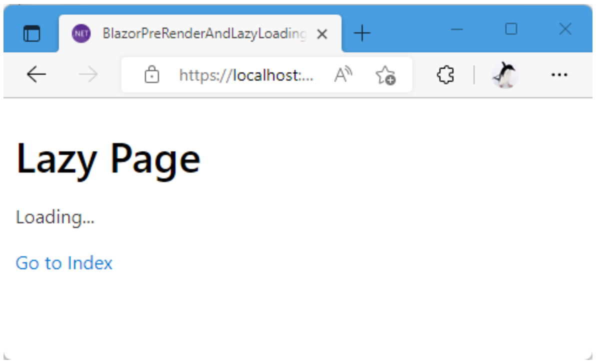
The scenario of unexpected behavior for me



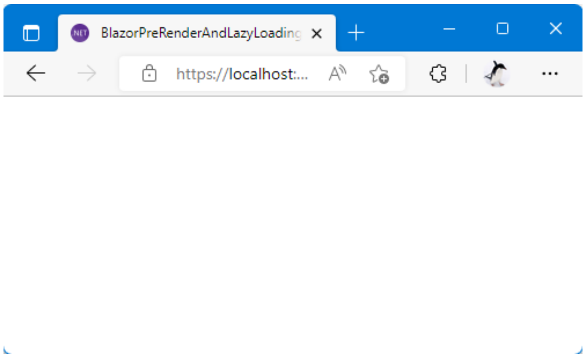
Lazy Page

Loading...

Go to Index

Start from "/lazy-page" (or reload on "/lazy-page").

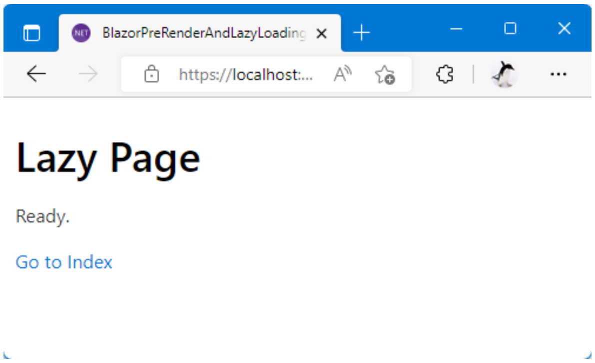
The pre-rendered contents of the "Lazy Page" page will be visible. At this time, Blazor runtime has not entirely started.
(You can see the status text indicates "Loading..." on the "Lazy Page" page)



After Blazor runtime has been started, then the app detects the current page is a lazy loading page.
So, the app starts lazy loading for the "Lazy Page" assembly.

But during loading the "Lazy Page" assembly, the contents that the browser is showing are **cleared**!

(**Desired Behavior**: the pre-rendered contents of the "Lazy Page" page will be kept even during the lazy loading of the assembly.)



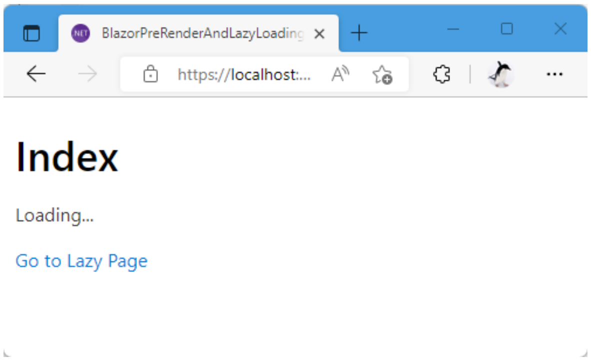
Lazy Page

Ready.

Go to Index

After lazy loading for the "Lazy Page" assembly completely, the contents of the "Lazy Page" page will be shown again with the "Ready" indicator.

Comparison scenario: An expected behavior for me



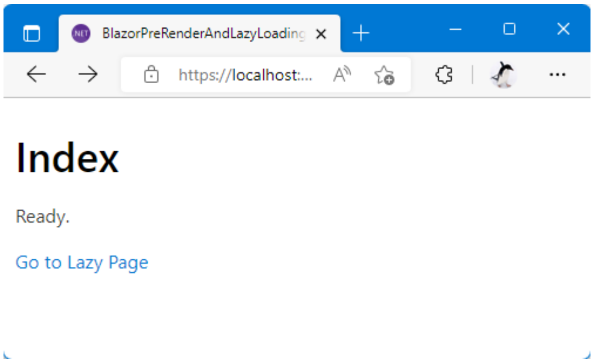
Index

Loading...

Go to Lazy Page

Start from "/" (or reload on "/").

The pre-rendered contents will be visible, and the Blazor runtime has not entirely started. (You can see the status text indicates "Loading..." on the “Index” page)



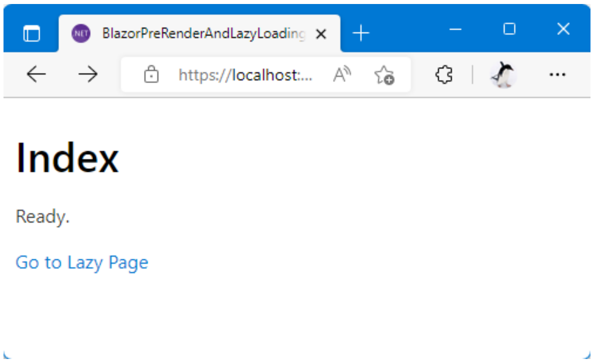
Index

Ready.

Go to Lazy Page

After Blazor runtime has been started, the status text will indicate "Ready".

Click the "Go to Lazy Page" hyperlink.



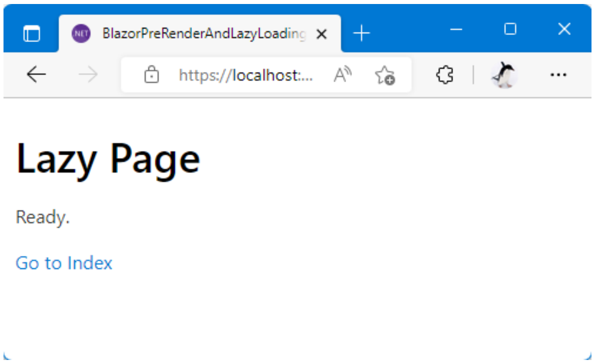
Index

Ready.

Go to Lazy Page

Lazy loading for the "Lazy Page" assembly will be started.

It takes a while, but the browser keeps the displayed contents of the "Index" page because the "<Naviating>" render fragment is not specified on this app.



Lazy Page

Ready.

Go to Index

After lazy loading for the "Lazy Page" assembly completely, the contents of the "Lazy Page" page will be shown.