

## Contents

1	Preparation	2
2	Setup your initial environment	2
3	Create a draggable blocks	4
4	specify editable content	6

# 1 Preparation

## 1.1 Create Wireframe

Draw a wireframe of a “maximum” email, it shall contain every variation and shape of email.

### 1.1.1 Identify Features

In the wireframe identify the points of interaction in the email editor

- Main Blocks
- editable content
- variation point

### 1.1.2 Identify draggable blocks

Identify the smallest parts which can be dragged to the mail. These are blocks

Decide if you need an extra block or you can handle it by variations in a block

## 2 Setup your initial environment

For sake of simplicity, we make a very simple HTML - Template.

1. create `templates/tutorial/template-tutorial.html` and enter the basic structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>

<div id="preface">
  <h1>This is the Mosaico Tutorial</h1>

</div>

<div data-ko-container="main">
</div>

<div id="trailer">
  <h1>This is the end of Mosaico Tutorial</h1>
</div>

</body>
</html>
```

The `<div data-ko-container="main">` marks the draggable area as well as the draggable blocks.

2. open `index.html` and add an entry

```

var viewModel = {
  showSaved: ko.observable(false),
  edits: ko.observableArray(initialEdits),
  templates: [{
    name: 'versafix-1', desc: 'The versatile template'
  },{
    name: 'tedc15', desc: 'The TEDC15 template'
  },
  {
    name: 'tutorial', desc: 'Mosaico-tutorial'
  }
  ]
};

```

3. create the previews

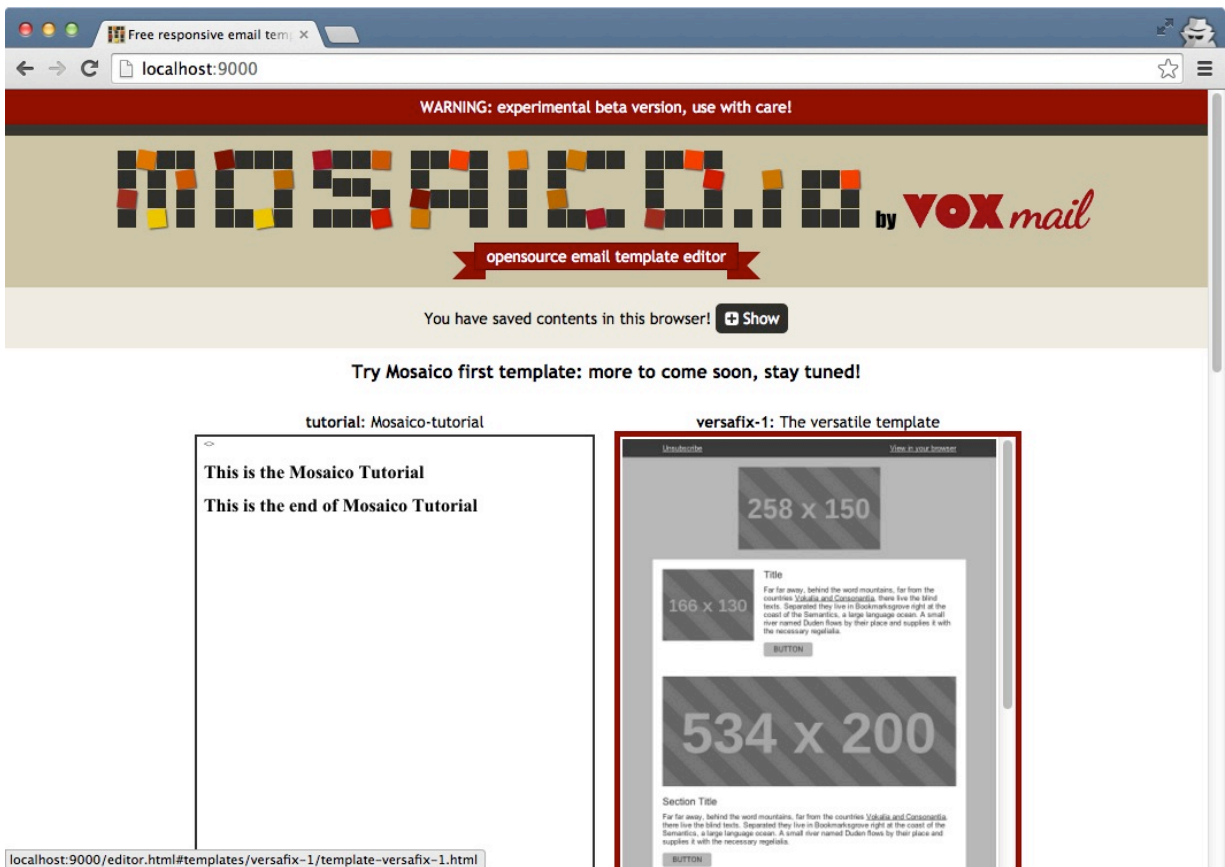
```
node_modules/grunt-cli/bin/grunt makeThumbs
```

4. start the local server

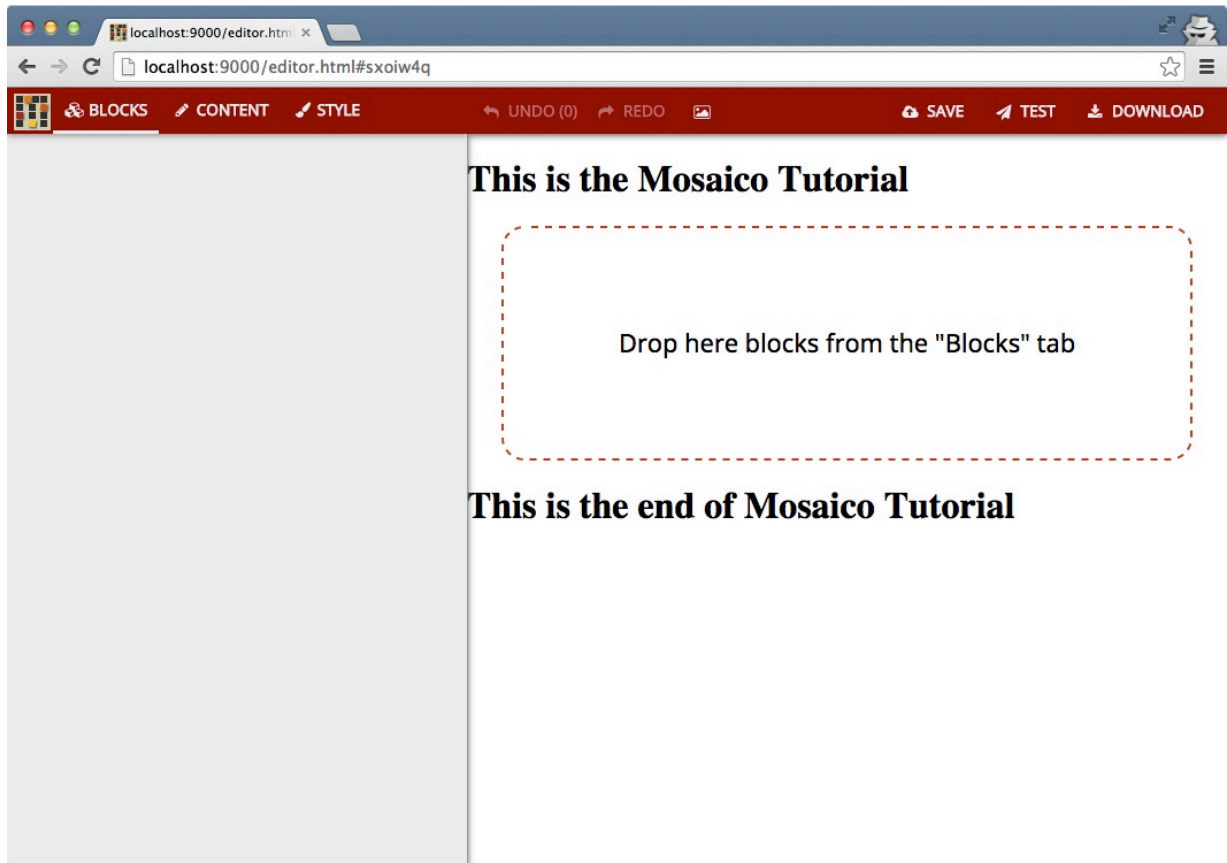
```
node_modules/grunt-cli/bin/grunt
```

5. Visit localhost:9000

you see the selection of templates with previews



6. click on "tutorial" in order to create an email which is an instance of "tutorial". In other words, the new email is based on the template "tutorial"



### 3 Create a draggable blocks

1. add to

```
<div data-ko-container="main">

  <div data-ko-block="HeaderAndText">
    <h1>this is header</h1>
    <div>
      this is text
    </div>
  </div>

</div>
```

With this we have one draggable block called "HeaderAndText"

2. recreate the previews

```
node_modules/grunt-cli/bin/grunt makeThumbs
```

3. start the local server again

```
node_modules/grunt-cli/bin/grunt
```

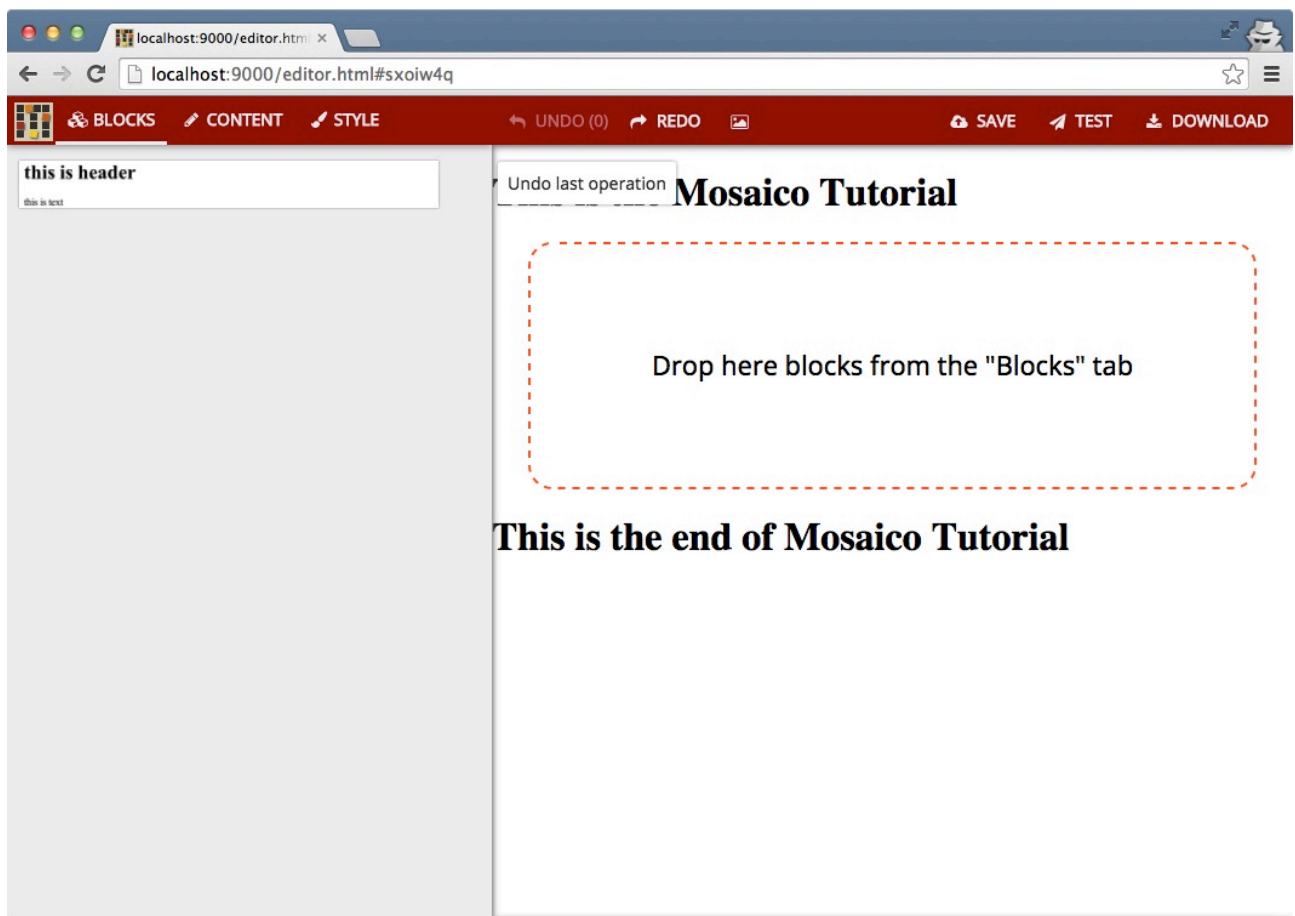


Figure 1: draggable

## 4 specify editable content

1. create the container for the mosaico classes. Mosaico classes ...

```
<head>
  <meta charset="UTF-8">
  <title></title>

  <style type="text/css">
    @supports
      -ko-blockdefs {

        text {
          widget: text
        }
      }

  </style>
</head>
```

A mosaico class is an entity which defines values applicable for html attributes, css properties, mosaico configuration.

In this case in `-ko-blockdefs` we have a mosaico class named "text". The attribute "widget" says that this is a text

2. enhance the draggable block with markup for editable text

```
<div data-ko-block="HeaderAndText">
  <h1 data-ko-editable="text">this is header</h1>

  <div data-ko-editable="text">
    this is text
  </div>
</div>
```

We have marked to locations as editable by the HTML-attribute 'data-ko-editable'. The value of this attribute refers to a mosaico class ("text").

3. Hit refresh on your browser

Now the header and the body are editable. If you click on it, you get a text editor. Note that if you change the header, the body changes as well. The reason is that the mosaico class "text" is referenced in both elements. In fact both elements refer to the same instance of "text".

4. to overcome this change the data-ko-editable attributes as follows:

```
<div data-ko-block="HeaderAndText">
  <h1 data-ko-editable="headerText">this is header</h1>

  <div data-ko-editable="bodyText">
    this is text
  </div>
</div>
```

This notation creates two different instances of the mosaic class "text". This is an implicit mechanism called "magic autodeprefix" so when you try to use a variable named "headerText" mosaico will try to see if you defined it as "headerText" (in the @supports section) and if it doesn't find it, it will look for "text" (camel-case deprefixing) and so on.

In our case we only have "text" so "headerText" and "bodyText" use the mosaico class "text".

5. refresh your browser to see the results. You can now edit the header and the body text independent from each other.

you also observe that header text is only a single line while the body text can be paragraphs etc. Reason is that mosaico derives the editing capabilities from the related html element:

At this time data-ko-editable autodetect a 2 kind of editable areas.

- If you apply it to `<TD>` or `<DIV>` then you get a “multiline” editor (full toolbar)
- If, instead, you apply it to another element, then you get a simpler editor thought for single line.