

Proyecto final

Como proyecto final quiero presentar todo los ejercicios que fuimos desarrollando a lo largo del curso, en el menú PROYECTOS ->STOCK`S es lo último que estaba intentando hacer en el cual me queda algunas cosas que resolver, os comento este último en concreto porque sino la explicación sería extensa.

la idea es el uso de Custom Post Type <!--Panel de administración para crear tipos de contenido personalizados y taxonomías
crear en una de ellas un bloque personalizado en mi caso he utilizado Genesis Custom Blocks <!-- hace que sea fácil aprovechar esto y construir bloques personalizados.


Proceso de desarrollo :

instalamos los plugin necesarios, en este proyecto “ Custom Post Type” & “Genesis Custom Blocks”

Block Personalizado :

en bloques personalizados creamos un nuevo bloque con el nombre stocks, en la edición del nuevo bloque le creamos un campo de categoría Texto

stocks

Template: `wp-content/themes/jjnico/blocks/block-stocks.php` 

Editor Fields

Inspector Fields

 stock

stock 



La idea general del proyecto es que mediante bloques personalizados se pueda hacer consultas a un web-service personalizado en el cual decida de donde obtener los datos si de una API externa o de su misma base de datos.

al estar utilizando wordPress como base de todo el conjunto donde se encuentra la web no he querido utilizar la misma base de datos de wordPress para generar su propia base de datos ya que la información a trabajar podría ser extensa

el proceso es el siguiente :

Web-service :

recibe **\$_GET** un parámetro en este caso "symbol" crea las tablas necesarias en otra base de datos externa.

De esta forma intentó agilizar los manejos de información al momento de cargar la vista, lo segundo importante es crear una tabla de respaldo en el caso que la información a consultar esté desactualizada .

parte de su código ->

```
if (isset($_GET['symbol'])) {
    $symbol = $_GET['symbol'];

    // sql to create table
    $sql = "CREATE TABLE `nascor01_DB`.`symbol` (
        `id_symbol` INT NOT NULL AUTO_INCREMENT ,
        `symbol` VARCHAR(10) NOT NULL ,
        PRIMARY KEY (`id_symbol`)) ENGINE = InnoDB;";

    $sql = "CREATE TABLE `nascor01_DB`.`meta_data` (
        `id` INT NOT NULL AUTO_INCREMENT ,
        `information` VARCHAR(100) NOT NULL ,
        `symbol` VARCHAR(20) NOT NULL ,
        `last_refreshed` DATE NOT NULL ,
        `out_size` VARCHAR(20) NOT NULL ,
        `id_symbol` int(11) NOT NULL,
        `time_zone` VARCHAR(50) NOT NULL ,
        PRIMARY KEY (`id`)) ENGINE = InnoDB;";

    $sql = "CREATE TABLE
        `nascor01_DB`.`time_series_daily`
        ( `id` INT NOT NULL AUTO_INCREMENT ,
        `daily` DATE NOT NULL ,
        `open` FLOAT NOT NULL ,
        `high` FLOAT NOT NULL ,
        `low` FLOAT NOT NULL ,
        `close` FLOAT NOT NULL ,
        `volume` FLOAT NOT NULL ,
        `id_symbol` INT NOT NULL ,
        PRIMARY KEY (`id`)) ENGINE = InnoDB;";
```

Una vez creada las tablas pregunta de dónde obtener la información, en este caso al ser creada por primera vez hace uso de un API externa obteniendo un JSON y los separa en array para poder insertar su valor en las diferentes tablas anteriormente creadas.

El primer registro contiene un array con clave => valor obtener la información de este me resultó más sencilla que el segundo array del JSON.

```
}
else{
    //echo 'de lo contrario guardar';
    $sql = "INSERT INTO `symbol` (id_symbol, symbol) VALUES (NULL, '$symbol')";
    $registro = $mysqli->query($sql);
    //obtengo el ultimo id ingresado
    $last_id= $mysqli -> insert_id;
    $data = json_decode(file_get_contents("https://www.alphavantage.co/query?
    .on=TIME_SERIES_DAILY&symbol=$symbol&outputsize=full&apikey=0XQB6K7M2RYU0QE6"), true);

    $information=$data['Meta Data']['1. Information'];
    $symbol=$data['Meta Data']['2. Symbol'];
    $last_refreshed=$data['Meta Data']['3. Last Refreshed'];
    $out_size=$data['Meta Data']['4. Output Size'];
    $time_zone=$data['Meta Data']['5. Time Zone'];

    // insertamos el meta data con el ultimo id_symbol
    $sql="INSERT INTO `meta_data`
        (`id`, `information`, `symbol`, `last_refreshed`, `out_size`, `id_symbol`, `time_zone`)
        VALUES (NULL, '$information', '$symbol', '$last_refreshed', '$out_size', '$last_id', '$time_zone')";
    $mysqli->query($sql);
```

En el segundo array obtenemos todos los valores por vuelta de cada vueltas.

Los valores necesarios se extraen del foreach a una variable convirtiendo los datos a float Una vez terminado el foreach interno se insertan los datos en la tabla SQL y realiza el nuevo ciclo hasta su fin principal.

```
$registro_json = []; //creamos un array
foreach ($data['Time Series (Daily)'] as $clave => $array){
    $registro_json = []; //creamos un array
    foreach ($array as $key => $value){
        $registro_json[] = $value;
    }

    $open= json_encode(floatval($registro_json[0]));
    $high= json_encode(floatval($registro_json[1]));
    $low= json_encode(floatval($registro_json[2]));
    $close= json_encode(floatval($registro_json[3]));
    $volume= json_encode(floatval($registro_json[4]));

    $sql="INSERT INTO `time_series_daily`
    (`id`, `daily`, `open`, `high`, `low`, `close`, `volume`, `id_symbol`)
    VALUES (NULL, '$clave', '$open', '$high', '$low', '$close', '$volume', '$last_id');";
    $mysqli->query($sql);
    |
}
```

Generar la respuesta del web service

realiza una consulta a la tablas necesarias para la respuesta

creamos una variable \$result donde almacenaremos los datos obtenidos en el formato que necesitamos

```
..
} else {
    //consultamos que los datos existen
    $sql = "SELECT * FROM `symbol` WHERE `symbol` = '$symbol'";

    if($mysqli->query($sql)->num_rows > 0 ){// comprueba que existen datos
        //echo 'si hay datos mostrar todos()';
        $sql = "SELECT * FROM `symbol` WHERE `symbol` = '$symbol'";

        //obtenemos el id_symbol de la consulta
        $registro = $mysqli->query($sql);
        while ($fila = $registro->fetch_assoc()) {
            $id_symbol = $fila['id_symbol'];
            $symbol = $fila['symbol'];
        }

        // recuperamos datos de la tabla meta_data
        $sql = "SELECT * FROM `meta_data` WHERE `id_symbol` = '$id_symbol'";

        //obtenemos el id_symbol de la consulta
        $registro = $mysqli->query($sql);
        $result = '{"meta_data":{';
        while ($fila = $registro->fetch_assoc()) {
            // $id_symbol = $fila['id_symbol'];
            // $symbol = $fila['symbol'];
        }

        $result
        .="information":'".$fila['information'].','."symbol":'".$fila['symbol'].','."last_refreshed":'".$fila['last_refreshed'].','."out_size":'".$fila[
        'out_size'].','."time_zone":'".$fila['time_zone'].','."';
    }
}
```

realizamos una segunda consulta a otra tabla que necesitamos para componer el JSON y los datos obtenidos y lo concatenamos a la variable \$result para conformar un objeto devolución de la respuesta

header('Content-Type: application/json');

```
echo $result;
```

```
// Realizar una consulta SQL
$sql = "SELECT * FROM `time_series_daily` WHERE `id_symbol` = '$id_symbol'";
if (!$resultado = $mysqli->query($sql)) {
    // ¡Oh, no! La consulta falló.
    echo "Lo sentimos, este sitio web está experimentando problemas.";
    echo "Error: La ejecución de la consulta falló debido a: \n";
    echo "Query: " . $sql . "\n";
    echo "Errno: " . $mysqli->errno . "\n";
    echo "Error: " . $mysqli->error . "\n";
    exit;
}

$registro_json = []; //creamos un array para mostrar en pantalla

    $i=0;
$result .= '"time_series_daily":{' ;
    $registro = $mysqli->query($sql);
    while ($fila = $registro->fetch_assoc()) {
        //' $fila['id'];
        if($i>0){
            $result .= ',';
        }
        $i++;

        $result .= '': $fila['daily'].':':
{"id":'':$fila['id'].',', "open":'':$fila['open'].',', "high":'':$fila['high'].',', "low":'':$fila['low'].',', "close":'':$fila['close'].',', "volume":'':
'.'$fila['volume'].',', "id_symbol":'':$fila['id_symbol'].',', '}}';

    }
    $result .= '}}';

    header('Content-Type: application/json');
    echo $result;
```

para comprobar su funcionamiento del web service

<https://nascor01.md360.es/web-service/web-service/webService.php?symbol=ocgn>

resumiendo..

Utilizando un themes personalizado en este caso jnico añadido tres carpetas principales

llamadas blocks , blocks-models y blocks-views

la primera carpeta blocks contiene todos los bloques creados

el archivo **block-stocks.php**

```
<?php
:
: /*incluimos la url del modelo solo una vez para genenerar una nueva instancia por cada vez que se llame al bloque*/
: include_once get_template_directory() . '/blocks-models/timeSeriesIntradayModel.php';
: /*creamos una nueva instancia y le pasamos un parametro que retornara un valor que será almacenado en la variable rows*/
: $intraday = new Intraday();
: /*llamo a la funcion que se encuentra dentro de la clase generada por la instancia y le paso un string recibido */
: $rows = $intraday->timeSeriesIntraday(block_field('stock',false));
: /*convierto el string a objeto que es obtenido de la consulta al web-service para poder trabajarlo mejor*/
: $object = json_decode($rows);
:
: /*genero las variables del primero array que se encuentra dentro del objeto para un mejor manejo de la informacion y programación */
: $information = $object->meta_data->information;
: $symbol = $object->meta_data->symbol;
: $last_refreshed = $object->meta_data->last_refreshed;
: $out_size = $object->meta_data->out_size;
: $time_zone = $object->meta_data->time_zone;
:
: /*creo un foreach del arreglo que se encuentra dentro del segundo array del objeto principal*/
: /*obtengo un array del segundo array del objeto principal*/
: foreach($object->time_series_daily as $clave =>$array){
:     /*obtengo el arreglo y por cada interacción le asigno el valor de cada key a su variable correspondiente */
:     foreach($array as $key => $valor){
:         $daily[] = $clave; // obtengo la clave por cada vez que se ejecuta el primer foreach por vuelta
:         $key_id[] = $array->id;
:         $open[] = $array->open;
:         $high[] = $array->high;
:         $low[] = $array->low;
:         $close[] = $array->close;
:         $volume[] = $array->volume;
:     }
: }
:
: /*llamamos a la vista correspondiente para mostrar los valores */
: include get_template_directory() . '/blocks-views/metaDataCardStockView.php';
:
: ?>
```

en la carpeta **blocks-models** el archivo **timeSeriesIntradayModel.php**

```
class Intraday{
    private $db;
    public $rows;

    // esta funcion se encarga de llamar al web service pasandole un parametro y esta devolvera un array
    public function timeSeriesIntraday($symbol){

        $url ="https://nascor01.md360.es/web-service/web-service/webService.php?symbol=$symbol";

        return file_get_contents($url);

        /*
        return $data = json_decode(file_get_contents("https://www.alphavantage.co/query?
function=TIME_SERIES_DAILY&symbol=$symbol&outputsize=full&apikey=0XQB6K7M2RYU0QE6"), true);*/
    }
}
```

en la carpeta **blocks-views** el archivo **metaDataCardStockView.php**

```

<!--
// todos los parametros que puede recibir por variable
// $information
// $symbol
// $last_refreshed
// $out_size
// $time_zone
// $key_id[]
// $daily[]
// $open[]
// $high[]
// $low[]
// $close[]
// $volume[]
-->

<?php
date_default_timezone_set($time_zone);
$i = 0;
$maxValueOpen = 0;
$minValueOpen = 0;
?>
<div class="card">
    <h1 class="entry-header"><?php echo strtoupper($symbol);?></h1>
    <div class="container">
        <h2 class="entry-header"><?php echo $symbol;?></h2>
        <span id="daily"> <?php echo 'daily';?></span> <?php echo $daily[$i];?><br>
        <span><?php echo 'open';?></span> <?php echo $open[$i];?><br>
        <span>high</span> <?php echo $high[$i];?><br>
        <span>low</span> <?php echo $low[$i];?><br>
        <span>close</span> <?php echo $close[$i];?><br>
        <span>volume</span> <?php echo $volume[$i];?><br>
        <div class="wp-block-button">
            <a class="wp-block-button__link" href="https://nascor01.md360.es/wordpress/web-service/web-service/";
<?php echo $last_refreshed;?>"><?date('m/d/y');?></a>
        </div>
    <?php
    if ($open[$i] > $maxValueOpen){
        $maxValueOpen = $open[$i];
    }

    if ($low[$i] > $minValueOpen){
        $minValueOpen = $low[$i];
    }
    ?>
    <label for="file">Posición a fecha : </label><?php echo $last_refreshed;?><br>
    <label for="open">Status bar :</label>
    <progress id="open" value="<?php echo $minValueOpen;?>" max="<?php echo $maxValueOpen; ?>" 32% </progress><br>
</div>

</div>

```

ver resultado a falta de darle mejor estilos css y distribución de la información

<https://nascor01.md360.es/wordpress/index.php/stocks/stocks/>