

htmlAPI

MaachangComet には、HTML 処理系の 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。

HTTP **リクエストオブジェクトを取得**

```
function getRequest()
```

戻り値 : HTTP リクエストオブジェクトが返されます

HTTP **リクエストヘッダオブジェクトを取得**

```
function getRequestHeader()
```

戻り値 : HTTP リクエストヘッダオブジェクトが返されます

HTTP **リクエスト Body をバイナリ取得**

```
function getRequestBodyByBinary()
```

戻り値 : HTTP リクエスト Body がバイナリで返されます.

HTTP **リクエスト Body を文字列で取得**

```
function getRequestBodyByBinary(charset)
```

charset: キャラクタセットを設定します.[null]の場合 " UTF8 " で変換します.

戻り値 : HTTP リクエスト Body が文字列で返されます.

HTTP **リクエストバージョンを取得**

```
function getHttpVersion()
```

戻り値 : HTTP リクエストバージョンが返されます.

HTTP **リクエストメソッドを取得**

```
function getHttpMethod()
```

戻り値 : HTTP リクエストメソッドが返されます.

HTTP **リクエスト URL を取得**

```
function getURLPath()
```

戻り値 : HTTP リクエスト URL が返されます.

正常メッセージを設定

```
function setSuccessMessage(msg)
```

msg : 正常メッセージを設定します.

説明 : このメソッドは、MHTML で表示する処理結果表示などで利用されます.

異常メッセージ(Validate エラー)を設定

```
function setErrorMessage(msg)
```

msg : 異常メッセージを設定します.

説明 : このメソッドは、MHTML で表示する処理結果表示などで利用されます.

異常メッセージ(Validate エラー)が存在するかチェック

```
function isErrorMessage()
```

戻り値 : 異常メッセージが存在する場合は[true].

HTTP レスポンスオブジェクトを取得

```
function getHttpdResponse()
```

戻り値 : HTTP レスポンスオブジェクトが返されます

HTTP レスポンス送信における GZIP を OFF に設定

```
function gzipOff()
```

HTTP レスポンス送信における GZIP を ON に設定

```
function gzipOn()
```

HTTP レスポンス送信における HTTP キャッシュを OFF に設定

```
function cacheOff()
```

HTTP レスポンス送信における HTTP キャッシュを ON に設定

```
function cacheOn()
```

HTTP レスポンス送信における KEEP-ALIVE を OFF に設定

```
function keepAliveOff()
```

HTTP レスポンス送信における KEEP-ALIVE を ON に設定

```
function keepAliveOn()
```

HTTP レスポンスヘッダを追加

```
function addResponseHeader(key,value)
```

key : ヘッダーキーを設定します.

value : ヘッダパラメータを設定します.

説明 : HTTP レスポンスヘッダを追加します. 既にキー名の内容が存在する場合、その条件に内容を追加します.

HTTP レスポンスヘッダを設定

```
function setResponseHeader(key,value)
```

key : ヘッダーキーを設定します.

value : ヘッダパラメータを設定します.

説明 : HTTP レスポンスヘッダを設定します. 既にキー名の内容が存在する場合、その条件をクリアして、新しく設定します.

HTTP レスポンスヘッダにコンテンツタイプを設定

```
function setContentType(value)
```

value : コンテンツタイプを設定します.

指定 URL にリダイレクト

```
function redirect(url)
```

url : リダイレクト先の URL を設定します.

指定URL にリダイレクト (POST)

```
function redirectPost(url)
```

url : リダイレクト先の URL を設定します.

説明 : 現在の Method が POST で、HTTP1.1 場合、HTTP ステータス(307)で、リダイレクトします。
また、既にレスポンスに書き込んでいる場合は、リダイレクトしません。

指定相対URL にフォワード

```
function forwardMHTML(url)
```

url : フォワード先の URL を設定します.

説明 : 指定 URL へフォワードします。また、既にレスポンスに書き込んでいる場合は、フォワードしません。

指定相対URL のファイルをインクルード

```
function include(url)
```

url : インクルード先の URL を設定します.

説明 : 指定 URL の内容をインクルードします。また、この処理は、通常 Inner タイプのスクリプトを指定します。

対象ファイル内容をレスポンスにセット.

```
function responseToFile(name)
```

name : 対象のファイルを設定します.

対象ファイル名のMimeTypesを取得.

```
function getMimeType(name)
```

name : 対象のファイルを設定します.

戻り値 : Mime タイプが返されます.

指定条件のバイナリをレスポンスにセット

```
function responseToBinary(binary,mime)
```

binary : 対象のバイナリを設定します.

mime : 対象の Mime タイプを設定します.

対象文字列をレスポンスに設定

```
function httpPrint(val)
val      : 対象の文字列を設定します.
```

対象文字列をレスポンスに設定(改行あり)

```
function httpPrintln(val)
val      : 対象の文字列を設定します.
```

指定内容をURL エンコード

```
function urlEncode(url,charset)
url      : 変換対象の URL 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 " UTF8 " で変換します.
戻り値  : URL エンコードされた内容が返されます.
```

指定内容をURL デコード

```
function urlDecode(url,charset)
url      : 変換対象の URL 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 " UTF8 " で変換します.
戻り値  : URL デコードされた内容が返されます.
```

指定リクエストパラメータをバイナリオリジナル名として取得

```
function paramOriginal(name)
name      : 対象のパラメータ名を設定します.
戻り値  : バイナリオリジナル名が返されます.
```

指定リクエストパラメータをバイナリとして取得

```
function paramBinary(name)
name      : 対象のパラメータ名を設定します.
戻り値  : バイナリが返されます.
```

ブラウザ認証ユーザ名を取得

```
function getSuccessAuthUser(name)
```

戻り値 : ブラウザ認証ユーザ名が返されます.

ブラウザ認証パスワードを取得

```
function getSuccessAuthPasswd(name)
```

戻り値 : ブラウザ認証パスワードが返されます.

セッション情報を追加

```
function putSession(key,value)
```

key : 追加対象のセッションキー名を設定します.

value : 追加対象のセッション要素を設定します.

セッション情報を削除

```
function removeSession(key)
```

key : 削除対象のセッションキー名を設定します.

セッション情報を取得

```
function getSession(key)
```

key : 取得対象のセッションキー名を設定します.

戻り値 : 対象の要素が返されます.

セッション ID を取得

```
function getSessionId()
```

戻り値 : セッション ID が返されます.

コメントグループID を取得

```
function getGroupId()
```

戻り値 : コメントグループ ID が返されます

コメントパラメータを取得

```
function getCometArgs()
```

戻り値 : コメントパラメータが返されます

説明 : コメントレスポンス処理で利用します.

コメント実行を取得

```
function executionComet( args )
```

args : コメントパラメータを設定します.

説明 : コメントイベント発生させる場合に実行します。また、groupId が存在しない場合は、実行されません.

コメントトリガーからのコメント呼び出し

```
function triggerComet( args )
```

args : コメントパラメータを設定します.

説明 : コメントトリガーで、コメントイベント発生させる場合に、実行する必要があります.

JComet 形式で、レスポンス送信.

```
function sendJComet( res,args )
```

res : 処理結果基本値を設定します。この情報は、接続先とのデータ同期を取る情報などを設定します.

args : JSON 形式の内容を設定します.

説明 : JSONP 通信で利用します。主な利用想定は、JComet のレスポンス送信用など。

JSONP **形式でレスポンス送信**

```
function jsonp( args )
```

args : JSON 形式の内容を設定します.

説明 : JSONP 通信で利用します。

JSON **形式の内容を文字列変換**

```
function encodeJSON( args,buf )
```

args : JSON 形式の内容を設定します.

buf : StrBuf オブジェクトを設定します.[null]の場合は、コンソールか、httpPrint 出力されます.

戻り値 : 文字列変換された内容が返されます.

文字列の内容を JSON **形式に変換**

```
function decodeJSON( args )
```

args : 文字列を設定します.

戻り値 : JSON 形式の内容が返されます.