Expanding what's possible on the Web: Project Fugu

Thomas Steiner
@tomayac
Have you been there?

This PNG should really be an SVG!

The usual solution
The usual solution (continued)
The usual solution (continued)
There’s got to be a better way
Introducing SVGcode (svgco.de)
What SVGcode does differently

**PWA**

Offline-enabled PWA, runs entirely in the client. No server to trust.

Posterizes images and runs potrace per color, then reassembles the fragments for the resulting image.

Exposes all of potrace’s options, including expert options.

**SVG**

Optimizes SVGs via svgo.
Installable, offline-enabled PWA

SVGcode is an installable Progressive Web App and therefore fully offline enabled.

The app is based on the [Vanilla JS template](#) for [Vite.js](#) and uses the popular [Vite plugin PWA](#), which creates a service worker that uses [Workbox.js](#) under the hood.
Window Controls Overlay API

To maximize the available screen real estate, SVGcode uses **Window Controls Overlay** customization by moving its main menu up into the titlebar area.
const WINDOW_CONTROLS_OVERLAY = 'window-controls-overlay';

const onGeometryChange = () => {
    const visible = navigator.windowControlsOverlay.visible;
    menu.classList.toggle(WINDOW_CONTROLS_OVERLAY, visible);
    main.classList.toggle(WINDOW_CONTROLS_OVERLAY, visible);
};

navigator.windowControlsOverlay.addEventListener('geometrychange',
    debounce(async () => {
        onGeometryChange();
    }, 250),
);
File System Access API

SVGcode uses the File System Access API via the `browser-fs-access` ponyfill.

On supporting browsers, it can store a serialized `FileSystemFileHandle` to let the user continue where they left off after a reload.

This also works with the Drag and Drop API.
document.addEventListener('drop', async (event) => {
    event.preventDefault();

    const item = event.dataTransfer.items[0];
    if (item.kind === 'file') {
        inputImage.addEventListener('load', () => {
            URL.revokeObjectURL(blobURL);
        }, { once: true });

        const handle = await item.getAsFileSystemHandle();
        if (handle.kind !== 'file') return;
        const file = await handle.getFile();
        const blobURL = URL.createObjectURL(file);
        inputImage.src = blobURL;
        await set(FILE_HANDLE, handle);
    }
});
Async Clipboard API

SVGcode is also fully integrated with the operating system's clipboard via the Async Clipboard API.

You can paste images from the operating system's file explorer into the app either by clicking the **Paste Image** button or by pressing `command/control + v` on your keyboard.
Async Clipboard API (continued)

The Async Clipboard API has recently gained the ability to deal with SVG images as well, so you can also copy an SVG image and paste it into another application for further processing.
copyButton.addEventListener('click', async () => {
    let svg = svgOutput.innerHTML;
    showToast(i18n.t('optimizingSVG'), Infinity);
    svg = await optimizeSVG(svg);
    const textBlob = new Blob([svg], {type: 'text/plain'});
    const svgBlob = new Blob([svg], {type: 'image/svg+xml'});
    navigator.clipboard.write([new ClipboardItem({
        [svgBlob.type]: svgBlob,
        [textBlob.type]: textBlob,
    })],
    );
    showToast(i18n.t('copiedSVG'));
});
File Handling API

As an installed PWA, SVGcode can become a file handler, or even the default file handler, for image files.

This means image files can be opened in SVGcode right from the operating system’s file explorer.
window.launchQueue.setConsumer(async (launchParams) => {
  if (!launchParams.files.length) return;
  for (const handle of launchParams.files) {
    const file = await handle.getFile();
    if (file.type.startsWith('image/')) {
      const blobURL = URL.createObjectURL(file);
      inputImage.addEventListener('load', () => {
        URL.revokeObjectURL(blobURL);
      }, {once: true});
      inputImage.src = blobURL;
      await set(FILE_HANDLE, handle);
      return;
    }
  }
});
Web Share API

SVGcode can share SVGs via the Web Share API, for example, if you want to send the SVG to someone via email.
shareSVGButton.addEventListener('click', async () => {
  let svg = svgOutput.innerHTML;
  svg = await optimizeSVG(svg);
  const suggestedFileName = getSuggestedFileName(await get(FILE_HANDLE)) || 'Untitled.svg';
  const file = new File([svg], suggestedFileName, { type: 'image/svg+xml' });
  const data = {
    files: [file],
  };
  if (navigator.canShare(data)) {
    try {
      await navigator.share(data);
    } catch (err) {
      console.error(err.name, err.message);
    }
  }
});
Web Share Target API

In the opposite direction, SVGcode can also serve as a share target, so when you have an image file, you can share it to SVGcode.
"share_target": {
  "action": "https://svgco.de/share-target/",
  "method": "POST",
  "enctype": "multipart/form-data",
  "params": {
    "files": [
      {
        "name": "image",
        "accept": ["image/jpeg", "image/png", "image/webp", "image/gif"]
      }
    ]
  }
}
self.addEventListener('fetch', (fetchEvent) => {
    if (fetchEvent.request.url.endsWith('/share-target/') && fetchEvent.request.method === 'POST') {
        return fetchEvent.respondWith(
            (async () => {
                const formData = await fetchEvent.request.formData();
                const image = formData.get('image');
                const keys = await caches.keys();
                const mediaCache = await caches.open
                    (keys.filter((key) => key.startsWith('media'))[0],
                    );
                await mediaCache.put('shared-image', new Response(image));
                return Response.redirect('./?share-target', 303);
            })()
        );
    }
});
🌞 Project Fugu API Showcase

Check out SVGcode and many other amazing apps in the Project Fugu API Showcase.

Filter by Fugu API to see what apps using them exist.

Add not yet listed apps to the showcase.
Fugu API Tracker

Keep track of progress the Fugu team makes by regularly looking at the Fugu API Tracker.

See what APIs have shipped, are in origin trial, dev trial, or under consideration.
Expanding what's possible on the Web!

Thomas Steiner
@tomayac

Slides: goo.gle/web-engines-hackfest-fugu