[PHP Excel Reader](http://code.google.com/p/php-excel-reader/)

PHP Excel Reader – чтение информации из файлов XLS

<http://code.google.com/p/php-excel-reader/>

Создание объектов Reader

$data = new Spreadsheet\_Excel\_Reader("test.xls");

Для сохранения памяти для большего количества рабочих листов, не храня расширенную информацию о клетках как шрифты, цвета, и т.д.

$data = new Spreadsheet\_Excel\_Reader("test.xls",false);

Для использования кодирования кроме UTF-8 (по умолчанию), вы можете передать ее в качестве третьего параметра.

$data = new Spreadsheet\_Excel\_Reader("test.xls",true,"UTF-16");

Дампинг содержимого рабочего листа

Самый простой способ вывода содержимого из файла XLS это просто дамп его в HTML для отображения в браузере. Этот метод будет генерировать таблицу со встроенным CSS, со всем имеющимся форматированием.

$data->**dump**($row\_numbers=**false**,$col\_letters=**false**,$sheet=0,$table\_class='excel');

Вызов значения ячейки

Рекомендуется, чтобы общие функции API быть использованы для доступа к данным, а не полагались на базовую структуру данных, которые могут меняться между релизами.

Получить форматированное значение ячейки (то, что отображается в Excel) на первый (или единственный) лист:

$data->**val**($row,$col);

Вы можете также использовать имена столбцов, а не номера:

$data->**val**(10,'AZ');

Доступ к данным на другом листе:

$data->**val**($row,$col,$sheet\_index);

Информация о листе

Получить количество, сколько строк / столбцов находится на листе (по умолчанию: первый лист):

$data->**rowcount**($sheet\_index=0);

$data->**colcount**($sheet\_index=0);

Информация о ячейке

Тип данных в ячейке: number | date | unknown

$data->**type**($row,$col,$sheet=0);

Исходные данные сохраненные для ячейки. Например, ячейка может содержать 123.456, но показать как 123.5 из-за формата ячейки. Исходные данные, являются основным значением ячейки.

$data->**raw**($row,$col,$sheet=0);

Если ячейка имеет гиперссылку связанную с ней, то URL может быть восстановлен.

$data->**hyperlink**($row,$col,$sheet=0);

RowSpan / ColSpan ячейки.

$data->**rowspan**($row,$col,$sheet=0);

$data->**colspan**($row,$col,$sheet=0);

Форматирование

Метод извлекает всю имеющуюся информацию о форматировании и возвращает CSS строку со всеми атрибутами.

$data->**style**($row,$col,$sheet=0);

Формат строки для ячейки.

$data->**format**($row,$col,$sheet=0);

Выравнивание ячейки. Значения: left | center | right

$data->**align**($row,$col,$sheet=0);

Цвет фона, в формате # FFFFFF.

$data->**bgColor**($row,$col,$sheet=0);

Получить тип границы для любой из сторон ячейки. Возможные возвращаемые значения:

* Thin
* Medium
* Dashed
* Dotted
* Thick
* Double
* Hair
* Medium dashed
* Thin dash-dotted
* Medium dash-dotted
* Thin dash-dot-dotted
* Medium dash-dot-dotted
* Slanted medium dash-dotted

$data->**borderLeft**($row,$col,$sheet=0);

$data->**borderRight**($row,$col,$sheet=0);

$data->**borderTop**($row,$col,$sheet=0);

$data->**borderBottom**($row,$col,$sheet=0);

Получить цвета каждой из границ, в формате # FFFFFF.

$data->**borderLeftColor**($row,$col,$sheet=0);

$data->**borderRightColor**($row,$col,$sheet=0);

$data->**borderTopColor**($row,$col,$sheet=0);

$data->**borderBottomColor**($row,$col,$sheet=0);

Цвет шрифта, который может быть определен или свойствами ячейки или свойствами формата, в формате #FFFFFF.

$data->**color**($row,$col,$sheet=0);

Другие свойства шрифта:

$data->**bold**($row,$col,$sheet=0); // логическое

$data->**italic**($row,$col,$sheet=0); // логическое

$data->**underline**($row,$col,$sheet=0); // логическое

$data->**height**($row,$col,$sheet=0); // Количество в пикселях

$data->**font**($row,$col,$sheet=0); // Название шрифта

Дополнения

Просто добавьте метод ниже в класс.

/\*\*

\* Экспорт Excel файла в CSV формат

\* ---------------------------------------------------

\* Exports an excel file into .csv format

\* Written by Bimal ([http://bimal.org.np](http://bimal.org.np/))

\*/

function dump\_csv($row\_numbers=false, $col\_letters=false, $sheet=0, $table\_class='excel') {

$outs = array();

for($row=1; $row<=$this->rowcount($sheet); $row++) {

$outs\_inner = array();

for($col=1; $col<=$this->colcount($sheet); $col++) {

// Account for Rowspans/Colspans

$rowspan = $this->rowspan($row, $col, $sheet);

$colspan = $this->colspan($row, $col, $sheet);

for($i=0; $i<$rowspan; $i++) {

for($j=0; $j<$colspan; $j++) {

if ($i>0 || $j>0) {

$this->sheets[$sheet]['cellsInfo'][$row+$i][$col+$j]['dontprint']=1;  
        }

    }

}

if(!$this->sheets[$sheet]['cellsInfo'][$row][$col]['dontprint']) {

$val = $this->val($row, $col, $sheet);

$val = ($val=='')?'':addslashes(htmlentities($val));

$outs\_inner[] = "\"{$val}\""; # Quote or not?  
                                     #$outs\_inner[] = $val;

}

}

$outs[] = implode(',', $outs\_inner);

}

$out = implode("\r\n", $outs);

return($out);

}

/\*\*

\* Функция для получения column по номеру индекса

\*/

function letras($index)  
{  
        $letras = Array("A",   
                                        "B",   
                                        "C",   
                                        "D",   
                                        "E",   
                                        "F",   
                                        "G",   
                                        "H",   
                                        "I",   
                                        "J",   
                                        "K",   
                                        "L",   
                                        "M",   
                                        "N",   
                                        "O",   
                                        "P",   
                                        "Q",   
                                        "R",   
                                        "S",   
                                        "T",   
                                        "U",   
                                        "V",   
                                        "W",   
                                        "X",   
                                        "Y",   
                                        "Z");  
  
        $tam = count($letras);  
          
        if($index > $tam)  
        {  
                $div = ($index/$tam);  
                settype($div, "integer");  
                $sub = $tam\*$div;  
                $sub = ($index-$sub);  
                $div = $div-1;  
                $sub = $sub-1;  
                $exibe = $letras[$div].$letras[$sub];  
                  
                return $exibe;  
        }  
  
        return $letras[$index-1];  
}