Angular
Authentication
Cookies
Core
Forms
Government
i18n
Inform
Layout
Store
Table
Introduction
Installation
Implementation
Simple example
Detail row
Context row
Loading and empty state
Sorting, selecting and row-actions
Header
Custom cells
Defaults and NgxTableConfig
Styling
Accessibility
Acknowledgements
Tour
Utils
Express
Javascript
Pagination
Regex
RxJS
Testing

Context row

`ngx-table` provides a built-in way to add context rows to your table rows. This can be done by providing the `#contextRowTmpl` template as a `ContentChild` of the `NgxTable`.

Example use case

Since the use of this component is not easily described, below is an example of when a context row could be useful. Let's say you want to visualize if and when a certain data entry has been favorited, the context row can be useful. *Continue reading below for the (property) explanation.*

Name	FirstName
★ *Favorite since 2023-01-01*	
World	Hello
Tools	NGX
★ *Favorite since 2022-05-15*	
Angular	Framework

Explanation

Note: It is very important to be aware that this row does as its name suggests: it allows for additional context. It should not be used to add or define data of the row itself. This context row always has the `role="none"` aria tag. This means any and all content defined in this row gets excluded from the accessibility tree. Elements that are defined within this row that have a tabindex greater than or equal to 0 (and are therefor focusable) *are* included in the accessibility tree. This means that you may include buttons and links inside this row. For more information and what not to define in this context row, please visit the official ARIA docs: https://w3c.github.io/aria/#none.

Once a context row is provided, the row does not automatically become visible. We need to use an Input to handle the behavior of the table. By defining the `contextRowKeys` input to a non-empty array, the attributes of the `data` are checked for a possible match. If at least one key of the data the row consists of matches a value of the `contextRowKeys`, that row will get a context row. If no key matches, the row will still get a context row, but the class will be `ngx-context-row`. This is the same way that detail views are rendered when they are not expanded.

From the moment that the `#contextRowTmpl` has been defined, every main row will get a corresponding context row. It is set to `display: none` as long as it is empty, and it shall remain empty as long as the above condition is not met. Be cautious of applying styles to the `ngx-context-row` class.

- Angular
  - Authentication
  - Cookies
  - Core
  - Forms
  - Government
  - i18n
  - Inform
  - Layout
  - Store
  - Table
    - Introduction
    - Installation
    - Implementation
      - Simple example
      - Detail row
      - Context row
      - Loading and empty state
      - Sorting, selecting and row-actions
      - Header
      - Custom cells
      - Defaults and NgxTableConfig
      - Styling
      - Accessibility
    - Acknowledgements
  - Tour
  - Utils
- Express
- Javascript
  - Pagination
  - Regex
  - RxJS
  - Testing

{ "name": "World", "firstName": "Hello", "active": true }

World	Hello
Tools	NGX

Be cautious of applying styles to the `ngx-context-row` class, since hidden context rows also have this class. You can modify how all visible context rows look by applying styles to the `ngx-table-context-row-open` class. In the above example, only one key was provided to the table via the `contextRowKeys`. This resulted in the row with the `active: true` to also be assigned the `ngx-table-context-key--active` modifier class.

In the example below, the `name` key of the data also gets taken into account. This results in more than one modifier to be applied to the context row that also has the `active` property set to `true`. The order in which the classes are applied to the element, is the same order in which the array values are provided to the `contextRowKeys` input.

Note the classes in the example below using the DevTools. The first row should get the following classes (aside from the by-Angular-injected classes):

1. `cdk-row`
2. `ngx-table-context-row`
3. `ngx-table-context-row-open`
4. `ngx-table-row-first`
5. `ngx-table-context-key--name`
6. `ngx-table-context-key--active`

Name	FirstName
{ "name": "World", "firstName": "Hello", "active": true }	
World	Hello
{ "name": "Tools", "firstName": "NGX", "active": false }	
Tools	NGX

← Previous
Detail row

Next →
Loading and empty state