1. サーバ起動・停止 API

1.1. サーバ起動

名称	HttpServer #start

概要説明
Port 番号を指定して、サーバを起動する。

戻り値	
名称	コメント
-	-

引数	
名称	コメント
port	起動するサーバの Port 番号

処理説明
① HttpServer は、指定された Port 番号で Tcpsocket の Listen を開始する。

使用例

var port = 3000;
server = new HttpServer();
server.start(port);

1.2. サーバ停止

名称	HttpServer #stop

概要説明
サーバ停止する。

戻り値	
名称	コメント
-	なし

引数	
名称	コメント
callback	サーバ停止通知コールバック関数

処理説明
① サーバ停止し、指定された callback をコールし、サーバ停止を通知する。

使用例

function onstop() {
・・・
}
server.stop(onstop);

2. クライアントからの要求処理

2.1. パス Handler 登録

名称	HttpServer #get

概要説明
パス Hander を登録する。

戻り値	
名称	コメント
-	なし

引数		
名称	コメント	
path	クライアントからの要求 Path	
function	登録した Path にクライアントから要求があった場合に実行する関数	
	引数	
	request	クライアントからのリクエスト
	response	クライアントへのレスポンス
	oncomplete	結果通知関数

処理説明
クライアントからの要求 Path、および、登録した Path にクライアントから要求があった場合に実行する関数 function を登録する。 function には、 ・クライアントからのリクエスト（httpd.js の Request クラス） ・クライアントへのレスポンス（httpd.js の Response クラス） ・結果通知関数 が引数として設定され、サーバアプリでは、 ・クライアントからのリクエスト情報取得（「2.1.1 リクエスト取得処理」参照） ・クライアントへのレスポンス情報設定（「2.1.2 レスポンス設定処理（データ）」、「2.1.3 レスポンス取得処理（ファイル）」参照） ・クライアントへの正常、失敗の結果通知（「2.1.4 結果通知処理」参照） を行うことができる。

使用例

■サーバ側スクリプト

```
server.get('/request', onRequest);

function onRequest(request, response, oncomplete) {
   oncomplete();//結果通知処理.
}
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/request';
xhr.open('GET',url,true);
xhr.onload = function(e){
   if (xhr.readyState === 4 && xhr.status === 200) {
      //xhr のレスポンスを取得する.
   }
}
```

2.1.1. リクエスト取得処理

名称	Request

概要説明
クライアントからのリクエスト情報を格納しているクラス.

処理説明	
パス Handler 登録した Path へリクエストが行われた際、function の第 1 引数でリクエスト情報（httpd.js で定義する Request クラス）が通知される。 Request クラスのメンバー、メソッドから以下のリクエスト情報が取得できる。	
メンバー／メソッド	**説明**
Scheme	スキーマ名
Host	ホスト
Port	ポート番号
Method	メソッド（GET/POST）
httpVersion	HTTP バージョン
Path	要求 Path
queryString	クエリ（リクエスト Path 以降の'?'を付与し指定するクエリ）
getHeader(name)	ヘッダ情報取得（name 指定）
hasHeader(name)	ヘッダ有無（name 指定）
bodyInputStream	body 部の Stream
bodyBuffer	body 部のデータ

使用例

（例）クライアントから '/request' で要求（クエリ付与）があった場合、サーバ側でクエリを参照する

■サーバ側スクリプト

```
server.get('/request', onRequest);

function onRequest(request, response, oncomplete) {
   var query = request. queryString; // 'id=12345670'を取得.
   oncomplete();//結果通知処理.
}
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/request?id=12345670';
xhr.open('GET',url,true);
```

2.1.2. レスポンス設定処理(データ)

名称	Response#write

概要説明
レスポンスする body へデータを設定する。

戻り値	
名称	コメント
-	なし

引数	
名称	コメント
Data	String 型もしくは、uint8Array 型のデータ

処理説明
String 型もしくは、uint8Array 型のデータをレスポンスする body へ設定する。

使用例

(例)クライアントから '/request' で要求があった場合、'abcdef'をクライアントへ返す。

■サーバ側スクリプト

```
server.get('/request', onRequest);

function onRequest(request, response, oncomplete) {
   var data = 'abcdef';
   response.write(data);
   oncomplete();//結果通知処理.
}
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/request';
xhr.open('GET',url,true);
xhr.onload = function(e) {
   if (xhr.readyState === 4 && xhr.status === 200) {
      //xhr のレスポンスを取得する.
      var responseTxt = xhr.responseText;//サーバから受信したデータを取得.
   }
}
```

2.1.3. レスポンス設定処理（ファイル）

名称	Response# writeFileResponse

概要説明
ファイルパスを指定し、ファイルデータをレスポンスする。

戻り値	
名称	コメント
-	なし

引数			
名称	コメント		
localPath	レスポンスする body へ設定するファイルデータのパス		
readFile	ファイル読み取り処理関数（サーバ側スクリプトで定義する）		
	引数		
	localPath	レスポンスする body へ設定するファイルデータのパス	
	successCB	ファイル読み取り処理成功時の通知関数	
		引数	
		fileObj	ファイルオブジェクト
		modDateTime	ファイルの更新日時
	errorCB	ファイル読み取り処理失敗時の通知関数	
		引数	
		e	エラーステータス（「2.1.4 結果通知処理」参照）
req	クライアントからのリクエスト		
oncomplete	結果通知関数		

処理説明
ファイルパスを指定し、ファイルデータをレスポンスする。

使用例

（例）クライアントから‘/request’で要求があった場合、ファイルデータをクライアントへ返す。

■サーバ側スクリプト

```
server.get('/request', onRequest);

function onRequest(req, res, oncomplete) {
  var readFile = function(fpath, successCb, errorCb) {
    var storageName = 'pictures';
    var storage = window.navigator.getDeviceStorage(storageName);
      if (!storage) {
        errorCb(HTTP_500);
        return;
      }
      var obj = storage.get(fpath);
      obj.onsuccess = function() {
        var file = obj.result;
        successCb(file, file.lastModifiedDate.getTime());
      };
      obj.onerror = function objectOnerror(e) {
        errorCb(HTTP_404);
      };
    };
```

使用例

```
    var localPath = '/sdcard/xxx/yyy.jpg';
    res.writeFileResponse(localPath, readFile, req, oncomplete);
}

-------------------------------
■クライアント側スクリプト
function setImage() {
  var image = document.createElement('img');
  image.onload = function () {
    // 画像データ取得成功.
  };
  image.onerror = function () {
    // 画像データ取得失敗.
  };
  image.src = '/request';
}
```

2.1.4. 結果通知処理

名称	oncomplete

概要説明
パス Handler 登録した関数での処理結果（成功／失敗）を通知する。

戻り値	
名称	コメント
-	なし

引数	
名称	コメント
e	エラーコード（失敗時のみ）

処理説明
パス Handler 登録した関数での処理結果（成功／失敗）を通知する関数で、第 3 引数で渡される。（「2.1 パス Handler 登録」参照） パス Handler 登録した関数において、 ・正常終了した場合は、引数なしで、結果をクライアント側へ通知する。 ・何らかのエラーが発生した場合は、引数にエラーステータスを設定し、結果をクライアント側へ通知する。 サポートしているエラーステータスは以下

ステータス	説明
HTTP_400	Bad Request
HTTP_403	Forbidden
HTTP_404	Not Found
HTTP_500	Internal Server Error
HTTP_501	Not Implemented

使用例

■サーバ側スクリプト

```
server.get('/request', onRequest);

function onRequest(request, response, oncomplete) {
   ・・・
   if (成功) {
       oncomplete();//成功結果通知
    }
    else {
       oncomplete(HTTP_500);//失敗結果通知
    }
}
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/request';
xhr.open('GET',url,true);
xhr.onload = function(e){
   if (xhr.readyState === 4 && xhr.status === 200) {
      //xhr のレスポンスを取得する.
   }
}
```

2.2. 指定ディレクトリ登録(アプリ内ディレクトリ指定)

名称	HttpServer #get

概要説明

クライアントから要求に対応するサーバアプリ内のディレクトリを指定する。

戻り値

名称	コメント
-	なし

引数

名称	コメント
path	クライアントからの要求 Path
directory	登録するディレクトリ

処理説明

クライアントから要求に対応するサーバアプリ内のディレクトリを登録する。
登録後、クライアントからの"**要求 Path＋ファイル名**"でリクエストがあった場合、"**directory＋ファイル名**"のファイルデータをクライアントへレスポンスする。

使用例

■サーバ側スクリプト

```
this.server.get('/', './public');
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/index.html'; //サーバ側の'./public'ディレクトリ配下の index.html を取得.
xhr.open('GET',url,true);
xhr.onload = function(e){
  if (xhr.readyState === 4 && xhr.status === 200) {
    //xhr のレスポンスを取得する.
  }
}
```

2.3. 指定ディレクトリ登録(SD カード内ディレクトリ指定)

名称	HttpServer #get

概要説明
クライアントから要求に対応する SD カード内のディレクトリを指定する。

戻り値	
名称	コメント
-	なし

引数	
名称	コメント
path	クライアントからの要求 Path
directory	登録するディレクトリ ※ディレクトリの先頭は"/sdcard"で指定すること

処理説明
クライアントから要求に対応するサーバアプリ内のディレクトリを登録する。 登録後、クライアントからの"**要求 Path＋ファイル名**"でリクエストがあった場合、"**directory＋ファイル名**"のファイルデータをクライアントへレスポンスする。

使用例

■サーバ側スクリプト

```
this.server.get('/sd', '/sdcard/testserver');
```

■クライアント側スクリプト

```
var xhr = new XMLHttpRequest();
var url = '/sd/index.html'; //サーバ側の' /sdcard/testserver'ディレクトリ配下の index.html を取得.
xhr.open('GET',url,true);
xhr.onload = function(e){
   if (xhr.readyState === 4 && xhr.status === 200) {
      //xhr のレスポンスを取得する.
   }
}
```