

libcurl

개요

libcurl은, 다양한 타입의 서버와 프로토콜 상에서 커뮤니케이션과 연결들을 쉽게 할 수 있도록 해주는 라이브러리입니다.

libcurl은 현재 http, https, ftp, gopher, telnet, dict, file 및 ldap를 지원합니다.

설치 방법

현재, cocos2dx는 libcurl을 지원하려고 하는 단계에 있습니다.
그래서 기본적인 헤더파일이나 lib는, 처음에 다운받았던 cocos2dx 폴더에 첨가되어 있어, libcurl을 사용하려면 이 라이브러리 파일들을 복사해와 프로젝트에 추가하면 됩니다.

파일이 들어있는 경로는, 다운받았던 cocos2dx폴더 내의 cocos2dx로 들어갑니다.
그다음, platform, third_party로 들어가, 해당 os에 맞는 헤더파일과 라이브러리 파일을 복사하여 프로젝트에 추가하면 됩니다.

간단한 예제 - 1

현재 네트워크와 연결된 대상에 대하여 상태와 타입 응답 코드등을 출력하는 예제입니다.

```
curl_global_init(CURL_GLOBAL_ALL);

CURL *ctx = curl_easy_init();

curl_easy_setopt(ctx, CURLOPT_URL, "http://www.naver.com");

curl_easy_setopt(ctx, CURLOPT_WRITEHEADER, stdout);
//헤더를 출력

curl_easy_setopt(ctx, CURLOPT_WRITEDATA, stdout);
//바디를 출력

curl_easy_perform(ctx);
//데이터를 긁어옴

double statDouble;
int statLong;
char *statString = NULL;

if(curl_easy_getinfo(ctx, CURLINFO_HTTP_CODE, &statLong) == CURLE_OK)
    printf("Response code: %d \n", statLong);
//응답 코드를 출력

if(curl_easy_getinfo(ctx, CURLINFO_CONTENT_TYPE, &statString) == CURLE_OK)
    printf("Content Type %s \n", statString);
//컨텐츠의 타입을 출력

if(curl_easy_getinfo(ctx, CURLINFO_SIZE_DOWNLOAD, &statDouble) == CURLE_OK)
    printf("Download size: %lf \n", statDouble);
```

//파일의 크기를 출력

```
if(curl_easy_getinfo(ctx, CURLINFO_SPEED_DOWNLOAD, &statDouble) == CURLE_OK)
    printf("Speed %lf Wn", statDouble);
//속도를 출력
```

```
curl_easy_cleanup(ctx);
curl_global_cleanup();
//해제
```

출력화면:)

```
<div id="wrap">
<div style="position:relative; width:880px; margin:0 auto; z-index:11;">
<div id="da_top"></div>
<div id="da_brand"></div>
<div id="da_hook_t"></div>
<div id="da_hook_b"></div>
<div id="da_stake"></div>
<div id="da_casthelp"></div>
<div id="da_expwide"></div>
</div>
<div id="header">
<div class="gnb">
<h2 class="blind">홈 유형 선택</h2>
<ul class="nav_hm">
<li id="nav_hm1" class="m"><a href="http://www.naver.com/" onclick="clickcr(this,'HNB.cast','','',event);" accesskey="1"><span class="blind">네이버</span></a></li>
<li id="nav_hm2" class="m2"><a href="http://se.naver.com/" onclick="clickcr(this,'HNB.search','','',event);" accesskey="2"><span class="blind">네이버SE</span></a></li>
<li id="nav_hm3" class="m3"><a href="http://me.naver.com/" onclick="clickcr(this,'HNB.desk','','',event);" accesskey="3"><span class="blind">네이버me</span></a></li>
</ul>
<h2 class="blind">네이버 설정</h2>
<ul class="naver_set">
<!--[if IE]>
<li><a href="#mhome" id="mhome" onclick="clickcr(this,'top.mhome','','',event);this.style.behavior='url(#default#homepage)';this.setHomePage('http://www.naver.com');ToolBar.popup();return false;">네이버를 시작페이지로</a></li>
<li class="toolbar"><a href="http://cdn.naver.com/naver/toolbar/Naver_Toolbar.exe" onclick="clickcr(this,'top.toolbar','','',event);">툴바설치</a></li>
<!--[endif]>
<li class="naverapp">
<a href="#" class="btn" onclick="clickcr(this,'top.app','','',event);Andro_MSG.toggle();return false;">네이버 앱 다운로드</a>
<div id="naverapply" class="ly_hnb hide">
<p class="tit">모바일 라이프의 경쟁력, <a href="http://campaign.naver.com/naverapp"><strong>네이버 앱</strong></a>을 설치하세요!</p>
<fieldset>
<legend>다운로드URL 문자받기</legend>
<span class="txtsm">다운로드URL 문자받기</span>
<label class="blind" for="app_phone">휴대폰번호 입력</label>
<input class="empty" type="text" value="" id="app_phone" />
<button type="button" class="btnsm"><span class="blind">보내기</span></button>
</fieldset>
<p class="desc">입력하신 번호는 저장되지 않고, 전송은 무료입니다.</p>
<p class="qr_qr_naver"><span class="blind">네이버 앱 QR코드</span></p>
<ul class="app">
<li class="f"><a href="#naver"><strong>네이버 앱</strong></a></li>
<li><a href="#map">지도</a></li>
<li><a href="#ndrive">N드라이브</a></li>
<li><a href="#books">북스</a></li>
<li><a href="#music">뮤직</a></li>
<li><a href="#line">라인</a><span class="new">new</span></li>
</ul>
<p class="more"><a href="http://mobile.naver.com/home/index.nhn" title="네이버 앱 더보기" onclick="clickcr(this,'top*.appmore','','',event)">더보기</a></p>
```

간단한 예제 - 2

위 예제에서 확장하여, body부분이 stdout에 직접 출력되는 형식이 아닌, 따로 메모리를 할당하여 데이터를 담은 형태로 관리하여 출력하는 예제입니다.

```
struct MemoryStruct {
    char *memory;
    size_t size;
};
```

```
static size_t WriteMemoryCallback(void *ptr, size_t size, size_t nmemb, void *data)
{
    size_t realsize = size * nmemb;
    MemoryStruct *mem = (MemoryStruct*)data;
```

```
    mem->memory = (char*)realloc(mem->memory, mem->size + realsize + 1);
```

```
    if( mem->memory == NULL )
    {
        //메모리 넘침
    }
```

```
    memcpy(&(mem->memory[mem->size]), ptr, realsize);
    mem->size += realsize;
    mem->memory[mem->size] = 0;
```

```
    return realsize;
```

```
}

MemoryStruct memData;
memData.size = 0;
memData.memory = (char*)malloc(1);
memData.memory[0] = NULL;

curl_global_init(CURL_GLOBAL_ALL);

CURL *ctx = curl_easy_init();
CURLcode res;

curl_easy_setopt(ctx, CURLOPT_URL, "http://www.naver.com");
curl_easy_setopt(ctx, CURLOPT_WRITEDATA, &memData);
curl_easy_setopt(ctx, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);

if((res = curl_easy_perform(ctx)) == 0)
{
    printf("OK\n");
    printf("%s\n", memData.memory);
    //body 출력
}

free(memData.memory);

curl_easy_cleanup(ctx);
curl_global_cleanup();
```