一、断点续传原理

所谓断点续传，也就是要从文件已经下载的地方开始继续下载。在以前版本的 HTTP 协议是不支持断点的，HTTP/1.1 开始就支持了。一般断点下载时才用到 Range 和 Content-Range 实体头。

1.1 不使用断点续传

|  |
| --- |
| get /down.zip http/1.1  accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-  excel, application/msword, application/vnd.ms-powerpoint, \*/\*  accept-language: zh-cn  accept-encoding: gzip, deflate  user-agent: mozilla/4.0 (compatible; msie 5.01; windows nt 5.0)  connection: keep-alive |

服务器收到请求后，按要求寻找请求的文件，提取文件的信息，然后返回给浏览器，返回信息如下：

|  |
| --- |
| HTTP/1.1 200 Ok  content-length=106786028  accept-ranges=bytes  date=mon, 30 apr 2001 12:56:11 gmt  etag=w/"02ca57e173c11:95b"  content-type=application/octet-stream  server=microsoft-iis/5.0  last-modified=mon, 30 apr 2001 12:56:11 gmt |

1.2 使用断点续传

|  |
| --- |
| GET /down.zip HTTP/1.0  User-Agent: NetFox  RANGE: bytes=2000070-  Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2 |

多了这么一行Range: bytes=2000070-

这一行的意思就是告诉服务器down.zip这个文件从2000070字节开始传，前面的字节不用传了。

Range的完整格式是：

|  |
| --- |
| Range: bytes=startOffset-targetOffset/sum [表示从startOffset读取，一直读取到targetOffset位置，读取总数为sum直接]  Range: bytes=startOffset-targetOffset [字节总数也可以去掉] |

服务器收到这个请求以后，返回的信息如下：

|  |
| --- |
| HTTP/1.1 206 Partial Content  content-length=106786028  content-range=bytes 2000070-106786027/106786028  date=mon, 30 apr 2001 12:55:20 gmt  etag=w/"02ca57e173c11:95b"  content-type=application/octet-stream  server=microsoft-iis/5.0  last-modified=mon, 30 apr 2001 12:55:20 gmt |

和前面服务器返回的信息比较一下，就会发现增加了一行：

|  |
| --- |
| Content-Range=bytes 2000070-106786027/106786028 |

返回的代码也改为206了，而不再是200了：

|  |
| --- |
| HTTP/1.1 206 Partial Content |

二、PHP实现

|  |
| --- |
| /\*\* php下载类,支持断点续传  \* download: 下载文件  \* setSpeed: 设置下载速度  \* getRange: 获取header中Range  \*/  class FileDownload{  /\*\* 下载  \* @param String $file 要下载的文件路径  \* @param String $name 文件名称,为空则与下载的文件名称一样  \* @param boolean $reload 是否开启断点续传  \*/  public function download($file, $name='', $reload=false){  $fp = @fopen($file, 'rb');  if($fp){  if($name==''){  $name = basename($file);  }  $header\_array = get\_headers($file, true);  //var\_dump($header\_array);die;  // 下载本地文件，获取文件大小  if (!$header\_array) {  $file\_size = filesize($file);  } else {  $file\_size = $header\_array['Content-Length'];  }  $ranges = $this->getRange($file\_size);  $ua = $\_SERVER["HTTP\_USER\_AGENT"];//判断是什么类型浏览器  header('cache-control:public');  header('content-type:application/octet-stream');  $encoded\_filename = urlencode($name);  $encoded\_filename = str\_replace("+", "%20", $encoded\_filename);  //解决下载文件名乱码  if (preg\_match("/MSIE/", $ua) || preg\_match("/Trident/", $ua) ){  header('Content-Disposition: attachment; filename="' .$encoded\_filename . '"');  } else if (preg\_match("/Firefox/", $ua)) {  header('Content-Disposition: attachment; filename\*="utf8\'\'' . $name . '"');  }else if (preg\_match("/Chrome/", $ua)) {  header('Content-Disposition: attachment; filename="' . $encoded\_filename . '"');  } else {  header('Content-Disposition: attachment; filename="' . $name . '"');  }  //header('Content-Disposition: attachment; filename="' . $name . '"');  if($reload && $ranges!=null){ // 使用续传  header('HTTP/1.1 206 Partial Content');  header('Accept-Ranges:bytes');  // 剩余长度  header(sprintf('content-length:%u',$ranges['end']-$ranges['start']));  // range信息  header(sprintf('content-range:bytes %s-%s/%s', $ranges['start'], $ranges['end'], $file\_size));  //file\_put\_contents('test.log',sprintf('content-length:%u',$ranges['end']-$ranges['start']),FILE\_APPEND);  // fp指针跳到断点位置  fseek($fp, sprintf('%u', $ranges['start']));  }else{  file\_put\_contents('test.log','2222',FILE\_APPEND);  header('HTTP/1.1 200 OK');  header('content-length:'.$file\_size);  }  while(!feof($fp)){  //echo fread($fp, round($this->\_speed\*1024,0));  //echo fread($fp, $file\_size);  echo fread($fp, 4096);  ob\_flush();  }  ($fp!=null) && fclose($fp);  }else{  return '';  }  }  /\*\* 设置下载速度  \* @param int $speed  \*/  public function setSpeed($speed){  if(is\_numeric($speed) && $speed>16 && $speed<4096){  $this->\_speed = $speed;  }  }  /\*\* 获取header range信息  \* @param int $file\_size 文件大小  \* @return Array  \*/  private function getRange($file\_size){  //file\_put\_contents('range.log', json\_encode($\_SERVER), FILE\_APPEND);  if(isset($\_SERVER['HTTP\_RANGE']) && !empty($\_SERVER['HTTP\_RANGE'])){  $range = $\_SERVER['HTTP\_RANGE'];  $range = preg\_replace('/[\s|,].\*/', '', $range);  $range = explode('-', substr($range, 6));  if(count($range)<2){  $range[1] = $file\_size;  }  $range = array\_combine(array('start','end'), $range);  if(empty($range['start'])){  $range['start'] = 0;  }  if(empty($range['end'])){  $range['end'] = $file\_size;  }  return $range;  }  return null;  }  }  $obj = new FileDownload();  $obj->download('http://down.golaravel.com/laravel/laravel-master.zip','', true); |