

“Designing” with GraphQL

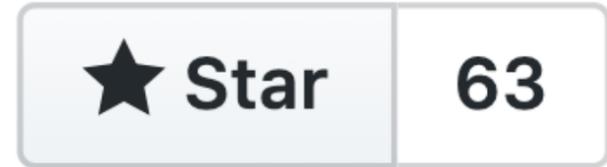
#ReactAdvanced

Hi!

I'm Bernardo

TravelRepublic

@braposo



react-responsive-picture

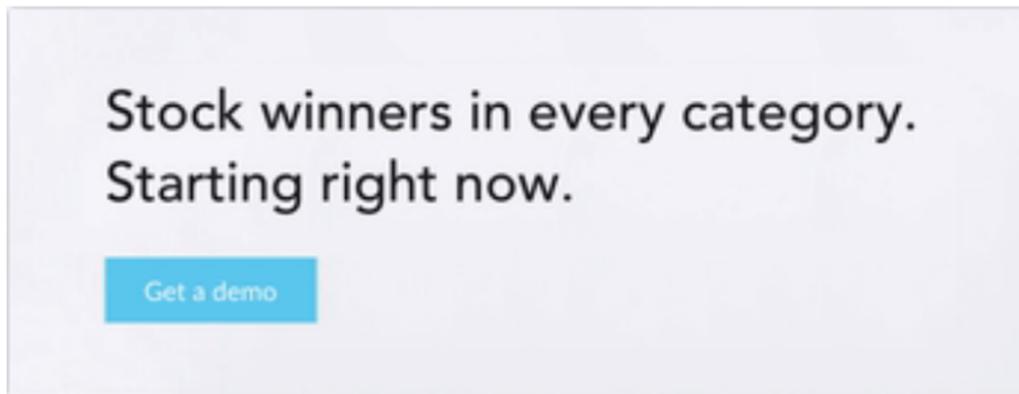
A future-proof responsive image component that supports latest `<picture>` specification. Uses [picturefill](#) for backward compatibility from IE9+.

npm v3.2.2 downloads 8.1k/month gzip size 7.62 kB license MIT PRs welcome

brps.me/rrp

★ Star 216

react-text-loop



An animated loop of text nodes for your headings. Uses [react-motion](#) for the transition so it handles super fast animations and spring params.

npm v2.1.1 downloads 17k/month gzip size 8.07 kB license MIT PRs welcome

brps.me/rtl



GraphQLCSS

★ Star

638

`graphql-css` is a blazing fast CSS-in-GQL™ library that converts GraphQL queries into styles for your components.

Comes with a bunch of utilities so it's easy to integrate with your favourite way of building components.

build **passing** coverage **100%** npm **v2.0.0** downloads **33/month** gzip size **12.8 kB** license **MIT**

module formats **umd, cjs, esm** code style **prettier** PRs **welcome** **Blazing Fast** **Modern** **Enterprise Grade**

brps.me/gqlcss



Jani Eväkallio
@jevakallio

This library by @braposo is going to be a game-changer! Building UIs will never be the same again, my mind is just exploding with all the possibilities this pattern will unlock! What a time to be alive!



Michele Bertoli @MicheleBertoli · 20 Mar 2018
A blazing fast CSS-in-GQL™ library buff.ly/2HObMro



Andrew Clark
@acdrite

You had my curiosity. But now you have my attention.

README.md



GraphQLCSS

graphql-css is a blazing fast CSS-in-GQL™ library that converts GraphQL queries into styles for your components.

Comes with a bunch of utilities so it's easy to integrate with your favourite way of building components.

[View all of README.md](#)



Kitze @thekitze · 19 Mar 2018

Haven't even clicked on the link but I swear if css-in-graphql becomes a thing

I
Am
Out
Of
Webdev
Forever
Bye



I'm so undefined 🤪 @NaveenS16 · 18 Mar 2018

[github.com/braposo/graphql...](https://github.com/braposo/graphql-css)

'graphql-css' is a blazing fast CSS-in-GQL™ library that converts GraphQL queries into styles for your components. #GraphQL #React #CSS #FrontEnd

2

2

27

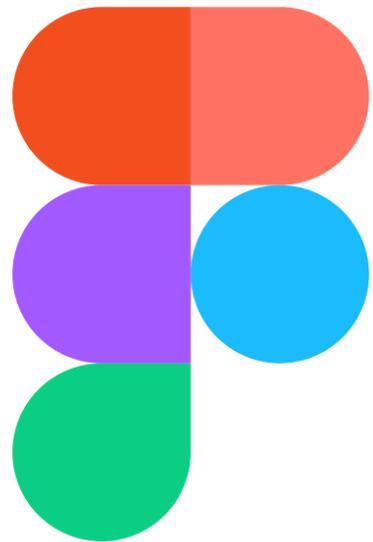


swyx @ReactAdvanced
@swyx

Replying to @thekitze

but its blazing fast.

BLAZING!!



figma.com/developers

Figma API

Multiple REST endpoints

```
GET /v1/files/:key
GET /v1/images/:key

GET /v1/teams/:team_id/projects
GET /v1/projects/:project_id/files

GET /v1/files/:key/comments
POST /v1/files/:key/comments

GET /v1/teams/:team_id/components
GET /v1/teams/:team_id/styles
```

Rigid tree structure

```
{  
  "id": "0:1",  
  "name": "Assets",  
  "type": "CANVAS",  
  "children": [  
    {  
      "id": "20:25",  
      "name": "Box",  
      "type": "COMPONENT",  
      "children": [  
        {  
          "id": "12:5",  
          ...  
        }  
      ]  
    }  
  ]  
}
```

Let's see an example

Get all text styles in a specific frame

The screenshot shows a design tool interface. On the left is a sidebar with a 'Pages' section containing a '+' icon and a list of items: 'forms', 'controls', 'icons', and 'styles' (which is checked). Below this are sections for '# grids' and '# typography'. The typography section lists 'content', 'h4', 'h3', 'h2', 'h1', and 'caption', each with a dashed box icon. On the right is a preview window titled 'typography' showing a grid of text styles: 'H1 desktop', 'H1 tablet', 'H1 mobile', 'H2 desktop', 'H2 tablet', 'H2 mobile', 'H3 desktop', 'H3 tablet', 'H3 mobile', 'H4 desktop', and 'H4 mobile'. Below the grid are three caption styles: 'CAPTION LARGE', 'CAPTION MEDIUM', and 'caption small'. A vertical axis on the left of the preview window shows pixel values from -750 to 500. A zoomed-in view of a text element is shown above the preview window, with a green highlight on the left and a purple highlight on the right.

Get all text styles in a specific frame

- 1. Fetch file data**
- 2. Get the styles page**
- 3. Get the typography frame**
- 4. Extract font styles from each group**

```
1  async function getFontStyles(figmaId, figmaApiKey) {
2    // 1 - Fetch File info
3    const result = await fetch("https://api.figma.com/v1/files/" + figmaId, {
4      method: "GET",
5      headers: {
6        "X-Figma-Token": figmaApiKey
7      }
8    });
9    const figmaTreeStructure = await result.json();
10
11   // 2 - Get styles page
12   const stylesPage = figmaTreeStructure.document.children.filter(item => {
13     return item.name === "styles";
14   })[0].children;
15
16   // 3 - Get font styles artboard
17   const fontStylesArtboard = stylesPage.filter(item => {
18     return item.name === "typography";
19   })[0].children;
20
21   // 4 - Go through each group to extract font styles
22   const fontStyles = fontStylesArtboard.map(fontGroup => {
23     return {
24       name: fontGroup.name,
25       styles:
26         fontGroup.children &&
27         fontGroup.children.reduce((groupStyles, subFontItem) => {
28           return {
29             ...groupStyles,
30             [subFontItem.name]: {
31               family: `${subFontItem.style.fontFamily}`,
32               size: `${subFontItem.style.fontSize}px`,
33               weight: subFontItem.style.fontWeight,
34               lineHeight: `${subFontItem.style.lineHeightPercent}%`,
35               spacing:
36                 subFontItem.style.letterSpacing !== 0
37                   ? `${subFontItem.style.letterSpacing}px`
38                   : "normal"
39             }
40           };
41         }, {})
42     };
43   });
44
45   return fontStyles;
46 }
```

**Get all text
styles in a
specific
frame**

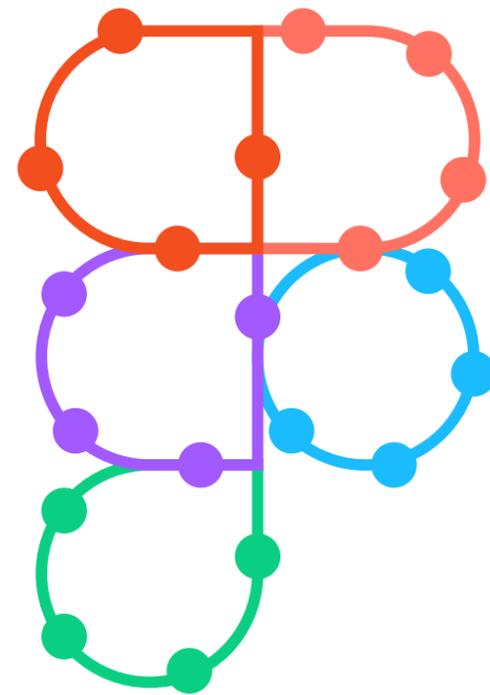
```
[  
  {  
    "name": "caption",  
    "styles": {  
      "large": {  
        "family": "Arimo",  
        "size": "16px",  
        "weight": 400,  
        "lineheight": "140%",  
        "spacing": "3.2px"  
      },  
      ...  
    }  
  }  
  ...  
]
```

**What if we could use
something simpler?**

**Get all text
styles in a
specific
frame**

```
file(id: "fileID") {  
  pages(name: "styles") {  
    frames(name: "typography") {  
      groups {  
        name  
        texts {  
          name  
          styles {  
            fontSize  
            fontFamily  
            fontWeight  
          }  
        }  
      }  
    }  
  }  
}
```

It's just GraphQL™



figma-graphql

figma-graphql.party

Let's see it working!

Watch the full talk at

brps.me/gqlconf19

Special thanks:

Sara
@NikkitaFTW

Andrey
@okonetchnikov

Jon
@jongold

Thank you!

@braposo