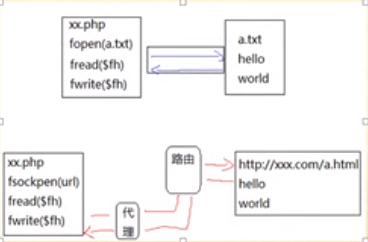
一、

PHP+socket编程 发送HTTP请求

要求能：模拟下载，注册，登录，批量发帖

1. 连接某URL的80端口（打开）
2. 发送头信息（写）
3. 读取网页内容（读）

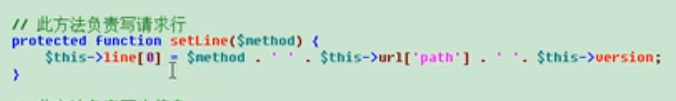
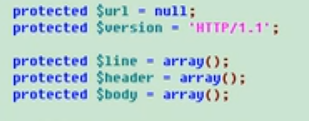


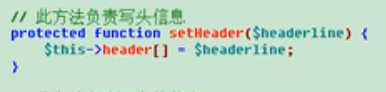
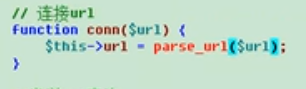
二、

1. 基础大框架

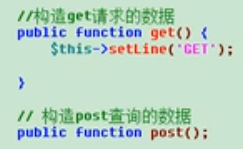
 

1. SetLine

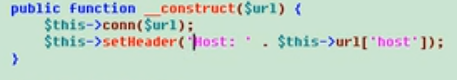


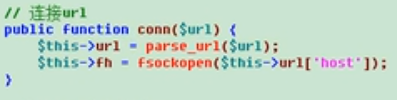
1. SetHeader
2. conn

（实例：）

1. Ways
2. 进行测试





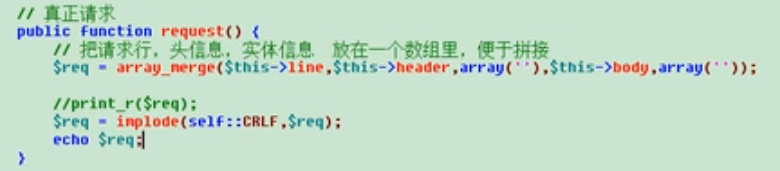


测试：

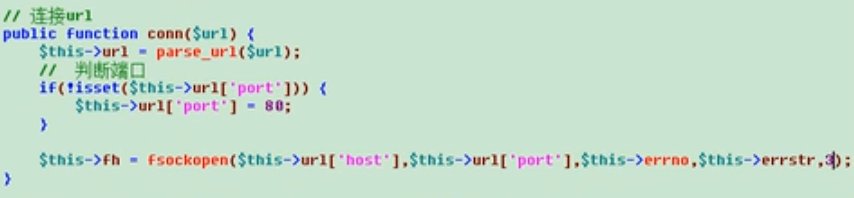


当时不同浏览器不同的对换行代码的标准，所以选择共同能用的，将换行符写死：















附：整体代码

//http请求类的接口

interface Proto {

    //连接url

    function conn ($url);

    //发送get查询

    function get();

    //发送post查询

    function post();

    //关闭连接

    function close();

}

class Http implements Proto {

    const CRLF = "\r\n";

    protected $errno = -1;

    protected $errstr = '';

    protected $response = '';

    protected $url = null;

    protected $version = 'HTTP/1.1';

    protected $fh = null;

    protected $line = array();

    protected $header = array();

    protected $body = array();

    public function \_\_construct($url) {

        $this->conn($url);

        $this->setHeader('Host: ' . $this->url['host']);

    }

    //此方法负责写请求行

    protected function setLine() {

        $this->line[0] = $method . ' ' . $this->url["path"] . ' '. $this->version;

    }

    //此方法负责写头信息

    protected function setHeader($headerline) {

        $this->header[] = $headerline;

    }

    //此方法负责写主体信息

    protected function setBody() {

    }

    //连接url

    public function conn ($url) {

        $this->url = parse\_url($url);

        //判断端口

        if(isset(this->url['port'])) {

            $this->url['port'] = 80;

        }

        $this->fh = fsockopen($this->url['host'], $this->url['port'], $this->errno, $this->errstr, 3);

    }

    //构造get请求的数据

    public function get() {

        $this->setLine('GET');

        $this->request();

        return $this->response;

    }

    //发送post查询

    public function post() {

    }

    //真正请求

    public protected function request() {

        //把请求行，头信息，实体信息，翻盖一个数组里，便于拼接

        $req = array\_merge($this->line, $this->header, array(''), $this->body, array(''))

        //print\_r($req);

        $req = implode(self::CRLF, $req);

        //echo $req;

        fwrite($this->fh, $req);

        while(!feof($this->fh)) {

            $this->response .= fread($this->fh),1024

        }

        $this->close(); //关闭连接

        return $this->response;

    }

    //关闭连接

    public function close() {

    }

}

/\*

$url = 'https://www.bilibili.com/video/BV1js411g7Fw?p=3';

$http = new Http($url);

echo $http->get();

\*/