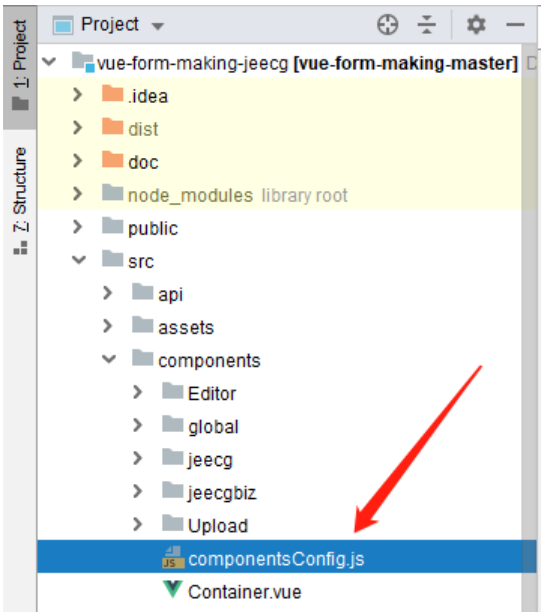
用代码方式新增一个自定义组件的方法

第一步：显示在拖动候选栏里

首先打开 `src/components/componentsConfig.js` 文件，所有组件的基础配置都是在这个文件里写的。



在这个文件里新增一段固定格式的JSON，包含新组件的信息

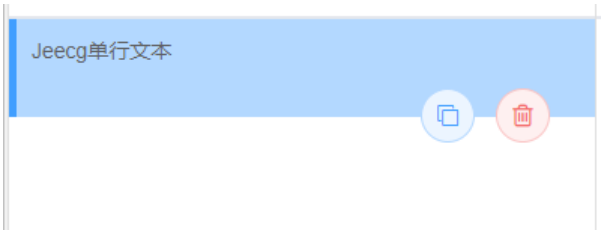
```
export const basicComponents = [
  {
    type: 'jeecg-input', // 组件的唯一 type
    name: 'Jeecg单行文本', // 组件的标题
    className: 'form-jeecg-input', // 默认类名
    icon: 'icon-input', // 图标，目前必须用存在 iconfont 中的图标
    options: { // 组件的选项，用户可修改的
      width: '100%',
      defaultValue: '',
      placeholder: '',
      readonly: false,
      disabled: false,
    }
  },
  {
    type: 'input',
    name: '单行文本',
    className: 'form-input',
    icon: 'icon-input',
    options: {
      width: '100%',
      defaultValue: '',
      required: false,
      dataType: 'string',
      pattern: '',
      placeholder: '',
      readonly: false,
      disabled: false,
    }
  },
```

新增组件的JSON

`options` 里的属性根据组件的要求按需整改。

保存后打开页面就可以发现已经添加到 **基础字段** 里了

虽然可以拖动到设计器中，但是不会有任何显示，因为我们没有定义组件的实现



第二步：编写组件实现

首先新建一个vue文件： `src/components/jeecg/JeecgInput.vue` ，暂时先写上下图的这些代码

JeecgInput.vue

```vue
<template>
  <label>
    <input/>
  </label>
</template>

<script>
  export default {
    name: 'JeecgInput',
    props: []
  }
</script>

<style scoped>

</style>
```

然后在 `src/components/WidgetFormItem.vue` 文件里引用一下

```
import JeecgInput from './jeecg/JeecgInput'


export default {
  name: 'WidgetFormItem',
  mixins: [WidgetDraggable],
  props: ['element', 'select', 'index', 'data'],
  components: {
    JeecgInput,
```

并且在页面里同步引用，加个 v-if 判断，只有当当前组件是 jeecg-input 的时候才显示

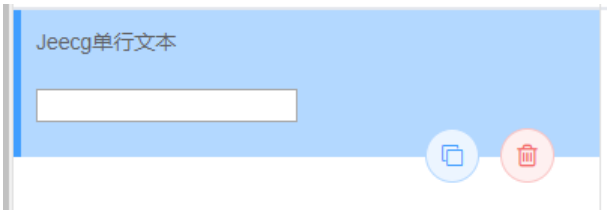
```
        <template v-if="element.type==ctypes.fileUpload">
          <j-file-upload
              :element="element"
              v-model="element.options.defaultValue"
              :readOnly="element.options.disabled"
          />
        </template>

        <template v-if="element.type=='jeecg-input'">
          <jeecg-input></jeecg-input>
        </template>
        <!-- update-end--Author:sunjianlei Date:20190613 for：新增自定义组件 -->

        <el-button title="删除" @click.stop="handleWidgetDelete(index)" class=
          <!-- <icon name="icon-trash" style="width: 12px;height: 12px;"></ic
          <i class="iconfont icon-trash"></i>
        </el-button>
        <el-button title="复制" @click.stop="handleWidgetClone(index)" class="
          <!-- <icon name="icon-clone" style="width: 12px;height: 12px;"></ic
          <i class="iconfont icon-clone"></i>
        </el-button>

      </el-form-item>
```

再回到页面上，就可以看到能正常显示出来了



但是点击预览仍然是显示不出来组件的，因为设计和预览用的是两个不同的组件

我们还需要打开 `src/components/GenerateFormItem.vue` ，用通用的方式再引用一下刚刚新建的组件

```
import JeecgInput from './jeecg/JeecgInput'

export default {
  name: 'GenerateFormItem',
  props: ['widget', 'models', 'rules', 'remote', 'r
  components: {
    JeecgInput,
```

第三步：用户自定义组件属性

组件的属性可以在 `src/components/WidgetConfig.vue` 文件里整改，打开这个文件，新增以下代码

```
<template v-if="data.type === 'jeecg-input'">
  <h1>这里是新增的属性</h1>
</template>
```

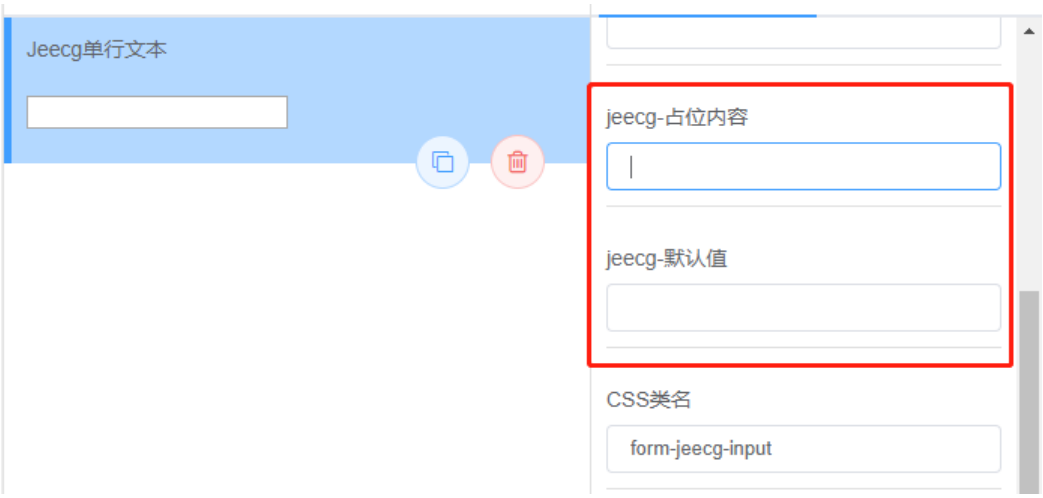
回到页面里就可以看到效果

现在我们给 `defaultValue` 和 `placeholder` 这两个属性开放给用户修改

```
options: { // 组件的选项，用户可修改的
  width: '100%',
  defaultValue: '',
  placeholder: '',
  readonly: false,
  disabled: false,
}
```

```
<template v-if="data.type === 'jeecg-input'">
  <el-form-item label="jeecg-占位内容">
    <el-input v-model="data.options.placeholder"/>
  </el-form-item>
  <el-form-item label="jeecg-默认值">
    <el-input v-model="data.options.defaultValue"/>
  </el-form-item>
</template>
```

效果如下：



还需要组件内部配合修改下，修改成下图这样

JeecgInput.vue

```
<template>
  <label>
    <input
        v-model="value"
        :placeholder="options.placeholder"
    />
  </label>
</template>

<script>
  export default {
    name: 'JeecgInput',
    props: ['value', 'options']
  }
</script>

<style scoped>

</style>
```

在 `src/components/WidgetFormItem.vue` 和 `src/components/GenerateFormItem.vue` 组件里也要修改一下传值

```
<template v-if="element.type=='jeecg-input'">
  <jeecg-input
      v-model="element.options.defaultValue"
      :options="element.options"
  ></jeecg-input>
</template>

<template v-if="widget.type=='jeecg-input'">
  <jeecg-input
      v-model="dataModel"
      :options="widget.options"
  ></jeecg-input>
</template>
```

点击预览就可以看到效果了

上一篇：上传组件配置

下一篇：基础操作手册