HOWTO: Delete Annotations In WebDocumentViewer (WDV)

Background

WebDoucmentViewer annotation controller has a function to delete a given annotation.

[deleteFromPage(pageIndex, annoIndex)](#)

 usage: assumes your WebDocumentViewer object is named _viewer

```
ar pageIndex = _viewer.getCurrentPageIndex(); var annotationIndex = 0; // first annotation on
the page for simplicity _viewer.annotations.deleteFromPage(pageIndex, annotationIndex);
```

The difficulty is that in order to delete an annotation, you need to know two things, pageIndex and annoIndex (within the page)

pageIndex - the pages in WDV are zero index so first page is 0, second is 1, and so on. This is the page the specific annotation lives on

annoIndex - the annotations within the page, again 0 indexed, so the first annotation is index 0, the next is 1 and so on) within the page in question where the annotation appears

What can make this difficult/frustrating is that while there is a way to ask "given this annotation, what page is it on" but not "what is its index within that page"

However, lets back up one second. In order to do any of this we need to have a working annotation object "in hand"

There are plenty of functions that get you an annotation object...

for example - get the first annotation on the first page

var anno = _viewer.annotations.getAnnotationsFromPage(0)[0];

similarly, you could get selected annotations

this code will find the first selected annotation (if any)

var selectedAnno = _viewer.annotations.getSelected()[0];

So, now assuming that either anno or selectedAnno is not null, congratulations, you have a live reference to an annotation object

Now, as a side note, you could modify the annotation programmatically quite easily from here.. you could for example check what type it is and if it was a text annotation, update the text...

```
f (anno.type == 'text') { anno.text.value = anno.text.value + "\nI added A new Line Here!";
anno.update(); // NOTE it won't actually change until you call this }
```

Solving the Problem

so, you might think "hey, is there an anno.delete() I could call?"

That would be nice, but no.

you need to know the page the anno is on and the index

"OK fine, so surely I can get the indexes?"

Well, sort of

the annotation object does have a [getPageIndex()](#) function

so in this last case the anno was on page 0 so

var annoPageIndex = anno.getPageIndex();

and annoPageIndex will return 0

but

the anno does not have a "getIndexOnPage()" function

it doesn't have a delete() function either

so the problem we need to solve is "Given a specific annotation, determine its index within the page where it lives"

For this exercise we will pass in the viewer object as wdvViewer and the specific annotation as anno

```
unction getAnnoIndex(wdvViewer, anno) { var pageIndex = anno.getPageIndex() var pageAnnos =
wdvViewer.annotations.getFromPage(pageIndex); for (var i = 0; i < pageAnnos.length; i++) { if
(pageAnnos[i] === anno) { return i; } } return null; }
```

The anno is a javascript object being passed by reference so the === operator will ensure that we only return when we find the exact object passed in.

The anno knows what page its on, so we are safe in assuming that the call to getFromPage will include an array that will contain the target anno. However, just in case we do return null if something went wrong

However, in theory we will return the zero based index within the page where the annotation lives

Great, now, we can use this to delete the annotation

_viewer.annotations.deleteFromPage(anno.getPageIndex, getAnnoIndex(_viewer, anno));

however, note that if the annotation happens to be selected at the time, we can get a nasty issue with "disappearing grips"

to get around that we may want to ensure the selected is false

So, I've wrapped this all up in a nice little utility function deleteAnnotation

```
unction deleteAnnotation(wdvViewer, anno) { var pageIndex = anno.getPageIndex(); var
annoIndex = getAnnoIndex(wdvViewer, anno); if (annoIndex !== null) { if (anno.selected) {
anno.selected = false; anno.update(); } wdvViewer.annotations.deleteFromPage(pageIndex,
annoIndex); } }
```

Then, to really make it super lazy for me, I make a final "convenience function" where I know the name of MY viewer object and use that

```
unction deleteAnno(anno) { deleteAnnotation(_viewer, anno); }
```

This is purely optional but it just makes it so that I don't have to manually pass the viewer in every time I'm calling the delete.

So we can put all this together in one block

```
unction deleteAnno(anno) { deleteAnnotation(_viewer, anno); } function
deleteAnnotation(wdvViewer, anno) { var pageIndex = anno.getPageIndex(); var annoIndex =
getAnnoIndex(wdvViewer, anno); if (annoIndex !== null) { if (anno.selected) { anno.selected =
false; anno.update(); } wdvViewer.annotations.deleteFromPage(pageIndex, annoIndex); } }
function getAnnoIndex(wdvViewer, anno) { var pageIndex = anno.getPageIndex() var pageAnnos =
wdvViewer.annotations.getFromPage(pageIndex); for (var i = 0; i < pageAnnos.length; i++) { if
(pageAnnos[i] === anno) { return i; } } return null; }
```

The only thing you might need to change is the deleteAnno code if you call your viewer
something other than _viewer

Binding Delete Annotation to Delete Key Keypress

So, lets say you're good with all this now.. you're cool with how to get selected
annotations/annotation and how to delete, but a very common request is "I want to delete
the currently selected annotation when the user presses the delete key"

Technically, keyboard events are outside of the scope of DotImage support. They're just
standard JavaScript events you can bind directly to or bind using standard jQuery. However,
just for the sake of answering a very common question, we'll put together what we already
had with a simple keybinding

 We kind of need a function to delete selected annotations for this... Since we already have
our neat little deleteAnno(anno) we'll just write some code to get the selected annotations
and delete them

```
unction deleteSelectedAnnos() { var selected = _viewer.annotations.getSelected().flat(1);
_viewer.annotations.deselectAll(); //important to avoid "ghost grips" for (var i =
selected.length - 1; i >= 0; i--) { deleteAnno(selected[i]); } selected = undefined; }
```

Note how we are getting the selected annotations in a flat array (this is because
getSelected() returns a 2 dimensional array and we just need to simplify this down) and
that we walk backward.. this is because if there are multiple annos on a page we want to
avoid removing the wrong ones and the easiest way is to walk backward from highest index
to lowest.. just trust me on this

Now that you have the delete selected the keybind is really easy... just add this code to
your page

HOWTO: Delete Annotations In WebDocumentViewer (WDV)

```
(document).on('keydown', documentKeyDown); function documentKeyDown(e) { if (e.which == 46) {
deleteSelectedAnno(); } }
```

The first line is the jQuery selector that gets the whole document and we bind to the
keydown event and set our documentKeyDown handler the handler is simply looking for
what key you hit - 46 is the the DEL key... so if the key hit was DEL then we call
deleteSelected() and voila! we are DONE!

Atalasoft Knowledge Base

https://www.atalasoft.com/kb2/KB/50430/HOWTO-Delete-Annotations-In-WebDocum...