Rethinking the Figma API

#FigmaLondon

Hi!
I’m Bernardo

TravelRepublic
@braposo

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
```

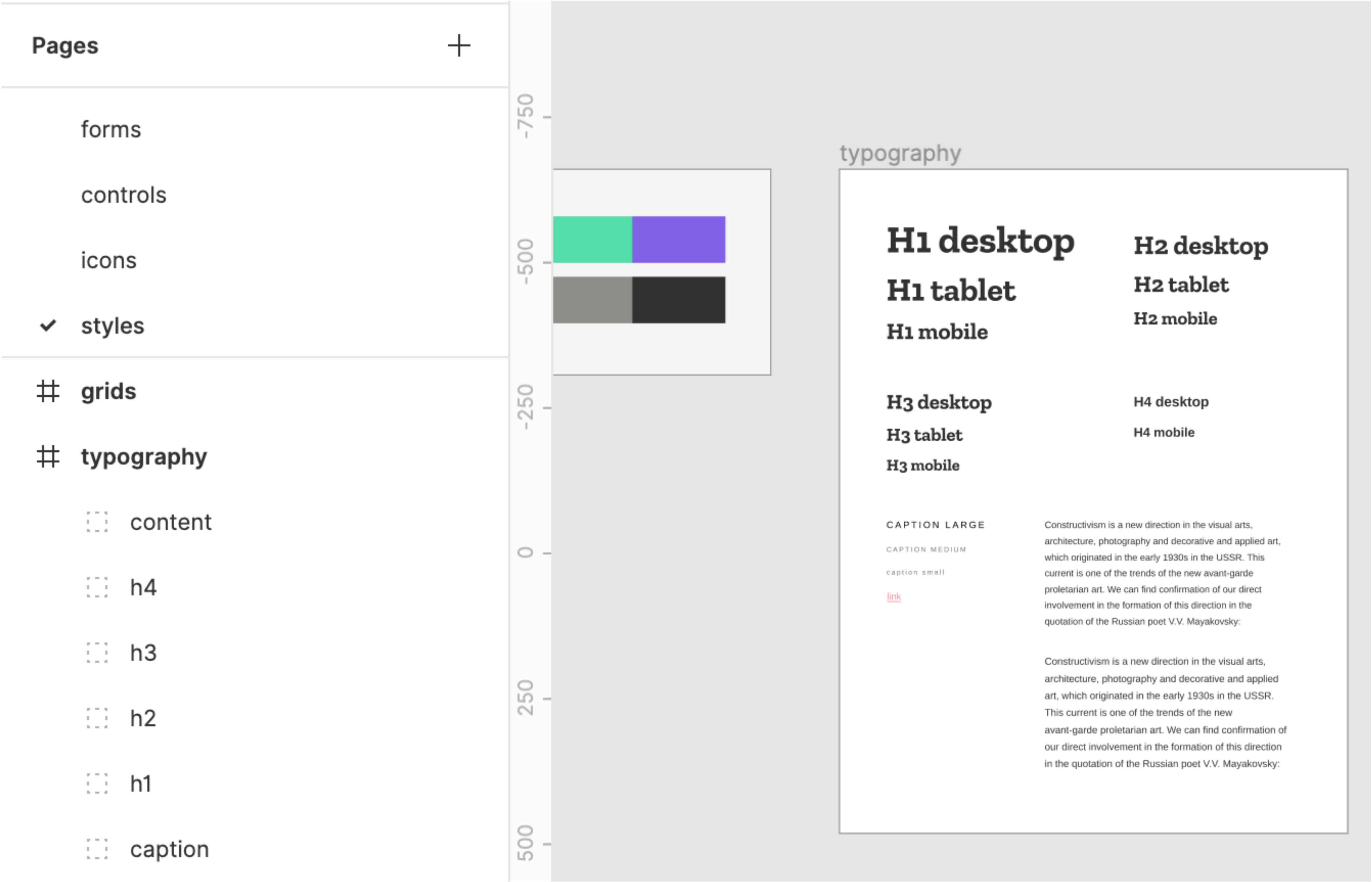
Rigid file structure

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

API is built for machines

Let’s see an example

Get all text styles in a specific frame



Pages +

- forms
- controls
- icons
- ✓ styles

- # grids
- # typography
  - content
  - h4
  - h3
  - h2
  - h1
  - caption

typography

H1 desktop

H1 tablet

H1 mobile

H2 desktop

H2 tablet

H2 mobile

H3 desktop

H3 tablet

H3 mobile

H4 desktop

H4 mobile

CAPTION LARGE

CAPTION MEDIUM

caption small

link

Constructivism is a new direction in the visual arts, architecture, photography and decorative and applied art, which originated in the early 1930s in the USSR. This current is one of the trends of the new avant-garde proletarian art. We can find confirmation of our direct involvement in the formation of this direction in the quotation of the Russian poet V.V. Mayakovsky:

Constructivism is a new direction in the visual arts, architecture, photography and decorative and applied art, which originated in the early 1930s in the USSR. This current is one of the trends of the new avant-garde proletarian art. We can find confirmation of our direct involvement in the formation of this direction in the quotation of the Russian poet V.V. Mayakovsky:

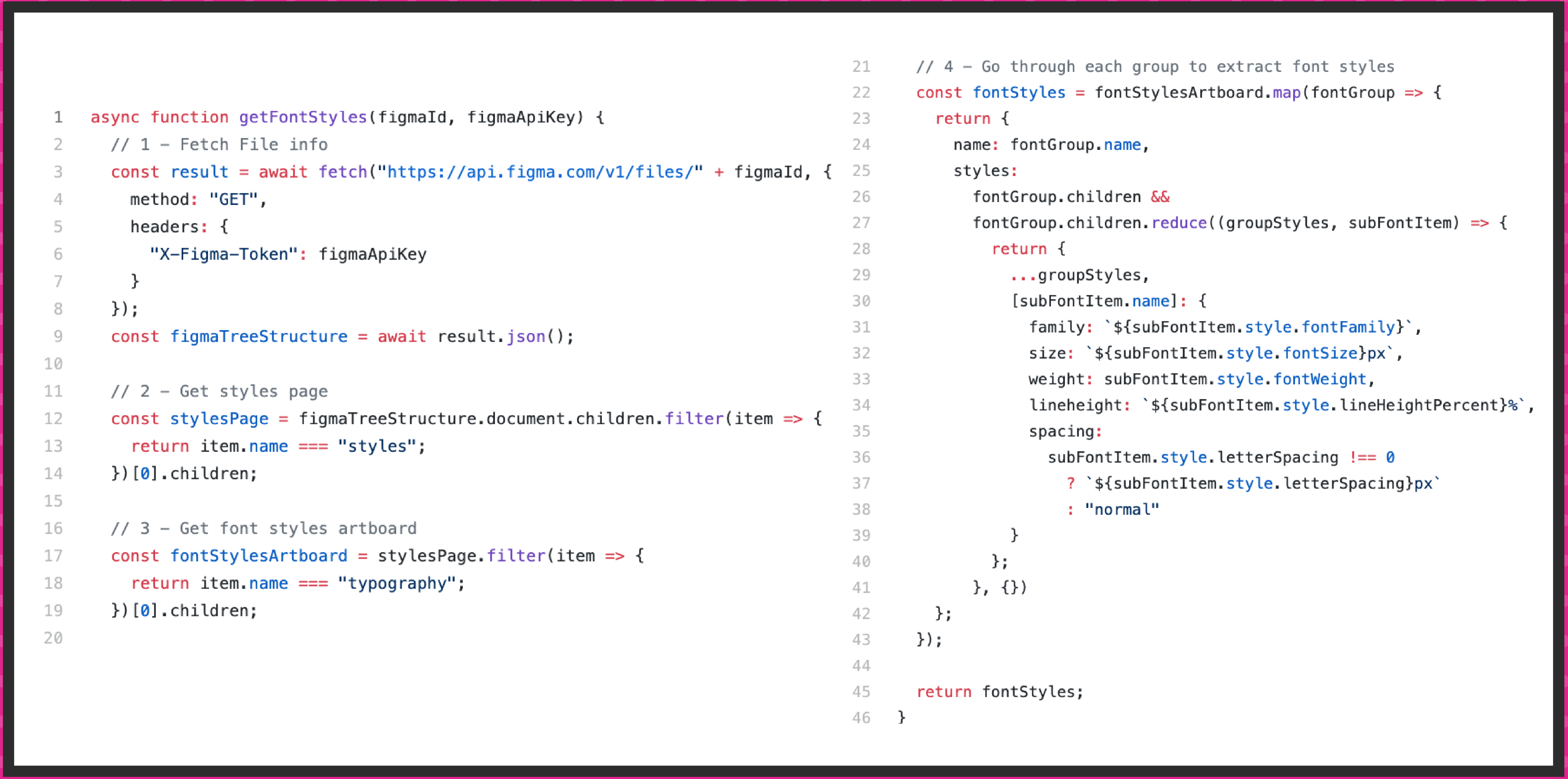
Get all text styles in a specific frame

1. Fetch file data

2. Get the styles page

3. Get the typography frame

4. Extract font styles from each group



```
async function getFontStyles(figmaId, figmaApiKey) {
  // 1 – Fetch File info
  const result = await fetch("https://api.figma.com/v1/files/" + figmaId, {
    method: "GET",
    headers: {
      "X-Figma-Token": figmaApiKey
    }
  });
  const figmaTreeStructure = await result.json();

  // 2 – Get styles page
  const stylesPage = figmaTreeStructure.document.children.filter(item => {
    return item.name === "styles";
  })[0].children;

  // 3 – Get font styles artboard
  const fontStylesArtboard = stylesPage.filter(item => {
    return item.name === "typography";
  })[0].children;

  // 4 – Go through each group to extract font styles
  const fontStyles = fontStylesArtboard.map(fontGroup => {
    return {
      name: fontGroup.name,
      styles:
        fontGroup.children &&
        fontGroup.children.reduce((groupStyles, subFontItem) => {
          return {
            ...groupStyles,
            [subFontItem.name]: {
              family: `${subFontItem.style.fontFamily}`,
              size: `${subFontItem.style.fontSize}px`,
              weight: subFontItem.style.fontWeight,
              lineheight: `${subFontItem.style.lineHeightPercent}%`,
              spacing:
                subFontItem.style.letterSpacing !== 0
                  ? `${subFontItem.style.letterSpacing}px`
                  : "normal"
            }
          };
        }, {})
    };
  });

  return fontStyles;
}
```

Get all text styles in a specific frame

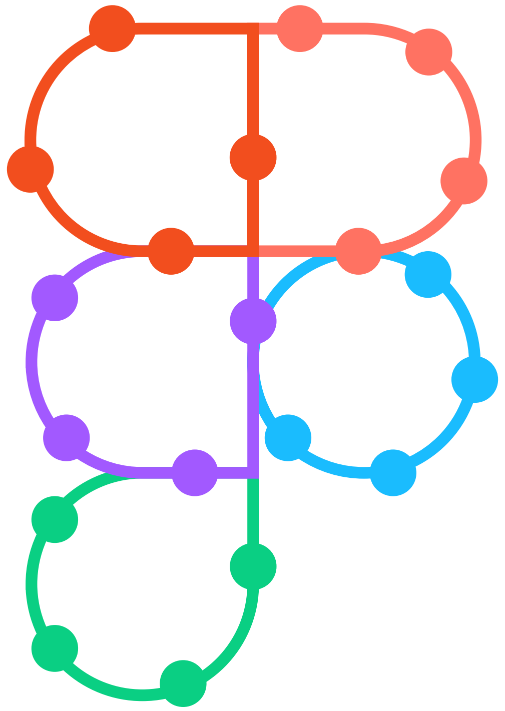
```
[
  {
    "name": "caption",
    "styles": {
      "large": {
        "family": "Arimo",
        "size": "16px",
        "weight": 400,
        "lineheight": "140%",
        "spacing": "3.2px"
      },
      ...
    }
  }
...
```

What if we could use something simpler?

Get all text styles in a specific frame

```
file(id: "fileID") {
  frames(name: "typography") {
    groups {
      name
      texts {
        name
        style {
          fontSize
          fontFamily
          ...
        }
      }
    }
  }
```

It’s just GraphQL™



figma-graphql

figma-graphql.party

Human-first API

Let’s see it working!

Why should you use it?

Shortcuts

Unified queries

Extended information

Special thanks:

Sara
@NikkitaFTW

Andrey
@okonetchnikov

Jon
@jongold

Thank you!

@braposo