

TABLA v.1.8

Julio Alejandro Santos Corona

02 de mayo de 2021

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.
editable	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: "estado", size: "3em", argument: true, visible: true },
11    4: { name: "salario", size: "5em", align: "right", editable: 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:

```
1  {
2    0: {
3      0: "val1",
4      1: "val2",
5      2: "val3",
6      .
7      .
8      .
9    },
10
11   1: {
12     0: "val1",
13     1: "val2",
14     2: "val3",
15     .
16     .
17     .
18   },
19   .
20   .
21   .
22 }
```

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 arguments .
include	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:

```
1  var tabla = new JS_TABLA.main ("100%", "400px");
2  let consulta;
3
4  tabla.setConfig (config); //Suponiendo que config ya fue asignado
5
6  consulta = new Promise (/*Petición ajax*/);
7
8  consulta.then (function (data) {
9    tabla.lines (data);
10 });
```

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.
arguments	object	En caso de haber configurado columnas como argument este objeto contendrá los valores con el nombre configurado en name .

adicional las líneas son construidas incluyendo la propiedad **tblProperties** con los argumentos de la fila.

Ejemplo:

```

1 let consulta = new Promise (/*Peticion ajax*/);
2
3 /*
4  * Este callback valida a traves de un argumento el estatus y coloca un
5  * estilo para toda la fila.
6  */
7 consulta.then (function (data) {
8     tabla.lines (data, function (line , arguments) {
9         //Codigo
10         if (arguments.status == 2) {
11             line.style.background = "rgb(250, 0, 0)";
12         }
13     });
14 });

```

3. Eventos

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

Método	Requerimiento	Descripción		
onEdit	edit	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 tres argumentos la línea, la celda modificada (ambos HTMLElement) y el valor anterior.		
onScroll	callbackScroll	Su ejecución se realizará al llegar al final del scroll. Funcionalidad útil si se requiere cargar datos extra.		
onPagination	pagination	La función se ejecuta al intentar paginar la tabla y se inactiva para evitar problemas, utilice commit o rollback para activarla; dicha función recibe un objeto como argumento:		
		Argumento	Tipo	Descripción
		currentPage	number	Página actual.
		nextPage	number	Página siguiente luego de la paginación.
		start	number	Inicio del rango a obtener en la paginación.
		end	number	Final del rango a obtener en la paginación.
		rollback	function	Método que activa la paginación sin actualizar página.
commit	function	Método que actualiza el número de página.		

4. Métodos extra

- **setLimit**: Configura la cantidad de filas a mostrar; sólo aplica en la configuración general y en la paginación; requiere un argumento number mayor a 0.
- **getLimit**: Obtiene el límite de filas configurado.
- **getCols**: Obtiene la cantidad de columnas configuradas.
- **getRows**: Obtiene la cantidad de filas mostradas.
- **setDeleteTime**: Configura el tiempo de duración para la eliminación de una fila.
- **deleteLine**: Elimina una fila, solicita un argumento que puede ser el HTMLElement del objeto a eliminar o el id de la fila, es posible pasar un callback que será ejecutado luego de la eliminación de la fila.
- **clear**: Limpia las filas mostradas.

- **reset:** Elimina la tabla completa, columnas y filas.