Giovanni Urdaneta

28288477

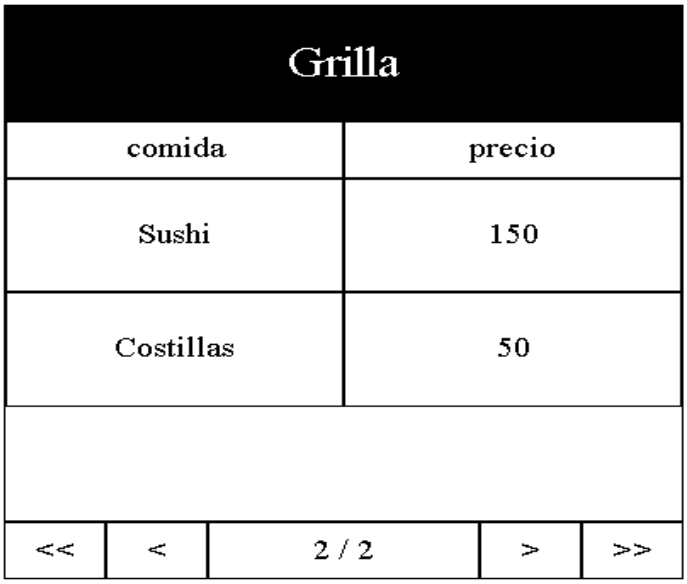
Grid

El web component consiste en una grilla de datos. En la cual ingresaremos arrays de objetos JSON y seran automaticamente desplegados en la misma (tambien es posible cargar archivos JSON en la misma.

El componente cuenta con un sistema de paginacion. Se establece por defecto (o por decision del desarrollador) un conjunto de columnas y un numero maximo de filas por pagina, y en base a esta restriccion se agregan las celdas a paginas secuencialmente, y se puede navegar a traves de ellas usando codigo o una barra de control opcional.

Grilla	
comida	precio
Hot dog	70
Hamburguesa	100
Helado	25

<<	<	1 / 2	>	>>

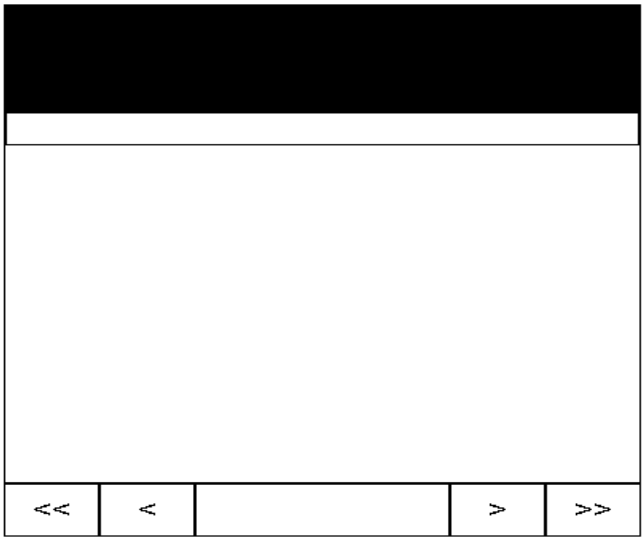


Inicializacion

La clase Grid pasa como parametro un objeto JSON, el cual contiene todas las propiedades del componente:

const grid = new Grid({ title: “Grilla”, rows: 5, columns: [“comida”, “precio”], … });

Si el componente no se inicializa en Javascript, sino en HTML (con la etiqueta data-grid) o se inicializa en Javascript sin parametros, se aplicaran propiedades por defecto que ya contiene el componente.



En el estado actual del web component, podria concluirse en que las propiedades son su “estado definitivo”, si se cambian despues de haber sido conectado al DOM, los efectos no se veran reflejados en el grid.

Esto implica que, si una propiedad no es especificada al instanciar el objeto, el grid mantendra el valor por defecto de dicha propiedad permanentemente. De la misma forma, las propiedades que definamos al instanciar dicho objeto tambien seran permanentes.

Insercion de datos

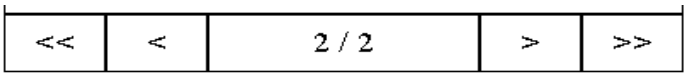
Los datos tienen un formato determinado: array de arrays. Cada array representa una fila, por lo cual, los valores deben estar organizados en el mismo orden que las columnas introducidas en las propiedades.

grid.addRows([[“Hamburguesa”, “100”], [“sushi”, “150”], …]);

Esta funcion es la manera mas simple de añadir datos al grid: pasando como parametro un array de arrays. Sin embargo, tambien se pueden agregar desde un archivo JSON local. En la seccion de metodos se analiza a detalle.

Barra de controles

Se trata de una interfaz que contiene el component para poder hacer paginacion.



Empezando por el marco central, este muestra la pagina del grid en la que el usuario se encuentra, al lado del total de paginas. El resto de los elementos de la barra son botones.

Haciendo una seguidilla de los botones de izquierda a derecha:

- Boton 1: primera pagina del grid.
- Boton 2: pagina anterior.
- Boton 3: siguiente pagina.
- Boton 4: ultima pagina.

Propiedades

- *title*: (cadena) es la cadena que se despliega como titulo del grid. Ubicado en la parte superior.

- *showBar*: (booleano) si su valor es true, se desplegar la barra de controles en la parte inferior de la grilla.

- *showItem*: (booleano) si su valor es true, se desplegara como primera columna un contador de cada celda.

- *rows*: (numero entero) cantidad de filas que puede tener cada pagina (maximo).

- *columns*: (array de cadenas) titulos de las columnas del grid.

- *size*: (numero real) tamaño del grid en numero, que representa unidades px en CSS.

- *titleBackColor*: color del background del titulo.

- *titleFontColor*: color de letra del titulo.

- *titleFontSize*: tamaño de letra del titulo.
- *cellBackColor*: color del background de cada celda.
- *cellFontColor*: color de letra de cada celda.
- *cellFontSize*: tamaño de letra de cada celda.
- *barBackColor*: color del background de la barra de controles.
- *barFontColor*: color de letra de la barra de controles.
- *barFontSize*: tamaño de letra de cada celda.
- *borderColor*: color del borde del grid.
- *borderSize*: tamaño del borde.

Nota: las propiedades de colores soportan cualquier tipo de formato de color de CSS. Las propiedades de tamaño soportan numeros que seran pasados como unidades de px en CSS.

Metodos

- *addRows(data):* inserta datos en el grid. El parametro data consiste en un array de arrays, donde cada uno de estos representa una fila del grid. Los valores deben estar organizados de acuerdo a lo que fue introducido en las columnas.
- *loadJSON(path):* carga un archivo JSON (cuyo formato debe ser un array de arrays) al grid. Es menester destacar que se trata de una peticion desde el metodo fetch; es decir, una operación asincronica, por lo cual, es necesario conectarse a un servidor para poder lograr esto.

Los siguientes metodos de paginacion ya estan automatizados con los botones de la barra de controles. Sin embargo, tambien pueden usarse en codigo y tendran el mismo efecto:

- *firstPage():* se posiciona en la primera pagina.

- *previousPage():* se posiciona en la pagina anterior.
- *nextPage():* se posiciona en la siguiente pagina.
- *lastPage():* se posiciona en la ultima pagina.

Valores por defecto

Como se menciono anteriormente, las propiedades que no son definidas adquieren automaticamente un valor por defecto, tanto las de estilo como las del formato del componente.

- *title* = “”
- *showBar* = true
- *showItem* = false
- *rows* = 5
- *columns*: [“”]

  Lo anterior implica que, al no especificar las columnas, la grilla se adaptara a una sola columna y a 5 filas maximo.

- *size* = 400
- *titleBackColor* = “black”
- *titleFontColor* = “white”
- *titleFontSize* = size * 0.05 + (size * 0.05 * 0.5)
- *cellBackColor* = “white”
- *cellFontColor* = “black”
- *cellFontSize* = size * 0.05

- *barBackColor* = “white”
- *barFontColor* = “black”
- *barFontSize* = size * 0.05
- *borderColor* = “black”
- *borderSize* = 1