSetRadio(name, value[, bool])

- ❖ `rigSetSelect(name, value[, bool])`
- ❖ `rigSetValue(name[, value])`
- ❖ `rigValidationErrors()`

Commands:

- ❖ `rigAddValidationGrp [group]`
- ❖ `rigAddValidationRules field, label, rules, confArrayName`
- ❖ `rigSetErrorDelimiters openTag, closeTag`
- ❖ `rigSetMessage rule, message`
- ❖ `rigSetRules {fieldName, humanName, rules | rulesArray}`
- ❖ `rigSetPrefilledValues defaultFormFieldValues`

FTP Library*Functions:*

- ❖ `rigFtpPrintDebugger([messageType])`

Commands:

- ❖ `rigFtpUpload localPath, remotePath[, {TRUE | FALSE}][, {TRUE | FALSE}][, octal]`
- ❖ `rigFtpDownload remotePath, localPath`
- ❖ `rigFtpRenameMove oldFilePath, newFilePath[, {TRUE | FALSE}]`
- ❖ `rigFtpDeleteFile filePath`
- ❖ `rigFtpDeleteFolder folderPath[, {TRUE | FALSE}][, {TRUE | FALSE}]`

- ❖ `rigFtpListFiles folderPath[, {TRUE | FALSE}]`
- ❖ `rigFtpMirror localPath, remotePath`
- ❖ `rigFtpCreateFolder folderPath[, octal]`
- ❖ `rigFtpSetPermissions path, octal`
- ❖ `rigFtpChangeDir folderPath`
- ❖ `rigFtpClose`

HTML Table Library*Functions:*

- ❖ `rigGenerateTable([dataArray])`
- ❖ `rigMakeTableColumns(dataArray, numberOfColumns)`

Commands:

- ❖ `rigAddTableRow {valuesList | valuesArray}`
- ❖ `rigClearTableVars`
- ❖ `rigSetEmptyCellVal value`
- ❖ `rigSetTableCaption value`
- ❖ `rigSetTableHeading {valuesList | valuesArray, firstCharUppercase}`
- ❖ `rigSetTableTemplate templateData`

Image Library*Functions:*

- ❖ `rigDisplayImgErrors([openTag, closeTag])`
- ❖ `rigImageCrop()`
- ❖ `rigImageResize()`
- ❖ `rigImageRotate()`
- ❖ `rigImageWatermark()`
- ❖ `rigIMknownFonts()`
- ❖ `rigIMversion()`

Commands:

- ❖ `rigImageNewSettings configArray`
- ❖ `rigImgClear`

Input Library*Functions:*

- ❖ `rigIpAddress()`
- ❖ `rigSanitizeFilename(fileName[, relativePath])`
- ❖ `rigUserAgent()`
- ❖ `rigValidIP(ip[, which])`
- ❖ `rigVarCookie(name, xssFilter)`
- ❖ `rigVarGet(name, xssFilter)`
- ❖ `rigVarGetPost(name, xssFilter)`
- ❖ `rigVarPost(name, xssFilter)`
- ❖ `rigVarServer(name)`
- ❖ `rigXssClean(data[, testImage])`

Jquery Library

Functions:

- ❖ `rigJQsource()`
- ❖ `rigJQalertR(alert)`
- ❖ `rigJQconfirmR(variableName, question, {TRUE | FALSE})`
- ❖ `rigJQpromptR(variableName, prompt [, defaultValue][, {TRUE | FALSE}])`
- ❖ `rigJQsetR(selector, method, propertyValues [, {TRUE | FALSE}][, {TRUE | FALSE}])`
- ❖ `rigJQgetR(selector, method [, property])`
- ❖ `rigJQeffectR(selector, effect [, duration][, callback])`
- ❖ `rigJQanimateR(selector, properties [, duration][, easing][, callback])`
- ❖ `rigJQajaxR(ajaxOptions [, {TRUE | FALSE}])`
- ❖ `rigJQxhrRequestR(type [, URI][, data][, callback][, dataType][, selector])`
- ❖ `rigJQconditionR(condition, ifAction [, elseAction])`
- ❖ `rigJQeachOpenR([selector][, collection], callback)`
- ❖ `rigJQeachCloseR()`

Commands:

- ❖ `rigJQopen`
- ❖ `rigJQclose`

- ❖ `rigJQajaxForm submitBtnSelector, selectorList, postURI [, {TRUE | FALSE}][, dataToLoadURI][, dataContainerSelector]`
- ❖ `rigJQeventAlert selector, eventType, alert [, {TRUE | FALSE}][, eventData]`
- ❖ `rigJQalert alert`
- ❖ `rigJQconfirm variableName, question, {TRUE | FALSE}`
- ❖ `rigJQprompt variableName, prompt [, defaultValue][, {TRUE | FALSE}]`
- ❖ `rigJQset selector, method, propertyValues [, {TRUE | FALSE}][, {TRUE | FALSE}]`
- ❖ `rigJQeventOpen selector, event [, {TRUE | FALSE}][, eventData][, eventObject][, functionArguments]`
- ❖ `rigJQeventClose`
- ❖ `rigJQeventEffect triggerSelector, targetSelector, eventType, effect [, effectParameter][, {TRUE | FALSE}][, callback]`
- ❖ `rigJQeffect selector, effect [, duration][, callback]`
- ❖ `rigJQeventAnim triggerSelector, targetSelector, event, cssProperties [, duration][, easing][, callback]`
- ❖ `rigJQanimate selector, properties [, duration][, easing][, callback]`
- ❖ `rigJQajax ajaxOptions [, {TRUE | FALSE}]`
- ❖ `rigJQxhrRequest type [, URI][, data][, callback][, dataType][, selector]`
- ❖ `rigJQcallFunction [selector], functionName, functionArguments`

- ❖ `rigJQfunctionOpen functionName [, functionArguments]`
- ❖ `rigJQfunctionClose`
- ❖ `rigJQcb`
- ❖ `rigJQcondition condition, ifAction [, elseAction]`
- ❖ `rigJQeachOpen [selector][, collection], callback`
- ❖ `rigJQeachClose`
- ❖ `rigJQvar variables`
- ❖ `rigJQaddStatement statement`

Language Library

Functions:

- ❖ `rigLangLangLine(languageArrayKey)`
- ❖ `rigLangLoadLang(name, language)`

Loader Library

Functions:

- ❖ `rigLoadDatabase({connectArray | dsn}, returnID, enableActiveRecord)`
- ❖ `rigLoadView(name, returnData)`

Commands:

- ❖ `rigLoadConfig` `name`
- ❖ `rigLoaderLoadLibrary` `name[, configArray]`
- ❖ `rigLoadHelper` `name`
- ❖ `rigLoadLanguage` `name, language`
- ❖ `rigLoadModel` `name`
- ❖ `rigLoadPlugin` `name`
- ❖ `rigLoadStack` `name[, configArray] [, "front" | "behind"]`

Modules Library*Commands:*

- ❖ `rigLoadModule` `segments`

Functions:

- ❖ `rigFetchModuleSegment`(`number`, `noResultValue`)
- ❖ `rigModuleUriToAssoc`(`segmentOffset`, `defaultArray`)
- ❖ `rigRunModule`(`segments[, paramsArray]`)

Output Library*Functions:*

- ❖ `rigGetOutput()`

Commands:

- ❖ `rigEnableProfiler` `{TRUE | FALSE}`
- ❖ `rigOutputCache` `minutes`
- ❖ `rigSetHeader` `data`
- ❖ `rigSetStatusHeader` `code[, data]`
- ❖ `rigSetOutput` `data`

Pagination Library*Functions:*

- ❖ `rigCreatePaginationLinks()`
- ❖ `rigAjaxPaginationCode`(`[nonce]`)

Commands:

- ❖ `rigInitializePagination` `configArray`
- ❖ `rigAjaxPagination` `paginationData`, `paginationLinks`

Session Library*Functions:*

- ❖ `rigSessFlashdata`(`[item]`)
- ❖ `rigSessTempdata`(`[item]`)
- ❖ `rigSessUserdata`(`[item]`)

Commands:

- ❖ `rigKeepSessFlashdata` `item`
- ❖ `rigSessCreate`
- ❖ `rigSessDestroy`
- ❖ `rigSessMarkAsFlash` `{item | itemArray}`
- ❖ `rigSessMarkAsTemp` `{item[, seconds] | itemArray[, seconds]}`
- ❖ `rigSetSessFlashdata` `{item, value | itemValueArray}`
- ❖ `rigSetSessTempdata` `{item, value[, seconds] | itemValueArray[, , seconds]}`
- ❖ `rigSetSessUserdata` `{item, value | itemValueArray}`
- ❖ `rigUnsetSessUserdata` `{item | itemArray}`

URI Library*Functions:*

- ❖ rigAssocToURI(array)
 - ❖ rigFetchRsegment(segmentNumber[, default])
 - ❖ rigFetchSegment(segmentNumber[, default])
 - ❖ rigRSegmentArray()
 - ❖ rigRuriString()
 - ❖ rigRuriToAssoc(offset[, defaultKeyNames])
 - ❖ rigSegmentArray()
 - ❖ rigSlashRsegment(segmentNumber[, slashPosition])
 - ❖ rigSlashSegment(segmentNumber[, slashPosition])
 - ❖ rigTotalRsegments()
 - ❖ rigTotalSegments()
 - ❖ rigUriString()
 - ❖ rigUriToAssoc(offset[, defaultKeyNames])
-

- ❖ rigBrowser()
 - ❖ rigBrowserVersion()
 - ❖ rigIsBrowser()
 - ❖ rigIsMobile()
 - ❖ rigIsReferral()
 - ❖ rigIsRobot()
 - ❖ rigMobile()
 - ❖ rigReferrer()
 - ❖ rigRobot()
-

Useragent Library*Functions:*

- ❖ rigAcceptCharset(charSet)
- ❖ rigAcceptLang(language)
- ❖ rigAgentPlatform()
- ❖ rigAgentString()