

How to add interactivity to PDF documents using Actions

Written by Apitron Documentation Team

Introduction

In our [latest post](#) we've explored all possible ways to use links in PDF and promised to write an additional post about pdf actions. As you may recall, we used `Destination` objects for setting the targets of our links. Another way to get job done and extend the range of possible usages is to use `Actions`. They are described in section 12.6 "*Actions*" of the pdf specification and provide an alternative way to add interactivity to your documents.

In addition to jumping to a destination, actions can be used to launch application, execute Adobe JavaScript, play sound, change annotation's appearance state, etc.

All available action types are listed below:

- `GoTo` - go to a destination in the current document
- `GoToR` - ("Go-to remote") go to a destination in another document
- `GoToE` - ("Go-to embedded") go to a destination in an embedded file
- `Launch` - launch an application, usually to open a file
- `Thread` - begin reading an article thread
- `URI` - resolve a uniform resource identifier
- `Sound` - play a sound
- `Movie` - play a movie
- `Hide` - set an annotation's `Hidden` flag
- `Named` - execute an action predefined by the conforming reader
- `SubmitForm` - send data to a uniform resource locator
- `ResetForm` - set fields to their default values
- `ImportData` - import field values from a file
- `JavaScript` - execute a JavaScript code
- `Rendition` - controls the playing of multimedia content
- `Trans` - updates the display of a document, using a transition dictionary
- `GoTo3DView` - sets the current view of a 3D annotation

Actions can be assigned to annotations objects and outline items ("bookmarks"), see the code sample provided in the next section.

Code sample

Below is the slightly modified code sample from the post about links, its only difference is the use of `Action` object instead of `Destination`. It creates a bookmark and a link both having the same JavaScript action assigned. This action shows a simple message box with “hello world” message.

```
public void CreateLinkAndBookmarkWithJavaScriptAction()
{
    // create pdf document
    FixedDocument doc = new FixedDocument();

    // create javascript action, showing the message box
    Action action = new JavaScriptAction("app.alert('Hello world using Apitron PDF Kit!');");

    // add a bookmark for second page
    doc.Bookmarks.AddFirst(new Bookmark(action, "JS Bookmark"));

    // add first page with text link
    Page firstPage = new Page();

    // add link text
    TextObject linkTextObject = new TextObject();
    linkTextObject.AppendTextLine("Click to execute JavaScript action");

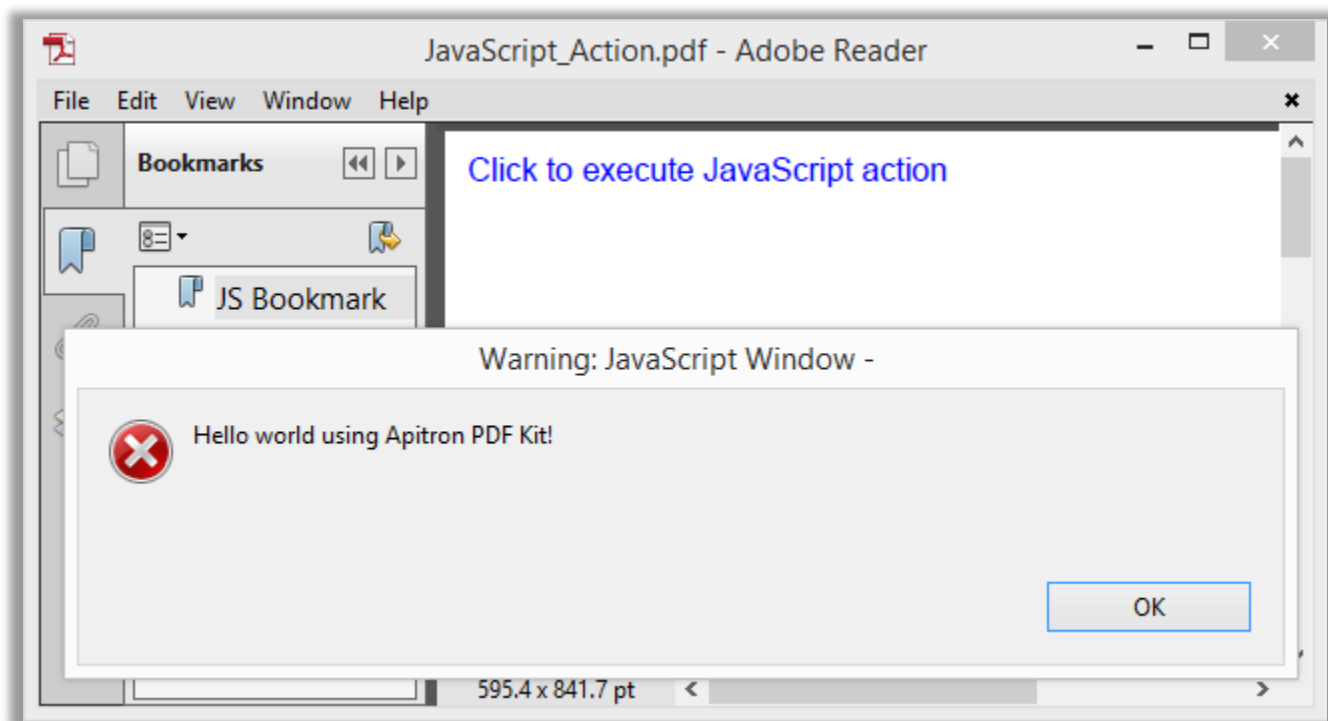
    firstPage.Content.SaveGraphicsState();
    firstPage.Content.SetDeviceNonStrokingColor(RgbColors.Blue.Components);
    firstPage.Content.Translate(10, 820);
    firstPage.Content.AppendText(linkTextObject);
    firstPage.Content.RestoreGraphicsState();

    // add link annotation covering the text
    LinkAnnotation link = new LinkAnnotation(new Boundary(10, 815, 230, 835));
    link.BorderStyle = new AnnotationBorderStyle(0);
    link.Action = action;
    firstPage.Annotations.Add(link);

    // append pages
    doc.Pages.Add(firstPage);

    // save document
    using (Stream stream = File.Create("JavaScript_Action.pdf"))
    {
        doc.Save(stream);
    }
}
```

See the image below showing the results of clicking the link with assigned action.



Pic. 1 Bookmark and link with JavaScript action

It's easy and simple as it should be. If you'd like to read more about Adobe's JavaScript implementation, read the following book - "JavaScript™ for Acrobat® API Reference".

Conclusion

As you can see, our product, [Apitron PDF Kit for .NET](#), can be used to add interactivity to your pdf documents with unmatched ease and clearness. Using its API you'll be able to: create links, actions and bookmarks, create or modify pdf forms, and much more. Read our [free book](#), or contact our support if you have any questions regarding its capabilities.