TABLA v.1.2

Julio Alejandro Santos Corona

14 de noviembre de 2020

La clase provee los elementos principales para la creación de una tabla a través de una configuración inicial y la definición de comportamiento.

1. Instancia

La instancia se crea a través del constructor de la función **JS_TABLA.main** al que se le pasan 3 argumentos, width, height, id (default: js_tabla) del div a utilizar.

```
1  var tabla = new JS_TABLA.main ("100%", "400px");
```

1.1. Configuración

La configuración de la tabla se realiza a través del método **setConfig** que solicita un argumento tipo object con la siguiente estructura:

Parámetro	Tipo	Descripción
colnameAlign	boolean	En caso de ser activada, la alineación de los títulos será la misma que la del contenido de cada celda en la columna.
limitLines	boolean	Activa la limitación de filas en la tabla, el número límite se define con el método **limit**.
pagination	boolean	Muestra los controles de paginación.
callbackScroll	boolean	Activa la posibilidad de ejecutar una petición al llegar al final del scroll de filas (útil para tablas que muestran información particionada).
columns	object	Configuración de las columnas.

Columns requiere los objetos por columna con la siguiente estructura:

Parámetro	Tipo	Descripción
name	string	Nombre de la columna.
size	string	Tamaño de la columna; admite los valores aceptados de **width**, CSS.
align	string	Valores admitidos por text-align.
id	boolean	Registra el valor de la celda como id para la fila. (Oculta por default).
argument	boolean	Guarda el valor de la celda en un objeto que será aprovechado por el callback del método **lines**. (Oculta por default).
visible	boolean	Muestra la columna configurada como id o argumento.
edit	boolean	Permite modificar el valor de la celda.

```
1  var config = {
2      colnameAlign: true,
3      limitLines: true,
4      pagination: false,
5      callbackScroll, true,
6      columns: {
7          0: { name: "id", id: true },
8          1: { name: "nombre", size: "10em" },
9          2: { name: "edad", size: "2em", align: "center" },
10         3: { name: "salario", size: "5em", align: "right", edit: true },
11         4: { name: "estado", size: "3em", argument: true, visible: true }
12     }
13 };
```

2. Crear filas

Las filas pueden provenir de una consulta a base de datos pues la estructura arrojada por la misma cumple sin problema con la estructura siguiente:

```
{
    0: {
        0: "val1",
        1: "val2",
        2: "val3",
        .
        .
        .
    },

    1: {
        0: "val1",
        1: "val2",
        2: "val3",
        .
        .
        .
    },
    .
    .
    .
}
```

para imprimir las filas bastará con utilizar el método **lines**, dicho método solicita 3 argumentos que a continuación se describen:

Argumento	Tipo	Opcional	Descripción
data	object	No	Valores a rellenar, el objeto debe contener la estructura antes descrita.
callback	function	Si	Función que se ejecutará por cada fila creada, recibe 2 argumentos, **line** y **argments**.
add	boolean	Si	**False**(default): La tabla se limpia antes de insertar los valores de data. **True**: Los valores son agregados inmediatamente abajo de los existentes.

Ejemplo:

```
var tabla = new JS_TABLA.main ("100%", "400px");
let consulta;

tabla.setConfig (config); //Suponiendo que config ya fue asignado

consulta = new Promise (/*Peticion ajax*/);

consulta.then (function (data) {
    tabla.lines (data);
});
```

2.1. Utilizando callback en lines

El método **lines** tiene la posibilidad de ejecutar una función por cada fila a crear y recibe dos argumentos:

Argumento	Tipo	Descripción
line	HTMLElement	Objeto HTMLELement que contiene todas las propiedades de la fila.
argments	object	En caso de haber configurado columnas como argument este objeto contendrá los valores con el nombre configurado en **name**.

Ejemplo:

```
let consulta = new Promise (/*Peticion ajax*/);

/*
 * Este callback valida a traves de un argumento el estatus y coloca un
 * estilo para toda la fila.
*/
consulta.then (function (data) {
    tabla.lines (data, function (line, argments) {
        //Codigo
        if (argments.status == 2) {
            line.style.background = "rgb(250, 0, 0)";
        }
    });
});
```

3. Eventos

La clase cuenta con 3 eventos disponibles **onEdit**, **onScroll** y **onPagination**, cada uno requiere previa configuración y solicitan una función como argumento.

<table>
<tr><th>Método</th><th>Requerimiento</th><th>Descripción</th></tr>
<tr><td>onEdit</td><td>edit</td><td>La función se ejecuta al cambiar el valor de una celda configurada como editable. La función no se ejecutará si se oprime la tecla Escape o si el valor editado resulta igual al original. Adicionalmente el callback recibe dos argumentos HTMLElement la línea y la celda modificada.</td></tr>
<tr><td>onScroll</td><td>callbackScroll</td><td>Su ejecución se realizará al llegar al final del scroll. Funcionalidad útil si se requiere cargar datos extra.</td></tr>
<tr><td>onPagination</td><td>pagination</td><td>La función se ejecutar al intentar paginar la tabla, dicha función recibe un objeto como argumento:
<table>
<tr><th>Argumento</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td>start</td><td>number</td><td>Inicio del rango a obtener en la paginación.</td></tr>
<tr><td>end</td><td>number</td><td>Final del rango a obtener en la paginación.</td></tr>
<tr><td>page</td><td>number</td><td>Página siguiente luego de la paginación.</td></tr>
<tr><td>update</td><td>function</td><td>Método que actualiza el número de página.</td></tr>
</table>
</td></tr>
</table>

4. Métodos extra

- **limit**: Configura la cantidad de filas a mostrar; sólo aplica en la configuración general y en la paginación; requiere un argumento number mayor a 0.
- **getCols**: Obtiene la cantidad de columnas configuradas.
- **getRows**: Obtiene la cantidad de filas mostradas.
- **getLimit**: Obtiene el límite de filas configurado.
- **deleteLine**: Elimina una fila, solicita un argumento que puede ser el HTMLElement del objeto a eliminar o el id de la fila.
- **clear**: Limpia las filas mostradas.
- **reset**: Elimina la tabla completa, columnas y filas.