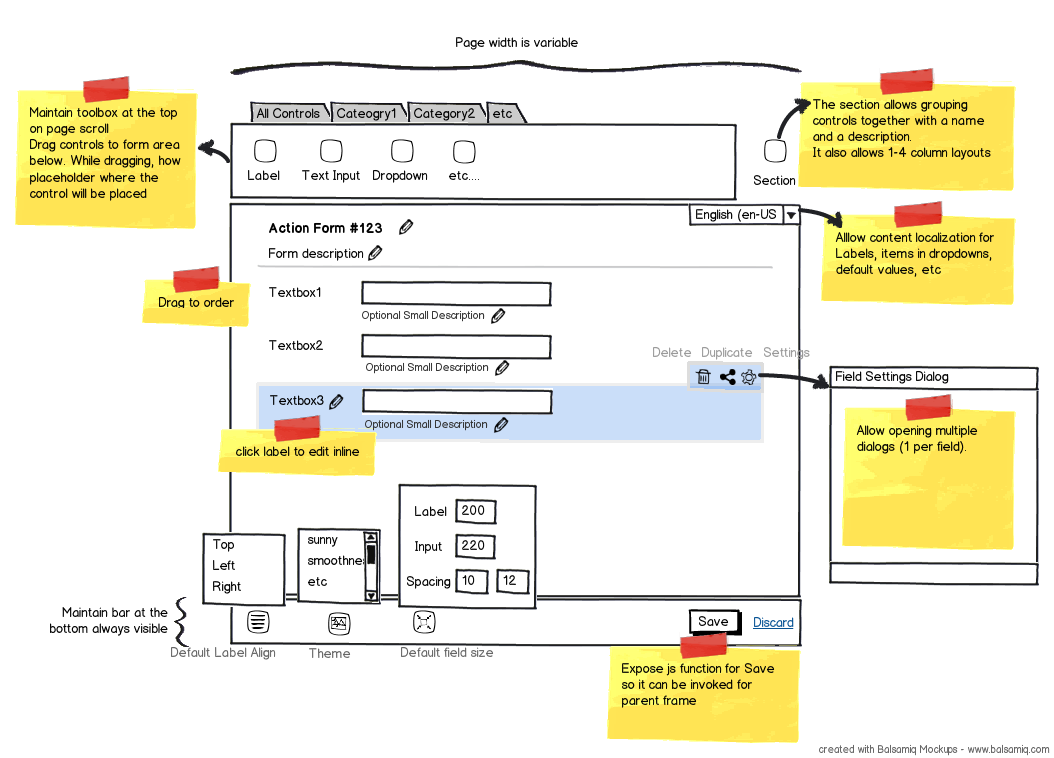
# Action Forms – jQuery From Builder

We need a simple WYSWYG Form Builder to allow users to build custom forms without the hassle. We need the simplicity of [wufoo.com](http://wufoo.com/) or [jotform.com](http://www.jotform.com).

We only need the UI for this. Most of the backend is already place. The communication will be through AJAX and JSON serialization/deserialization, so both the backend and the frontend will operate with same objects.

Most likely the form builder will run inside an IFrame.

## Form Builder Window



At the top there will be a list with all the controls we support. Each control can belong to one or several categories. From here controls can be dragged to the screen and they will initialize with minimal default settings. There will be a web service that returns the list of supported controls.

While dragging controls, placeholders need to appear where the control will be placed. I suggest implementing this with jQuery UI draggable, droppable and sortable but I’m open to other recommendations.

All fields we currently support can be translated to basic html controls except the datepicker, hidden fields and captcha. So there will be an option per field type to say how the control should be placed for rendering purposes. For now it will be either the name of an html field or an URL to an image placeholder. But in the future we’ll need a mechanism to render complex controls too.

When hovering controls they need to provide a visual feedback to reflect the hovered control. Also, new icons should appear to allow editing labels, deleting, cloning. Clicking the settings button or anywhere in the highlighted control will bring up the Field Settings dialog described below.

Dragging controls allows adjusting the order.

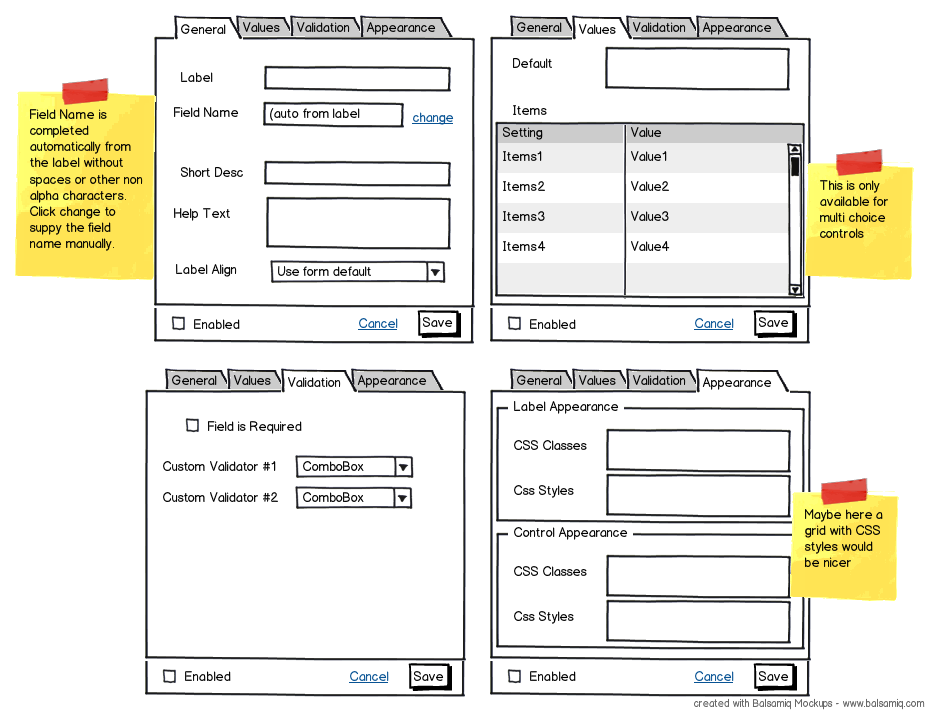
There are a few more options at the bottom that allow settings the label placement, the jQuery theme to use and the vertical and horizontal spacing between controls and labels. Note that these changes need to reflect immediately on the screen.

Saving the form will build a json with the form setting and a list of fields, each with its own settings. Then post this json to a Web Service (I’ll provide this). This json will also be provided to you on initialization via a web service so you can load the form back.

## Sections

A section is basically a fieldset where several fields can be grouped together, but also it allows multi-column setups. So a section needs to have its own settings dialog where it accepts a Title, a Description and the number of columns. Therefore fields can be dragged before the section, after the section or inside one of the columns of the section. Also allow for overriding default label placement per section and simple CSS Styles and CSS Classes text inputs.

## Field Settings



When clicking on an item or on the settings icon the settings dialog opens. Allow a dialog per field so admins can open multiple dialogs and copy data between them.

On general screen, the Field Name populates automatically by removing non alpha characters from the label. If the user clicks Change and he inputs something manually then the button needs to change to “Build From Label” and changing the label will no longer update the name field.

The values screen has the default value and a list of possible values. Allow some variations so default value can be actually named differentially (for example, Agree to Terms field will have the terms). Also, the list of possible values is only relevant to some field types.

The Validation screen is just saving and passing some settings around. The list of custom action names will come from a web service.

The Appearance screen allows putting additional styles and CSS classes on labels and fields. The CSS Styles need to take effect real time on front-end.

## Content Localization

There will be a default (fallback) language and a list of supported languages. When switching to a language all fields that have localizable content (such as labels, description, default value, list of items, etc) need to show the content for the selected language. If it doesn’t exist, then take the content from the fallback language.

## Other Notes

Since we’re running in an iframe we don’t have any restrictions. Use latest jQuery and jQuery plugins as needed. Make everything smooth and nice looking. No fancy design needed but it needs to look good, clean and simple like wufoo or jotform. Highlight fields when hovered, keep highlighted while forms are opened, use nice transitions when adding and removing fields, etc.