链滴

Base64 格式图片插入 DOM 卡顿解决方案

作者：zjhch123
原文链接：https://ld246.com/article/1574784240506
来源网站：链滴

Base64格式图片插入DOM卡顿解决方案

背景

在某些场景下我们会遇到将一张被转为base64编码的图片插入到DOM树中的需求。例如以下操作：

```
const b64str = 'blablabla'
const img = new Image()
img.src = b64str
document.querySelector(dom).append(img)
```

如果当这个图片很大导致base64编码的字符串过长，在插入DOM的过程中难免会造成UI上的卡顿。其是在当下，各种UI框架盛行，DOM节点乱用滥用使得大部分网站的DOM树过深过复杂，插入一个大的DOM节点势必会影响用户体验。

解决方案

我们可以将base64字符串转为Blob对象，并生成指向该对象的URL。

我们将使用到两个API：

1. fetch(resource, init)
2. URL.createObjectURL(object)

fetch自然不用多说，先简单介绍一下URL.createObjectURL(object)。其参数object类型可为File、Blob、MediaSource。返回值为DOMString类型，代表指向object的一个URL。

返回值：

A DOMString containing an object URL that can be used to reference the contents of the specified source object.

具体实现方案：

```
fetch(b64str).then(data => data.blob()).then((b) => {
  const url = URL.createObjectURL(b)
  const img = new Image()
  img.src = url
  document.querySelector(xxx).append(img)
})
```

可以封装成小函数：

```
function insertB64Image(b64str, targetDOMSelector) {
  return fetch(b64str).then(data => data.blob()).then((b) => {
    const url = URL.createObjectURL(b)
    const img = new Image()
    img.src = url
    document.querySelector(targetDOMSelector).append(img)
  })
}
```

兼容性是**IE11**以上的浏览器可以完美使用，也就是说现代化浏览器是完全不用担心兼容性问题的。

原理

解决方案的原理非常基础，其为

1. 使用fetch直接请求base64图片资源，将返回值转换为浏览器资源Blob对象；
2. 生成Blob对象的ObjectURL，并将其作为Image对象的src属性；
3. 在DOM指定位置插入Image对象。