New Formula Bar in Spreadsheets

(How to Create Custom UI Component with JSDialogs)

Szymon Kłos

Software Engineer

Collabora Productivity

szymon.klos@collabora.com

COOL days

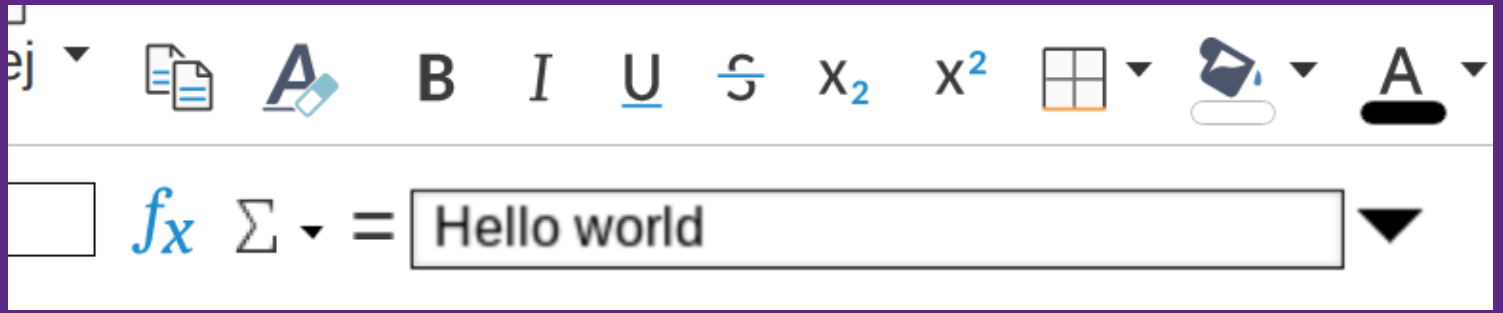
Agenda

- Problems in the previous version
- Why it needs dedicated solution
- Custom widgets implementation using JSDialogs

Collabora Productivity

Problems in the Previous Version

- Blurry text with non standard zoom level
- Impossible to style using CSS / theme
- Poor UX when trying to move cursor / selection
- Not accessible content for screen readers



Collabora Productivity

COOL days

Why it Needs a Dedicated Solution

- JSDialogs works only with “welded” widgets
- Formula bar welding was hard, possible only partially with current state of the welding code, risky also for desktop application
- Formula input is a DrawingArea widget which would be an image…

  => doesn’t fix UX…

Collabora Productivity

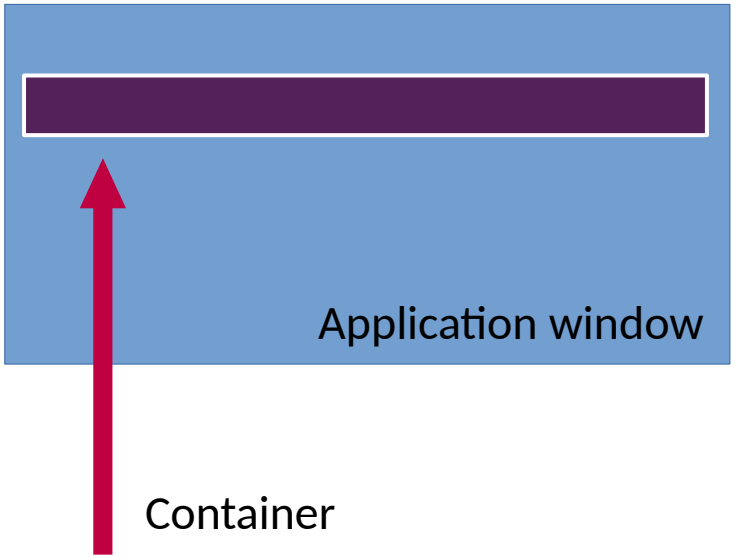
COOL days

Custom Widgets Implementation Using JSDialogs

Collabora Productivity

What We Need

1. New file for our UI component
2. Target container in the DOM
3. JSDialog builder
4. JSON source



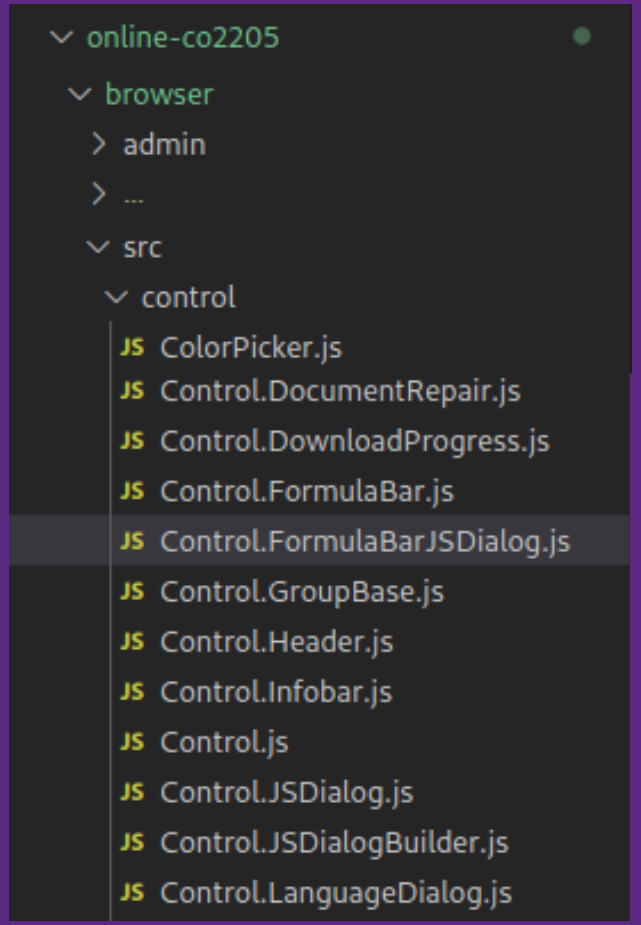
New File For Our UI Component

To keep naming unified
it would be good to put it in:

<REPO>/browser/src/control/

With name:

Control.<NAME>.js



New file for our UI component

- New “class” which extends **L.Control**
- Standard methods:

  onAdd(map), **onRemove**()
- Fabric method:

```
317
318   L.control.formulaBarJSDialog = function (options) {
319       return new L.Control.FormulaBarJSDialog(options);
320   };
321
```

- Register in Control.UIManager.js:

```
if (docType === 'spreadsheet') {
    this.map.addControl(L.control.sheetsBar({shownavigatior
    this.map.addControl(L.control.formulaBar());
    var formulabar = L.control.formulaBarJSDialog();
    this.map.formulabar = formulabar;
    this.map.addControl(formulabar);
    $('#toolbar-wrapper').addClass('spreadsheet');
```

```
6   /* global _ _UNO UNOKey */
7   L.Control.FormulaBarJSDialog = L.Control.extend({
8       container: null,
9       builder: null,
10      dirty: true, // if we should allow to update based on servers setText
11
12      onAdd: function (map) {      You, 6 months ago • jsdialogs: formulabar
13          this.map = map;
14
15          this.map.on('formulabar', this.onFormulaBar, this);
16          this.map.on('jsdialogupdate', this.onJSUpdate, this);
17          this.map.on('jsdialogaction', this.onJSAction, this);
18
19          this.builder = new L.control.jsDialogBuilder(
20              {
                    mobileWizard: this,
                    map: this.map,
                    cssClass: 'formulabar jsdialog',
                    callback: this.callback.bind(this)
                });
        },

28      onRemove: function() {
29          this.map.off('formulabar', this.onFormulaBar, this);
30          this.map.off('jsdialogupdate', this.onJSUpdate, this);
31          this.map.off('jsdialogaction', this.onJSAction, this);
32      },
33
```

Collabora Productivity

JSDialog Builder

Properties for builder's constructor:

- mobileWizard - reference to the component using the builder, old naming from original use-case
- map – reference to the map object
- cssClass – classes added to every widget generated by the builder
- callback – custom callback which will handle all the user interaction

```
this.builder = new L.control.jsDialogBuilder(
    {
        mobileWizard: this,
        map: this.map,
        cssClass: 'formulabar jsdialog',
        callback: this.callback.bind(this)
    });
},
```

Different builders we have:

- jsDialogBuiler – dialogs, sidebar
- mobileWizardBuilder – mobile UI
- notebookbarBuilder - notebookbar

```
this.container = L.DomUtil.create('div', 'inputbar_container', parent);
this.container.style.width = '100%';

this.builder.build(this.container, data);      You, 6 months ago • jsdialo
```

"build" method creates widgets described by "data" JSON in target container.

Collabora Productivity

JSON Messages

3 Types of Messages:

- Full window update (initial message)
- Single widget update (content change)
- Action in the widget (selection)

JSON messages

Example “INCOMING” messages:

- jsontype – which ui component is a target
- id – unique id of the dialog / window
- control_id – target widget id

```
L.DomUtil.addClass(L.DomUtil.get('fo
this.onJSUpdate({
    data: {
        jsontype: 'formulabar',
        id: this.builder.windowId,
        'control_id': 'expand',
        control: {
            id: 'expand',
            type: 'pushbutton',
            text: '',
            symbol: 'SPIN_UP'
        }
    }
});
```

```
showButton: function(action, show) {
    this.onJSAction(
        {
            data: {
                jsontype: 'formulabar',
                id: this.builder.windowId,
                data: {
                    'control_id': action,
                    'action_type': show ? 'show' : 'hide'
                }
            }
        });
},
```

Collabora Productivity

User Interaction Handling

- objectType – widget type (input, button, listbox…)
- eventType – what happened? (change, selection…)
- object – source (id property is required inside)
- data – additional data
- _defaultCallbackHandler – just sends the message to the server without any processing

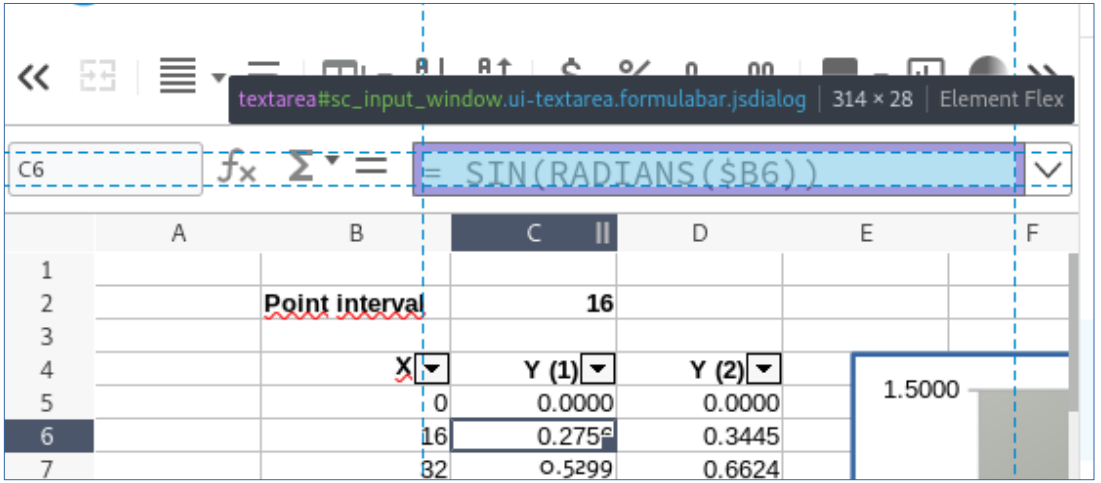
```
callback: function(objectType, eventType, object, data, builder) {
    if (object.id === 'expand') {
        var input = this.getInputField();
        if (input)
            this.toggleMultiLine(input);
        return;
    }

    // in the core we have DrawingArea not TextView
    if (object.id.indexOf('sc_input_window') === 0) {
        objectType = 'drawingarea';
        if (eventType === 'keypress' && data === UNOKey.RETURN || data === UNOKey.ESCAPE)
            builder.map.focus();
        else if (eventType === 'grab_focus')
            builder.map.onFormulaBarFocus();
    }

    builder._defaultCallbackHandler(objectType, eventType, object, data, builder);
},
```

Result

- Input field which talks to the server where we have only DrawingArea
- Messages from the Online are translated in the callback
- Server has additional code which sends JSDialog-like messages back
- It is using JSDialog framework pieces without having welded widget



Thanks !

By Szymon Kłos

Collabora
Productivity

@CollaboraOffice
hello@collaboraoffice.com
Collaboraoffice.com