BaseAPI

MaachangComet には、基本的な Javascript を拡張した基本 API が存在します。
これらの内容を次ページより、説明していきます。

カレントスクリプト名を取得。
function currentName()
戻り値 :現在実行対象スクリプト名が返されます。

カレントパスを取得。
function currentPath()
戻り値 :カレントパスが返されます.

スクリプトパスを取得。
function scriptPath()
戻り値 :スクリプトパスが返されます.

モデルパスを取得。
function modelPath()
戻り値 :モデルパスが返されます.

MaachangComet バージョンを取得。
function version()
戻り値 :MaachangComet バージョンが返されます.

maachangComet 更新日を取得。
function update()
戻り値 : maachangComet 更新日が返されます.

データベースが利用可能なチェック.
function isDatabase()
戻り値 : [true]の場合、データベースが利用可能です.
スクリプトキャッシュをリロード.
function reload()

スクリプトを実行

function eval(script)

script : 実行対象のスクリプト文字列を設定します.

戻り値 : スクリプト実行結果が返されます.

説明　 : 指定された文字列のスクリプトを実行します。

スクリプトを停止

function exit(result)

result : 停止結果の戻り値を設定.

特殊な条件として、”error”+httpState(404 とか 500 など)を返すことで、エラーステータスをクライアントに返せます。

説明　 : スクリプトを強制停止します。また、この命令を実行したからと言って、Java プロセスが停止することはありません。

Java ガベージコレクター実行

function gc()

Java トータルメモリー取得

function mtotal()

戻り値 : トータルメモリが返されます.

Java 最大メモリー取得

function mmax()

戻り値 : 最大メモリが返されます.

Java 空きメモリー取得

function mfree()

戻り値 : 空きメモリが返されます.

データタイプを取得

function valueof(val)

val　　：チェック対象のデータを設定します.

戻り値：データタイプが返されます。

“null”の場合、データは null です。

“string”の場合、データは文字列です。

“boolean”の場合、データは Boolean 型です。

“number”の場合、データは、数値です。

“function”の場合、データは、function(メソッド)です。

“date”の場合、データは、Date 型です。

“binary”の場合、データは、バイナリ(byte[])です。

“array”の場合、データは、配列です。

“map”の場合、データは、Map です。

“object”の場合、不明なオブジェクトです。

説明　：このメソッドは、typeof の拡張版です。

内容に合わせて Javascript に最適な値に変換

function parseValue(val)

val　　：変換対象の内容を設定します.

戻り値：javascript に最適に変換された内容が返されます.

説明　：この処理は主に Java と Javascript の差分を埋めるための処理。

NULL データ判別

function isNull(val)

val　　：対象内容を設定.

戻り値：null または undefined の場合は[true]

数値判別

function isNumeric(val)

val　　：対象内容を設定.

戻り値：数値の場合は[true]

文字列存在判別

function useString(val)

val　　：対象内容を設定.

戻り値：文字列が存在(trim 後に対しても)する場合は[true]

指定内容を java プリミティブ型に変換

function convertPrimitive(type,val)

type　：変換タイプを設定

“bool”または”boolan”の場合は、Boolean に変換します。

“char”または”character”の場合は、Charracter に変換します。

“short”の場合は、Short に変換します。

“int”または”integer”の場合は、Integer に変換します。

“long”の場合は、Long に変換します。

“float”の場合は、Float に変換します。

“double”の場合は、Double に変換します。

val　　：変換対処の内容を設定。

戻り値：変換結果が返されます。

指定内容のオブジェクト名を取得

function objectClassName(val)

val　　：対象内容を設定.

戻り値：指定オブジェクト名[Object.getClass().getName()]が返されます。

指定内容のオブジェクト名の名前を取得

function className(val)

val　　：対象内容を設定.

戻り値：指定オブジェクト名[Object.getClass().getName()]のオブジェクトの名前が返されます。

指定時間を停止

function sleep(val)

val　　：停止時間をミリ秒で設定します。

指定内容をコンソール出力

function p(val)

function puts(val)

function alert(val)

val　　：指定内容が表示されます。

Java オブジェクト配列長を取得

function arrayLength(val)

val　　：対象内容を設定.

戻り値：Java オブジェクト配列の場合、配列長が返されます。

Java オブジェクト配列内容を取得

function getArray(val,no)

val　　：対象内容を設定.

no　　：対象の項番を設定.

戻り値：Java オブジェクト配列の場合、指定項番の中身が返されます。

Java オブジェクト配列内容を設定

function setArray (val,no,set)

val　　: 対象内容を設定.

no　　: 対象の項番を設定.

set　　: 設定対象の情報を設定.

指定文字列をバイナリ変換

function convertStringByBinary(val,charset)

val　　: 対象内容を設定.

charset: 対象のキャラクタセット.[null]の場合は、”UTF8”

戻り値 : バイナリ情報が返されます。

指定バイナリを文字列変換

function convertBinaryByString(val,charset)

val　　: 対象内容を設定.

charset: 対象のキャラクタセット.[null]の場合は、”UTF8”

戻り値 : 文字列に変換された内容が返されます。

新しいバイナリを生成

function createBinary(size)

size　　: 新しいバイナリ長を設定.

戻り値 : 生成されたバイナリ情報が返されます。

指定バイナリを表示形式に変換

function toBinaryString(val)

val　　: バイナリを設定.

戻り値 : 16 進に変換された文字列が返されます。

指定バイナリ長を取得

function binaryLength(val)
val　　：バイナリを設定.
戻り値：バイナリ長が返されます。

1バイトのバイナリ内容を取得

function getBinary(val,no)
val　　：バイナリを設定.
no　　：対象の項番を設定.
戻り値：1バイトのバイナリ内容が返されます。

1バイトのバイナリ内容を設定

function setBinary(val,no,set)
val　　：対象内容を設定.
no　　：対象の項番を設定.
set　　：設定対象の情報を設定.

指定内容がバイナリか取得

function isBinary(val)
val　　：対象内容を設定.
戻り値：[true]の場合、バイナリです。

バイナリを Base64 に変換

function encodeBase64(val)
val　　：バイナリを設定.
戻り値：Base64 に変換された内容が返されます。

Base64 をバイナリに変換

function decodeBase64(val)

val　　: 文字列を設定.

戻り値 : バイナリ内容が返されます。

文字列を Trim

function trim(val)

val　　: 文字列を設定.

戻り値 : 前後のスペースを省きます。

文字列を半角変換

function toLowerCase(val)

val　　: 文字列を設定.

戻り値 : 半角変換された内容が返されます。

文字列を全角変換

function toUpperCase(val)

val　　: 文字列を設定.

戻り値 : 全角変換された内容が返されます。

文字列の最初の内容と一致しているかチェック.

function startsWith(val,c)

val　　: 文字列を設定.

c　　: チェック対象の内容を設定.

戻り値 : [true]の場合、一致しています。

文字列の最後の内容と一致しているかチェック.

function endsWith(val,c)

val　　: 文字列を設定.

c　　: チェック対象の内容を設定.

戻り値 : [true]の場合、一致しています。

指定文字列を Date 変換し、その内容から、総時間(数値)変換。

function dateParse(format,val)

format : 日付パースフォーマットを設定(SimpleDateFormat).

val　　: 変換対象内容を設定.

戻り値 : 変換された総時間(数値)が返されます。

指定文字列を Date 変換します。

function dateFormat(format,val)

format : 日付パースフォーマットを設定(SimpleDateFormat).

val　　: 変換対象内容を設定.

戻り値 : 変換された Date オブジェクトが返されます.

現在時刻を文字列で取得.

function dateToString(mode)

mode　　: [true]の場合、スペースを に変換して取得します.

戻り値 : 現在時刻が文字列で返されます.

現在時刻を文字列で取得.

function date ()

戻り値 : 現在時刻が文字列で返されます.

指定 Date オブジェクトを文字列変換.

function formatDateByString(mode,val)

mode　: [true]の場合、スペースを に変換して取得します.

val　: 変換対象の Date オブジェクトを設定します.

戻り値 : 指定時刻が文字列で返されます.

文字連結オブジェクト.

StrBuf オブジェクト

このオブジェクトを利用する場合は、インスタンス生成を行う必要があります.

例:var x = new StrBuf() ;

文字連結バッファを生成

function StrBuf::create(len)

len　: 生成開始バッファ長を設定します.

文字連結バッファを生成

function StrBuf::clear()

文字連結バッファを生成

function StrBuf::append(val)

val　: 連結対象文字列を設定.

戻り値 : このオブジェクトが返されます.

連結された文字列長を取得

function StrBuf::length()

戻り値 : 文字列長が返されます.

連結された文字列を取得

function StrBuf::toString()
戻り値 : 文字列が返されます.

指定ドメインの IP 一覧を取得

function ip(domain)
domain : ドメイン名を取得
説明　 : IP アドレス一覧がコンソール出力されます.

指定文字列を HTML 表示可能な形式に変換

function convertViewHtml(val)
val　　: 対象の文字列を設定.
戻り値 : HTML 表示可能な形式に変換された内容が返されます.
説明　 : HTML タグや、改行などを、HTML に表示可能な形式に変換します.

指定文字列に動的な HTML を含むかチェック.

function isDirectHtmlTag(val)
val　　: 対象の文字列を設定.
戻り値 : [true]の場合、動的な HTML タグを含んでいます。
説明　 : ここで言う動的なタグとは、script タグや object タグなどを指します。

指定文字列に動的な HTML を含んでいる場合、その個所を HTML 表示可能形式に変換

function convertDirectHtml(val)
val　　: 対象の文字列を設定.
戻り値 : 動的な HTML タグを HTML 表示可能な形式に変換された内容が返されます.
説明　 : ここで言う動的なタグとは、script タグや object タグなどを指します。

コンフィグ内容を取得

function getConfig(sec,key,no)

sec　　：コンフィグセクション名を設定

key　　：コンフィグキー名を設定

no　　：コンフィグ項番を設定

戻り値：コンフィグ内容が返されます.

説明　　：MaachangComet で定義されているコンフィグ内容を取得するメソッド.

指定ディレクトリを生成

function Io.mkdir(val)

val　　: 対象の文字列を設定.

指定ファイル/ディレクトリを削除

function Io.deleteFile(val)

val　　: 対象の文字列を設定.

指定ファイル/ディレクトリを移動

function Io.moveFile(src,dest)

src　　: 移動元の文字列を設定.

dest　: 移動先の文字列を設定.

指定ファイル長を取得

function Io.fileLength(val)

val　　: 対象の文字列を設定.

戻り値 : ファイル長が返されます.

指定ファイル時間を取得

function Io.fileLastTime(val)

val　　: 対象の文字列を設定.

戻り値 : ファイル更新時間が返されます.

指定ファイルフルパスを取得

function Io.getFullPath(val)

val　　: 対象の文字列を設定.

戻り値 : フルパス名が返されます.

指定ファイルパスからファイル名を取得

function Io.getFileName(val)
val　　: 対象の文字列を設定.
戻り値 : パスからファイル名が返されます.

指定ファイルが存在するかチェック

function Io.isFileExists(val)
val　　: 対象の文字列を設定.
戻り値 : [true]の場合、指定ファイルは存在します.

指定ディレクトリが存在するかチェック

function Io.isDirExists(val)
val　　: 対象の文字列を設定.
戻り値 : [true]の場合、指定ディレクトリは存在します.

指定ファイル/ディレクトリが読み込み可能かチェック

function Io.isReadFile(val)
val　　: 対象の文字列を設定.
戻り値 : [true]の場合、読み込み可能です.

指定ファイル/ディレクトリが書き込み可能かチェック

function Io.isWriteFile(val)
val　　: 対象の文字列を設定.
戻り値 : [true]の場合、書き込み可能です.

指定ディレクトリ以下のファイル名群を取得.

function Io.getDirList(val)
val　　: 対象の文字列を設定.
戻り値 : ファイル名一覧が返されます

指定ファイル内容を文字列で取得.

function Io.readFileByString(name,charset)

name　: ファイル名を設定します.

charset: キャラクタセットを設定します.[null]の場合は”UTF8”

戻り値 : ファイル内容が返されます.

指定ファイル内容をバイナリで取得.

function Io.readFileByBinary(name)

name　: ファイル名を設定します.

戻り値 : ファイル内容が返されます.

指定文字列をファイルに書き込む.

function Io.writeFileByString(mode,name,value,charset)

mode　: [true]の場合上書きします.[false]の場合、追加で書き込みます.

name　: ファイル名を設定します.

value　: 書き込み対象の内容を設定します.

charset: キャラクタセットを設定します.[null]の場合は”UTF8”

指定バイナリをファイルに書き込む.

function Io.writeFileBinary(mode,name,value)

mode　: [true]の場合上書きします.[false]の場合、追加で書き込みます.

name　: ファイル名を設定します.

value　: 書き込み対象の内容を設定します.

指定ファイル内容のスクリプトを実行

function Io.executionJS(name,charset)

name　: ファイル名を設定します.

charset: キャラクタセットを設定します.[null]の場合は”UTF8”

戻り値 : スクリプト実行結果が返されます.

スタートアップグローバルパラメータ定義.

function Io.putStartupParam(key,value)

key　　: 追加対象のキー名を設定します.

value　: 追加対象の要素を設定します.

スタートアップグローバルパラメータ削除

function Io.removeStartupParam(key)

key　　: 削除対象のキー名を設定します.