

## Data Binding

### 2.1 Fetching Data from an API:

- Use fetch or a library like Axios to retrieve data from an external API.
- Store the fetched data in the component's state.
- Pass the data to the DataTable component's value prop.

#### Example:

JavaScript

```
import React, { useState, useEffect } from 'react';
import { DataTable } from 'primereact/datatable';

function ProductTable() {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    fetch('https://api.example.com/products')
      .then(response => response.json())
      .then(data => setProducts(data));
  }, []);

  return (
    <DataTable value={products} columns={['id', 'name', 'price']} />
  );
}
```

### 2.2 Updating Data Dynamically:

- Modify the data in the component's state.
- The DataTable will automatically re-render with the updated data.

#### Example:

JavaScript

```
function ProductTable() {
  const [products, setProducts] = useState([
    // ... initial products
  ]);
}
```

```

const handleAddProduct = (newProduct) => {
  setProducts([...products, newProduct]);
};

// ...
}

```

## 2.3 Implementing Pagination:

- Use the paginator prop of the DataTable component.
- Configure the totalRecords prop to specify the total number of records.
- Handle page change events with the onPage prop.

### Example:

```

JavaScript
function ProductTable() {
  const [products, setProducts] = useState([]);
  const [totalRecords, setTotalRecords] = useState(0);
  const [currentPage, setCurrentPage] = useState(0);

  // ... fetch data and set totalRecords

  return (
    <DataTable
      value={products}
      paginator={true}
      rows={10} // Number of rows per page
      totalRecords={totalRecords}
      first={currentPage * 10} // Adjust first record based on
current page
      onPage={(e) => setCurrentPage(e.first / 10)}
      // ... other props
    />
  );
}

```