iws.viewer

# 특징

* MultiTif 이미지 파일 지원.
* Jpeg2000 이미지 파일 지원.
* 썸네일 지원.
* 주석 지원.
* Zoom, rotation support.
* Cross-browser support

# 제공

dist/

|  |  |
| --- | --- |
| iws.viewer.css | |
| iws.viewer.js | |
| engine/ | iws.w.loader.js |
| iws.w.split.js |
| Iws.w.resample.js |
| img/ | iws.viewer.background.png |
| iws.viewer.toolbar.png |
| Vendor/ | jquery-2.2.3.js |
| spectrum.css |
| spectrum.js |
| hammer.min.js |
| iws-scrollbar.css |
| iws-scrollbar.js |

# 설치

Include files:

|  |
| --- |
| <script src="/path/vendor/jquery-2.2.3.js"></script>  <link href="/path/vendor/spectrum.css" rel="stylesheet">  <script src="/path/vendor/spectrum.js"></script>  <script src="/path/vendor/hammer.min.js"></script>  **<link href="/path/iws.viewer.css" rel="stylesheet">**  **<script src="/path/iws.viewer.js"></script>** |

# 사용법

뷰어 객체 생성하기

|  |
| --- |
| <!-- 뷰어로 사용할 div element 선언 -->  <div **id="multiPage"** style="width:800px;height:600px;"></div> |
| <!-- element를 넣어 객체를 얻음 -->  var element = document.getElementById("multiPage");  var options = {  path:'./',  fitMode:{name:'auto'},  useAnnotation:true,  reSampleUse:true  };  var **viewer = new iws.multiPage**(element, options); |

# 옵션

* **path : 환경 셋팅 경로.**
  + Type : String
  + Default : “”

: 꼭입력해야 하는 항목으로 iws.viewer.js 파일이 있는 경로를 입력합니다.

예)

path:'./' ,

path:'/IWS\_Solution/WebContent/solution/iws/js',

path:'http://localhost:8181/IWS\_Solution/WebContent/solution/iws/js',

* fitMode : 화면 맞춤
  + Type : object
  + Default : {name:’auto’}
  + Value 종류
    - {name:”auto”} : 화면에 딱 맞도록 이미지 길이에 따라 가로맞춤/세로맞춤을 알아서 자동 지정해 줌.
    - {name:”width”} : 가로 길이에 맞추어 이미지크기를 조절하여 보여줌.
    - {name:”height”} : 세로 길이에 맞추어 이미지크기를 조절하여 보여줌.
    - {name:”scale”, scale:1.0} : scale 값에 따라 이미지를 확대하여 보여줌.
* useAnnotation : 주석 툴바 사용 여부.
  + Type : Boolean
  + Default : true
* reSampleUse: 화면에 보이는 이미지를 가공하여 더욱 잘보이도록 하는 기법 사용 여부
  + Type : Boolean
  + Default : true
* thumbWidth : 썸네일 가로 지정.
  + Type : Number
  + Default : 200
* thumbHeight : 썸네일 세로 지정.
  + Type : Number
  + Default : 160

# 버튼 지원

* 이미지 처리 버튼
  + 이미지 확대/축소
  + 가로 맞춤/세로 맞춤/자동 맞춤
  + 좌로 회전/우로 회전/180도 회전
  + 이미지 이동
  + 영역 선택
  + 돋보기 사용
  + 이전/다음 페이지 이동
* 주석 처리 버튼
  + 주석 선택
  + 자유선 그리기
  + 라인 그리기
  + 사각 그리기
  + 텍스트 넣기

# 키보드 지원

* ↑ : 썸네일 창에서 이전 썸네일 선택
* ↓ : 썸네일 창에서 다음 썸네일 선택
* Shift + a : 페이지 창에서 모든 주석 선택
* Del : 선택 주석 삭제

# 마우스 휠 지원

* 이미지 확대/축소

# Methods

* clear()

: 이미지 화면 초기화.

* loadFile(filedata)
  + filedata
    - Type : ArrayBuffer
    - 이미지 파일 바이너리.

: FileReader로 readAsArrayBuffer 함수를 사용하여 읽은 데이터 또는 서버에서 가져온 바이너리 데이터를 파라미터로 받아 이미지를 로딩한다.

* appendFile(filedata)
  + filedata
    - Type : ArrayBuffer
    - 이미지 파일 바이너리.

: FileReader로 readAsArrayBuffer 함수를 사용하여 읽은 데이터 또는 서버에서 가져온 바이너리 데이터를 파라미터로 받아 이미지를 추가한다.

* loadUrl(url)
  + url
    - Type : String
    - 이미지 파일을 받을 수 있는 http url

: 이미지 url로 이미지 로드.

* appendUrl(url)
  + url
    - Type : String
    - 이미지 파일을 받을 수 있는 http url

: 이미지 url로 이미지 추가.

* loadAnnotation(annotation)
  + annotation
    - Type : String
    - Json형태의 주석 정보 스트링.
* saveAnnotation()

: 주석 저장

* 주석 저장 성공하면 “saveAnnotation” 이벤트를 발생하여 결과를 넘김.

|  |
| --- |
| <!-- 이벤트로 주석 데이터 받는 방법 -->  viewer.observe(**'saveAnnotation'**, **function**(data){  **if**(data.**success**){  alert(data.annotation);  } **else** {  alert(**'There are no saved comments.'**);  }  }, **false**);  data.annotation이 주석 Json데이터임 |

* saveImage()

: 현재 보고 있는 이미지를 Png 포멧의 바이너리로 저장함.

* 이미지 저장함수 호출하면 “saveImage” 이벤트를 발생하여 결과를 넘김.

|  |
| --- |
| <!-- 이벤트로 이미지 데이터 받는 방법 -->  viewer.observe(**'saveImage'**, **function**(data){  **if**(data.**success**){  //data.**byteArray**  } }, **false**);  data.byteArray 부분이 이미지 데이터임. |

* zoomIn()

: 이미지를 10% 확대함.

* zoomOut()

: 이미지를 10% 축소함.

* rotate(angle)
  + angle
    - Type : Number
    - 90, 180, 270 도 중 각도 선택

: 주어진 각도 만큼 회전함.

* setFitMode(mode)
  + mode
    - Type : object
    - {name:”auto”} : 화면에 딱 맞도록 이미지 길이에 따라 가로맞춤/세로맞춤을 알아서 자동 지정해 줌.
    - {name:”width”} : 가로 길이에 맞추어 이미지크기를 조절하여 보여줌.
    - {name:”height”} : 세로 길이에 맞추어 이미지크기를 조절하여 보여줌.
    - {name:”scale”, scale:1.0} : scale 값에 따라 이미지를 확대하여 보여줌.
* setTrackMode(mode)
  + mode
    - Type : String
    - “move” : 이미지가 화면보다 크면 마우스 클릭하여 이미지 이동 가능.
    - “range” : 마우스 클릭하여 드레그 한 영역의 좌표를 얻어옴

|  |
| --- |
| 마우스 드래그 후 버튼업 할 때 발생되는 “**mouse:drawRanged** “이벤트로 선택 좌표 값을 얻는다.  viewer.observe(**"mouse:drawRanged"**, **function**(rc){  alert(**"영역좌표("**+rc.**left**+**", "**+rc.**top**+**", "**+rc.**right**+**", "**+rc.**bottom**+**")"**); }); |

* + - “glass” : 마우스 클릭하면 돋보기 화면을 볼 수 있음.
* pagePrev()

: 이전 페이지로 이동.

* pageNext()

: 다음 페이지로 이동.

* pageCount()

: 페이지 개수 리턴

* showAnnotation(show)
  + show
    - Type : String
    - “show” : 주석 관련 툴바를 보인다.
    - “hide” : 주석 관련 툴바를 감춘다.
* crop(rect, callback)

rect : {left: value, top:value, right:value, bottom:value};

callback(src, binary){} : 컷팅 후 호출되는 콜백함수.

* src : 컷팅된 base64스트링.
* binary : 컷팅된 이미지의 byte array데이터. 필요한 데이터 선택하여 활용가능.
* destroy()
* 뷰어 객체가 메모리 상에서 제거됨.

# Property

* pageIndex

: 현재 포커스가 있는 썸네일의 위치를 반환하거나 셋팅함.(0 base)

# Events

* 주석 저장 이벤트

|  |
| --- |
| viewer.observe(**'saveAnnotation'**, **function**(data){  **if**(data.**success**){  alert(data.**annotation**); *//data.annotation이 주석 Json데이터임*  } }, **false**); |

: 툴바의 저장 버튼 클릭 또는 saveAnnotation() 함수를 호출하면 발생함. Json 스트링의 주석데이터 받음.

* 이미지 저장 이벤트

|  |
| --- |
| viewer.observe(**'saveImage'**, **function**(data){  **if**(data.**success**){  alert(data.**byteArray**); *//png image*  } }, **false**); |

: saveImage() 함수 호출시 발생함. 바이너리로 된 Png이미지 포멧 받음.

* 영역 선택 이벤트

|  |
| --- |
| viewer.observe(**"mouse:drawRanged"**, **function**(rc){  alert(**"영역좌표("**+rc.**left**+**", "**+rc.**top**+**", "**+rc.**right**+**", "**+rc.**bottom**+**")"**); }); |

: fitMode가 영역지정 모드인 경우 마우스를 드레그하여 마우스버튼 업 하는 경우 좌표를 포함한 객체를 리던함.

# Browser support

* Chrome(latest)
* FireFox(latest)
* Edge(latest)
* Internet Exporer 11+

# License