CustomTabOrder

Custom Tab Order Within Trinidad

Overview

It has been requested by more than one person to have the ability to support custom tab orders within Trinidad. There are different ways to approach this, so I am creating a WIKI page to open the field for discussion

General Problem

The idea to support a custom tab order seems simple enough by just adding a `tabIndex` attribute onto the components where it makes sense to have it. The problem that was brought up by Trinidad developers was that it could lead to a mess of maintenance by possibly requiring that the user set a `tabIndex` value on every component so that the correct order is used. This would cause a lot of work to have to be done on large pages.

Different Approaches

tabIndex on specific components

The simplest implementation is to add a `tabIndex` attribute onto components that it would make sense (like `tr:commandLink`.

Pros:

- Simple
- Relatively easy to develop
- Familiar to HTML developers
- Would support EL for backing-bean customized numbers

Cons:

- Would have to be set on many components on a page if set on one
- Without an IDE, inserting a new component would possibly force a cascade of adjusting the numbers on the page
- Each renderer would have to be adjusted
- Requires a lot of custom EL work to make it nice for large applications
- Prone to user errors

Rating:

{*} {*} 💡 {o} 💡

ResponseWriter added attribute

With a ResponseWriter approach, `tabIndex` could once again be added to certain components. Code in the Trinidad ResponseWriters could look for a call to `startElement(String, UIComponent` and force the addition of a `tabindex` attribute onto the element if the component has a `tabIndex` PropertyKey.

Pros:

- Same as above
- Would not have to change the renderer

Cons:

- Would break non-HTML response writers
- Same as above

Rating:

{*} 💡 {o} 💡 {o}

Custom Parent Component - 1

Easier to show than tell:

```
<tr:tabIndex type="auto" id="ti1">
  <tr:commandLink id="a" />
  <tr:tabIndex type="manual" order="ti4 ti3" id="ti2">
    <tr:tabIndex type="manual" order="d c b" id="ti3">
      <tr:commandLink id="b" />
      <tr:commandLink id="c" />
      <tr:commandLink id="d" />
    </tr:tabIndex>
    <tr:tabIndex type="manual" order="e g" id="ti4">
      <tr:commandLink id="e" />
      <tr:commandLink id="f" />
      <tr:commandLink id="g" />
    </tr:tabIndex>
  </tr:tabIndex>
</tr:tabIndex>
```

What would have to happen to support this is probably a wrapped ResponseWriter. It would add `tabindex` to each element that supports a `tabindex` (like anchor tags). It would set aside indexes for the specified components in the page. It does have the complexity of what to do so that the order is known. In this case, ti4 is supposed to be tabbed before ti3. But at each element is rendered, there is no way to know how many tab indexes will be give out for ti4 as ti3 is being rendered. So the code would have to use jumps (say leave 1000 open). Meaning that ti1 would start at 1, ti2 start at 1000, ti3 start at 4000 and ti4 start at 3000. This is not too clean, but no the code work is less.

Pros:

- Current components would not have to be modified
- Simpler implementation for page authors

Cons:

- Wrapped ResponseWriter leaves a lack of fine level control for component authors
- Have to use component gaps (like 1000 in this case) which
    - Is a messy API
    - Can cause bugs with many elements (anchors in a tree for example)
- Can't turn off tabs for certain componets (tab index of -1)

Rating:

{*} {*} {*}  {o}

Custom Parent Component - 2

Easier to show than tell:

```
<tr:tabIndex startAt="next">
  <tr:commandLink id="a" />
  <tr:tabIndex startAt="next">
    <tr:tabIndex startAt="4">
      <tr:commandLink id="b" />
      <tr:commandLink id="c" />
      <tr:commandLink id="d" />
    </tr:tabIndex>
    <tr:tabIndex startAt="next">
      <tr:commandLink id="e" />
      <tr:commandLink id="f" />
      <tr:commandLink id="g" />
    </tr:tabIndex>
  </tr:tabIndex>
</tr:tabIndex>
```

Just a variation of above. This just makes the setting of tab indexes more manual instead of using 1000 increments

Pros:

- Current components would not have to be modified
- Simpler implementation for page authors
- No 1000 increments

Cons:

- Wrapped ResponseWriter leaves a lack of fine level control for component authors
- Can't turn off tabs for certain componets (tab index of -1)
- Prone to errors if startAt is not specified accurately

Rating:

{*} {*} {*} 💡 {o}

Custom Parent Component and tab indexes - 3

Easier to show than tell:

```
<tr:tabIndex id="ti1">
  <tr:commandLink id="a" />
  <tr:tabIndex id="ti2">
    <tr:commandLink id="b" />
    <tr:commandLink id="c" tabIndex="before:b" />
    <tr:commandLink id="d" tabIndex="before:c" />
  </tr:tabIndex>
  <tr:tabIndex tabIndex="before:ti2">
    <tr:commandLink id="e" />
    <tr:commandLink id="f" tabIndex="before:e" />
    <tr:commandLink id="g" />
  </tr:tabIndex>
</tr:tabIndex>
```

There could be more variations on this. For example, no `tabIndex` component, just use before, after and disabled like abilities in each component `tabIndex` attribute

Pros:

- Component level control over tab indexes
- Relative tab indexes to help with maintaining over time
- `tabIndex` component would make grouping section easier instead of having to do every component

Cons:

- Would be a pain in the rear to implement
  - Would need some pre-sweep of all the components to figure out their order before rendering
  - Each component or component renderer would have to support this pre-sweep
  - Would have to change each renderer to add `tabindex` onto certain values

Rating:

{*} {*} {*} {*} 💡

TODO

Suggest something else! Can we find a solution worthy of a 5 {*} rating?