



Sortarea Inteligentă

Asigurarea Integrității Datelor în Componentele DataGrid din React

Un concept din seminarul "Componente Externe în React"

Puterea Nativă a Componentei DataGrid

Componentele DataGrid moderne oferă sortare nativă, direct din interfață. Un singur click pe antetul coloanei este suficient pentru a reordona datele.

- Funcționalitate *out-of-the-box*, fără cod suplimentar.
- Primul click activează sortarea ascendentă.
- Al doilea click inversează ordinea (descendentă).

Titlu	Autor	Pagini
Dune	Frank Herbert	540
Fundația	Isaac Asimov	220

Fundația	Isaac Asimov	220
Dune	Frank Herbert	540

Ceva Nu Este în Regulă...

Atunci când încercăm să sortăm o coloană cu valori numerice, precum "Pagini", rezultatul este logic incorect.

De ce '100' apare înaintea lui '20'?

Pagini ✕
100
20
5
540

Confuzia Fundamentală: Text vs. Număr

Implicit, motorul de sortare tratează toate datele ca text. Acesta compară valorile **caracter cu caracter**, nu valoarea lor numerică reală. Vede '1' și decide că este înaintea lui '2', ignorând restul numărului.



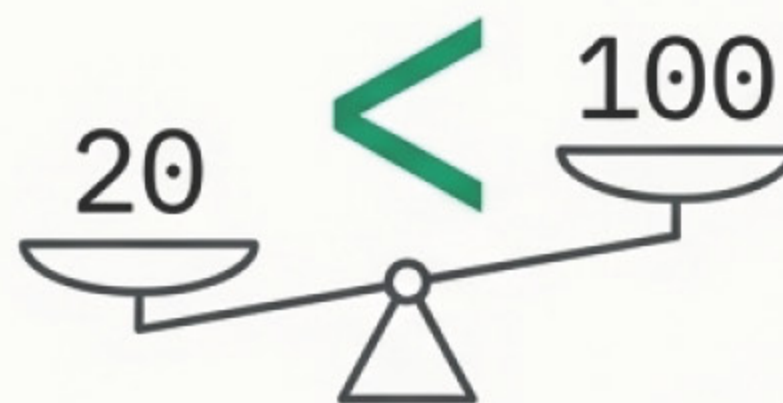
Sortare **Lexicografică** (Implicită)

'100' < '20'

'100' < '20' (deoarece '1' < '2')



Sortare **Semantică** (Corectă)



20 < 100

Soluția: O Singură Linie de Cod

Pentru a rezolva această eroare logică, trebuie să instruim explicit componenta DataGrid cu privire la **tipul de date** din coloană. Acest lucru este **obligatoriu** pentru a asigura o comparație numerică.

```
1  const columns = [  
2    // ...alte coloane  
3    {  
4      field: 'pages',  
5      headerName: 'Pagini',  
6      type: 'number' // Instruiește DataGrid să folosească un comparator numeric  
7    }  
8  ];
```

De la Haos la Claritate

Prin adăugarea proprietății **type**, algoritmul de sortare devine numeric. Datele sunt acum afișate într-o ordine logică, precisă și semantic corectă.

Pagini



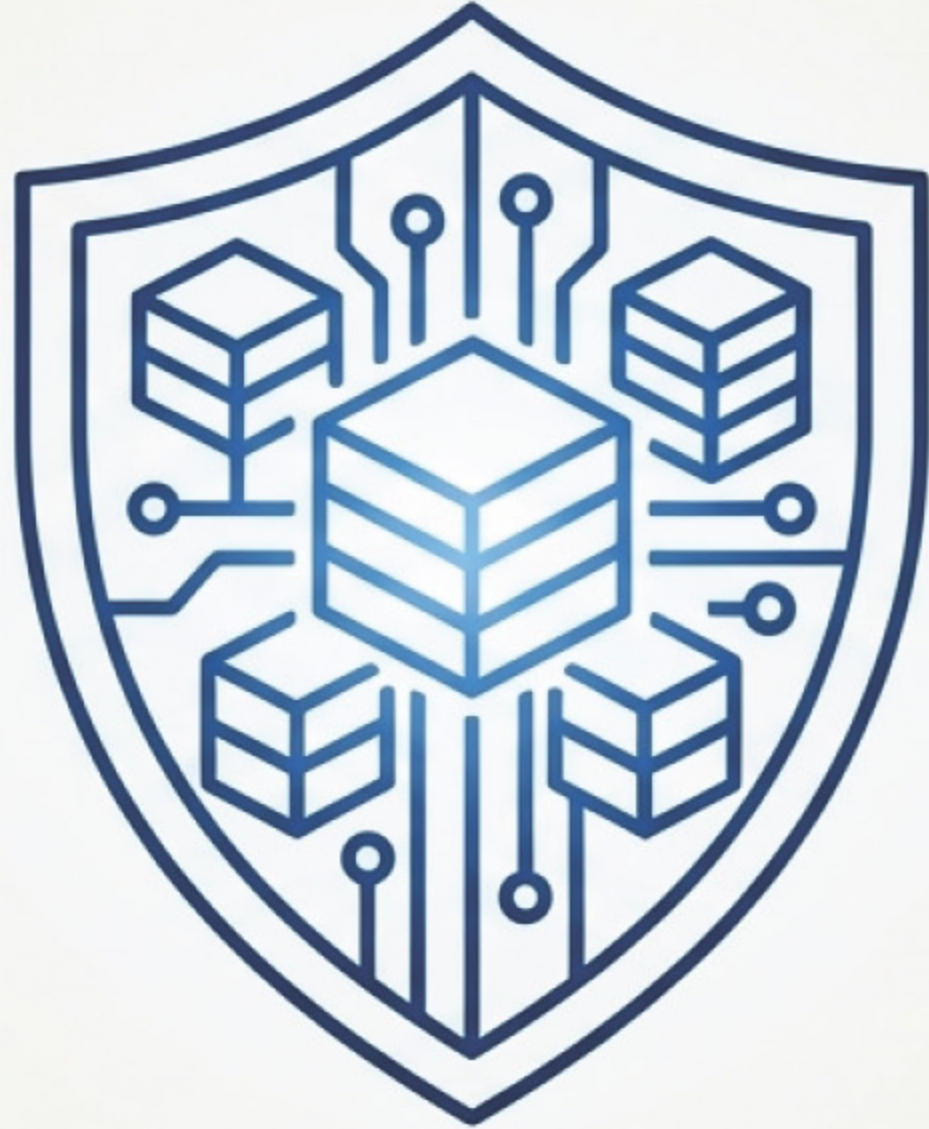
5

20

100

540





Principiul: Mai Mult Decât o Interfață

Aceasta nu este doar o ajustare vizuală, ci un pas esențial pentru a garanta **integritatea datelor**. O sortare corectă transformă un simplu tabel de afișare într-un

Instrument Analitic Funcțional și de încredere pentru utilizator.

Explorarea Opțiunilor Avansate

Pentru nevoi complexe, API-ul DataGrid permite personalizarea avansată a comportamentului de sortare.



Controlul Ciclului de Sortare

```
sortingOrder={['desc', 'asc', null]}
```



Logică Personalizată

```
sortComparator: customFunction
```

Recapitulare în 3 Puncte Esențiale

1

PROBLEMA

Sortarea implicită este **lexicografică** și produce erori logice pentru coloanele numerice sau de date calendaristice.

2

SOLUȚIA

Adaugă proprietatea `type: 'number'` (sau ``date``, ``dateTime``) la definiția coloanei pentru a activa sortarea semantică.

3

IMPACTUL

Garantează **integritatea datelor**, oferă o experiență corectă utilizatorului și transformă tabelul într-un instrument de încredere.

Resurse și Documentație

Aprofundați funcționalitățile de sortare și filtrare explorând documentația oficială a componentei.



Link către Documentația Oficială - Sorting

mui.com/x/react-data-grid/sorting/



Exemplu de Cod Sursă

github.com/your-repo/semantic-sort-example