

Remix
Bay Area

Prepare for Remix v2 And more!

With



Thanks to organizers and sponsors

Big up
Andre L. and Mux today!



MUX



Sébastien Morel

CTO at  crystallize



Backend/DevOps developer since 2003



French living in California



Open Source Knight & Package Maintainer



Dad of a 2-year old



Millisecond hunter



Plopix



San Francisco, CA

What's crystallize? **A Complete Commerce Layer**

An **Headless eCommerce Toolbox**. We got your back(end).



Remix

v2 IS COMING



The logo for Remix, featuring a stylized 'R' followed by the word 'emix' in a lowercase, sans-serif font. The letters are filled with a rainbow gradient and have a glowing, neon-like effect.

v2 is actually here!

“ All v2 APIs and behaviors are available in v1 with **Future Flags**. They **can be enabled one at a time** to avoid development disruption [...] upgrading to v2 should be a non-breaking upgrade. ”



What's new?

with Remix Run v2



What's new

Remix Run v2

- Dev Ex enhancement:
 - **HMR** and **Hot Data Revalidation**
 - **New Route Convention**
- **Performance, optimizations, bug-fixes**
- More **consistent naming** and responsibility split server/client
- New **Adapters**
- Built-in PostCSS/Tailwind support
- And many other things are coming

Migration Path to Remix Run v2



Remix Run v2 - Migration Path

Major changes

- File System Route Convention
- Route **headers** and **meta**
- **CatchBoundary** is gone
- **useTransition** has been flattened and renamed to **useNavigation**
- Flatten **useFetcher**, no more *type*
- **Remix CLI v2**

Remix Run v2 - Migration Path

Minor changes

- ***formMethod*** is normalized to **UPPERCASE**
- ***imagesizes*** and ***imagesrcset*** MUST be **camelCase**
- Configuration changes: (full list in the doc)
 - devServerPort -> **dev.port**
 - **serverNodeBuiltinsPolyfill**: Polyfill are not there by default
 - **serverModuleFormat**: goes to **ESM**

Migration Path

Tips and Tricks



```
/** @type {import('@remix-run/dev').AppConfig} */
module.exports = {
  ignoredRouteFiles: ["**/*.json"],
  // appDirectory: "app",
  // assetsBuildDirectory: "public/build",
  // serverBuildPath: "build/index.js",
  // publicPath: "/build/",
  serverModuleFormat: "cjs", // esm will be default in future
  future: {
    v2_dev: true,
    v2_errorBoundary: true,
    v2_headers: true,
    v2_meta: true,
    v2_normalizeFormMethod: true,
    v2_routeConvention: true,
  },
};
```

Tips and Tricks

Routing: Rename all of of them in 2 seconds

Remix Bay Area - Meetup 2023-08-29

```
curl -fsSL "https://gist.github.com/Plopix/58fc3f3be202d9915c466e71077d36a2/raw/migrate-to-v2-  
routing.bash" | bash -s EXISTING_ROUTES_FOLDER NEW_ROUTES_FOLDER
```

Tips and Tricks

Routing: You can play with *remix.config.js*.

```
Remix Bay Area - Meetup 2023-08-29

/** @type {import('@remix-run/dev').AppConfig} */
module.exports = {
  ignoredRouteFiles: ["**/*.jsx"],
  serverModuleFormat: "cjs", // esm will be default in future
  future: {
    v2_dev: true,
    v2_errorBoundary: true,
    v2_headers: true,
    v2_meta: true,
    v2_normalizeFormMethod: true,
    v2_routeConvention: true,
  },
  routes: async (defineRoutes) => {
    // If you need to do async work, do it before calling `defineRoutes`, we use
    // the call stack of `route` inside to set nesting.
    return defineRoutes((route) => {
      // - The first argument is the React Router path to match against
      // - The second is the relative filename of the route handler
      route("/alias/hello/world", "routes/my/specific/thing.tsx");
      // if you want to nest routes, use the optional callback argument
      route("some/:path", "some/route/file.js", () => {
        // - path is relative to parent path
        // - filenames are still relative to the app directory
        route("relative/path", "some/other/file");
      });
    });
  }
};
```

Tips and Tricks

Headers

Headers: Parent Sharing:

It just follows Remix way of thinking

- If a **HeadersFunction** exist, it will use it
 - your responsibility to merge with parent if needed
- If not
 - It will look in the ancestor route to find one.

Tips and Tricks

Meta

Meta: Same logic, *matches* to the rescue

- Merge with what you have in *matches* if needed.

Tips and Tricks

CatchBoundary

```
Remix Bay Area - Meetup 2023-08-29

export function ErrorBoundary() {
  const error = useRouteError();

  // when true, this is what used to go to `CatchBoundary`
  if (isRouteErrorResponse(error)) {
    return (
      <div>
        <h1>Oops</h1>
        <p>Status: {error.status}</p>
        <p>{error.data.message}</p>
      </div>
    );
  }

  // Don't forget to typecheck with your own logic.
  // Any value can be thrown, not just errors!
  let errorMessage = "Unknown error";
  if (isDefinitelyAnError(error)) {
    errorMessage = error.message;
  }

  return (
    <div>
      <h1>Uh oh ...</h1>
      <p>Something went wrong.</p>
      <pre>{errorMessage}</pre>
    </div>
  );
}
```

Let's review some of that live

Simple Demo time ;-)



That's it!
It's too easy!



But I have more
Just for you!



BFF: Backend Frontend Framework

The server is your friend, but you might still want to decouple things

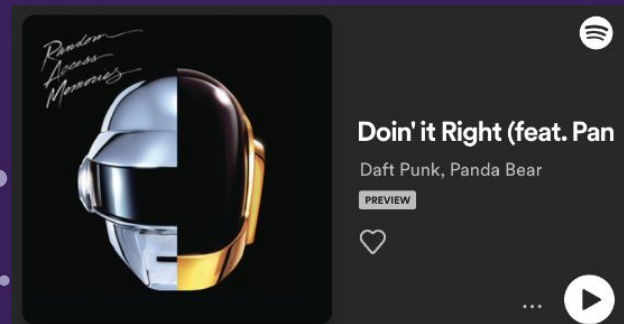
- You want to start ASAP
- You can access your DB from Loaders
- Create API routes

Best architecture is probably to decouple Backend and Frontend

- But that's a lot of hassle
- **You can do it later!**

Doin' It Right, and your team will be dancing!

– Daft Punk

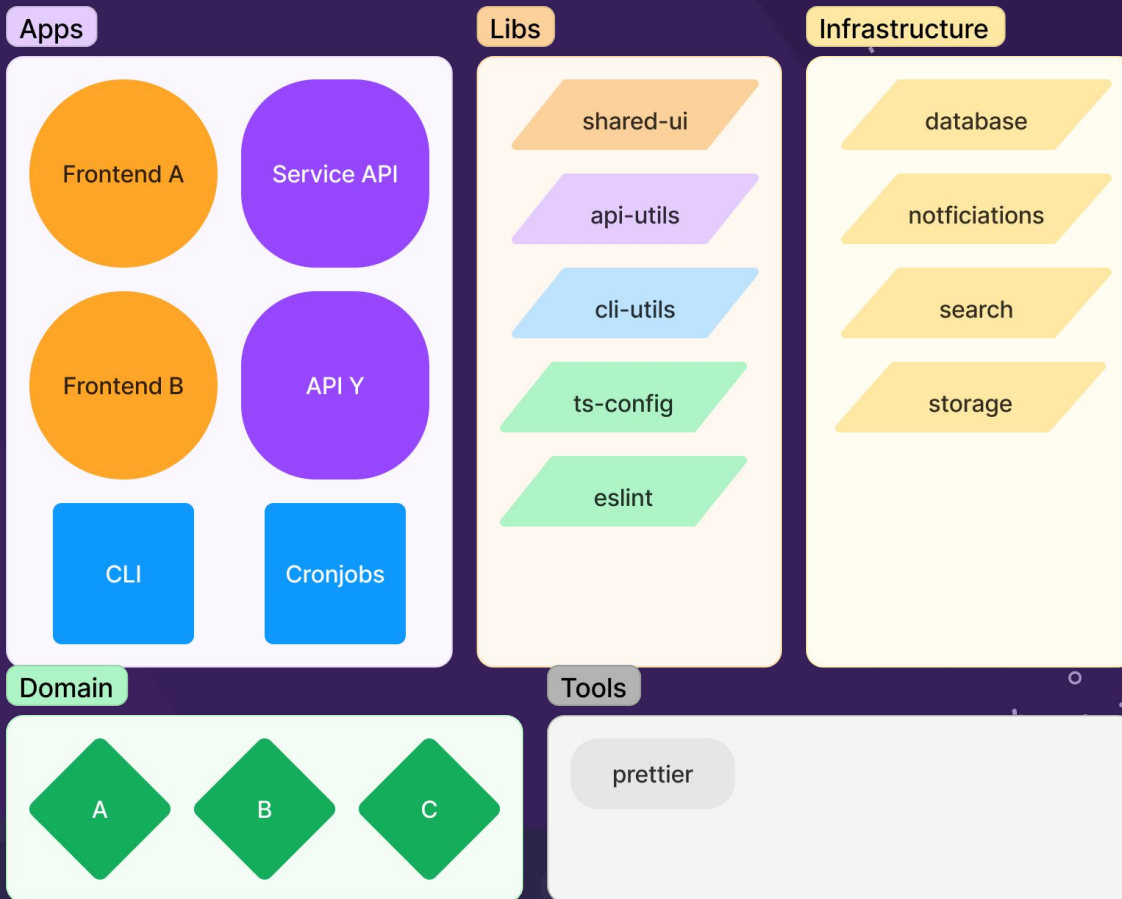


<https://open.spotify.com/track/36c4JohayB9qd64eidQMBi?si=9d229e8d02754f07>



Project Organization

Monorepo



Project Organization

Monorepo

Apps

- Frontend(s)
- Service Layer(s)
- CLI(s)

Domain

- Business logic

Libs

- Generic stuff

Tools

- External tooling

Infrastructure

- Connection layer to other tools/services

Project Organization

Monorepo



PNPM makes it **fast** to **install**



Turborepo makes it **simple** to **run**



Makefile makes it **simple** for your **coworkers**

Project Organization

Monorepo

```
Remix Bay Area - Meetup 2023-08-29

{
  "dependencies": {
    "@crystallize/design-system": "^1.17.3",
    "@crystallize/js-api-client": "^1.12.2",
    "@crystallize/reactjs-components": "^1.4.1",
    // many others
    "@remix-run/node": "^1.19.3",
    "@remix-run/react": "^1.19.3",
    "@remix-run/serve": "^1.19.3",
    // monorepo libs
    "my-domain": "workspace:*",
    "my-infrastructure": "workspace:*",
    "my-shared-ui": "workspace:*",
  },
  "devDependencies": {
    "@remix-run/dev": "^1.19.3",
    "@remix-run/eslint-config": "^1.19.3",
    // many others
    "my-tsconfig": "workspace:*",
    "typescript": "^5.1.6"
  },
  //hidden gem: if you don't know Volta.sh, have a look
  "volta": {
    "node": "18.16.1"
  }
}
```

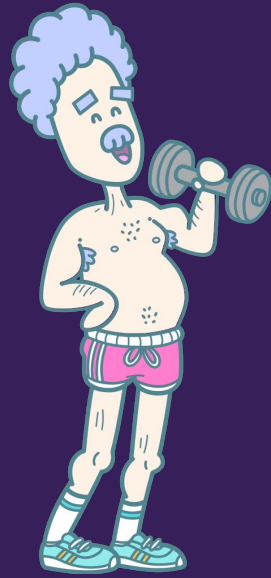
Project Organization

Monorepo

- Watch the build
- Tell Remix that those libs are ESM only

```
Remix Bay Area - Meetup 2023-08-29

/** @type {import('@remix-run/dev').AppConfig} */
module.exports = {
  ignoredRouteFiles: ['**/*.'],
  appDirectory: 'src',
  tailwind: true,
  watchPaths: [
    'src',
    '../../domain/build',
    '../../infrastructure/build',
    '../../libs/shared-ui'],
  serverModuleFormat: 'cjs',
  serverDependenciesToBundle: [
    'my-domain',
    'my-infrastructure',
    'my-shared-ui'
  ],
  // etc/
  future: {
    v2_dev: true,
    v2_errorBoundary: true,
    v2_headers: true,
    v2_meta: true,
    v2_normalizeFormMethod: true,
    v2_routeConvention: true,
  },
};
```



Let's have a look

On a real project





Thanks
Questions?

