

Apertura xlsx Viewer

La gwXlsxViewer è una particolare tipologia di scheda descritta [qui](#). Introdotta in **4.7.3** ([issue #1417](#))

La scheda può essere aperta in floatingPane o in tab.

openGwXlsxViewerFloatingPane()

Per aprire questa tipologia di floatingPane è esposta l'apposita function `openGwXlsxViewerFloatingPane()`.

Parametri

- **params** Object, required. Contains all parameters
- **title** String, optional, default Strina vuota
- **tabIndex** Integer, optional, default null

Parametri dentro params

- **href**: String, required, default `gwContextPath+'/rest/exportByGroovyExecution'`. url del controller che genera i bytes
- **method**: String, required, default 'POST' method associato ad href
- **fileName**: String, required, nome del file che verrà generato
- **data**: Object, optional, Dati da postare in acciazione a href e method Es: `{ groovy: 'export_pianificazione.groovy', fileName: fileName, otherStuff: 1 }`
- **viewerOptions**: Object, optional, default null. Opzioni passabili al visualizzatore basato su LuckySheet per sovrascrivere il comportamento di default (full readonly). Parametri <https://dream-num.github.io/LuckysheetDocs/guide/config.html#configuration-item>

Esempi

Apertura di un file xlsx prodotto in maniera standard:

[example.js](#)

```
var fileName = 'Pianificazione Attività.xlsx';

//see link for luckysheet options
//
https://dream-num.github.io/LuckysheetDocs/guide/config.html#configuration-item
//by default geoweb set all viewerOptions in aim to obtain readonly
behavior
var viewerOptions = {};
```

```
var params = {
  href: gwContextPath+'/rest/exportByGroovyExecution',
  method: 'POST',
  fileName: fileName,
  data: {
    groovy: 'export_pianificazione.groovy',
    fileName: fileName,
    startDate: startDate,
    endDate: endDate
  },
  viewerOptions: viewerOptions
};
var floatingPane = openGwXlsxViewerFloatingPane(params);
```

Apertura con viewerOptions con valori di default:

[example.js](#)

```
var fileName = 'Pianificazione Attività.xlsx';
var startDate = new Date(); //TODO
var endDate = new Date(new Date().getTime()+((1000*60*60*24*365*2)));
//TODO

//see link for luckysheet options
//
https://dream-num.github.io/LuckysheetDocs/guide/config.html#configuration-item
var viewerOptions = {
  autoFormatw: false,
  allowCopy: true,

  showinfobar: false,
  sheetFormulaBar: false,

  showtoolbar: false,
  /*showtoolbarConfig: {
    undoRedo: true, //Undo redo
    paintFormat: true, //Format brush
    currencyFormat: true, //currency format
    percentageFormat: true, //Percentage format
    numberDecrease: true, //'Decrease the number of decimal places'
    numberIncrease: true, //'Increase the number of decimal places'
    moreFormats: true, //'More Formats'
    font: true, //'font'
    fontSize: true, //'Font size'
    bold: true, //'Bold (Ctrl+B)'
    italic: true, //'Italic (Ctrl+I)'
```

```
    strikethrough: true, //'Strikethrough (Alt+Shift+5)'  
    underline: true, // 'Underline (Alt+Shift+6)'  
    textColor: true, //'Text color'  
    fillColor: true, //'Cell color'  
    border: true, //'border'  
    mergeCell: true, //'Merge cells'  
    horizontalAlignMode: true, //'Horizontal alignment'  
    verticalAlignMode: true, //'Vertical alignment'  
    textWrapMode: true, //'Wrap mode'  
    textRotateMode: true, //'Text Rotation Mode'  
    image: true, // 'Insert picture'  
    link: true, // 'Insert link'  
    chart: true, //'chart' (the icon is hidden, but if the chart  
plugin is configured, you can still create a new chart by right click)  
    postil: true, //'comment'  
    pivotTable: true, //'PivotTable'  
    'function': true, //'formula'  
    frozenMode: true, //'freeze mode'  
    sortAndFilter: true, //'Sort and filter'  
    conditionalFormat: true, //'Conditional Format'  
    dataVerification: true, // 'Data Verification'  
    splitColumn: true, //'Split column'  
    screenshot: true, //'screenshot'  
    findAndReplace: true, //'Find and Replace'  
    protection: true, // 'Worksheet protection'  
    print: false // 'Print'  
},*/  
  
showsheetbar: true,  
showsheetbarConfig: {  
    add: false, //Add worksheet  
    menu: true, //Worksheet management menu  
    sheet: true //Worksheet display  
},  
  
showstatisticBar: true,  
showstatisticBarConfig: {  
    count: false, // Count bar  
    view: true, // Print view  
    zoom: true // Zoom  
},  
  
showConfigWindowResize: true,  
  
userMenuItem: null,  
  
myFolderUrl: null,  
  
cellRightClickConfig: {  
    copy: false, // copy  
    copyAs: false, // copy as
```

```
    paste: false, // paste
    insertRow: false, // insert row
    insertColumn: false, // insert column
    deleteRow: false, // delete the selected row
    deleteColumn: false, // delete the selected column
    deleteCell: false, // delete cell
    hideRow: false, // hide the selected row and display the
selected row
    hideColumn: false, // hide the selected column and display the
selected column
    rowHeight: false, // row height
    columnWidth: false, // column width
    clear: false, // clear content
    matrix: false, // matrix operation selection
    sort: false, // sort selection
    filter: false, // filter selection
    chart: false, // chart generation
    image: false, // insert picture
    link: false, // insert link
    data: false, // data verification
    cellFormat: false // Set cell format
},

sheetRightClickConfig: {
    'delete': false, //Delete
    copy: false, //Copy
    rename: false, //Rename
    color: false, //Change color
    hide: false, //Hide, unhide
    move: false //Move to the left, move to the right
},

enableAddRow: false,
enableAddBackTop: false
};

var params = {
    href: gwContextPath+'rest/exportByGroovyExecution',
    method: 'POST',
    fileName: fileName,
    data: {
        groovy: 'export_pianificazione.groovy',
        fileName: fileName,
        startDate: startDate,
        endDate: endDate,
        //viewerOptions: viewerOptions
    }
};
```

```
var floatingPane = openGwXlsxViewerFloatingPane(params);
```

openGwXlsxViewerTab()

Per aprire questa tipologia di tab è esposta l'apposita function openGwXlsxViewerTab().

Parametri

- **params** Object, required. Contains all parameters
- **title** String, optional, default String vuota
- **tabIndex** Integer, optional, default null

Parametri dentro params

- **href**: String, required, default gwContextPath+'rest/exportByGroovyExecution'. url del controller che genera i bytes
- **method**: String, required, default 'POST' method associato ad href
- **fileName**: String, required, nome del file che verrà generato
- **data**: Object, optional, Dati da postare in acciazione a href e method Es: { groovy: 'export_pianificazione.groovy', fileName: fileName, otherStuff: 1 }
- **viewerOptions**: Object, optional, default null. Opzioni passabili al visualizzatore basato su LuckySheet per sovrascrivere il comportamento di default (full readonly). Parametri <https://dream-num.github.io/LuckysheetDocs/guide/config.html#configuration-item>

Esempi

Apertura scheda con valori di default:

[example.js](#)

```
var fileName = 'Pianificazione Attività.xlsx';
var startDate = new Date(); //TODO
var endDate = new Date(new Date().getTime()+(1000*60*60*24*365*2));
//TODO

//see link for luckysheet options
//
https://dream-num.github.io/LuckysheetDocs/guide/config.html#configuration-item
//full readonly behavior set by geoweb as default
var viewerOptions = {
    //...
};

//it cointains properties destinated to groovy
```

```
var groovyData = {
  groovy: 'export_pianificazione.groovy',
  fileName: fileName,
  startDate: startDate,
  endDate: endDate
};

var params = {
  href: gwContextPath+'/rest/exportByGroovyExecution',
  method: 'POST',
  fileName: fileName,
  data: groovyData,
  viewerOptions: viewerOptions
};
var title = 'title';
openGwXlsxViewerTab(params, title);
```

From:
<https://wiki.geowebframework.com/> - **GeowebFramework**

Permanent link:
https://wiki.geowebframework.com/doku.php?id=custom:api_js_opengwlsxviewertab&rev=1730219836

Last update: **2024/10/29 17:37**

