Exact Code Splitting

by Dmitry A.

22 October 2020 @ 9am CEST

Dmitry A.

- Founder and CTO at veliovgroup
- Mentor and remote-CTO
- Author of Meteor-files (ostrio:files)
- Author of other packages under ostrio namespace
- Twitter: @smart_egg
- GitHub: @dr-dimitru

Contents

- What are PWA principles
- How to measure the size of the bundle
- Estimate bundle size before exact code splitting
- Utilize routing for dynamic imports
- Showcase `flow-router-extra` usage
- Split codebase
- Estimate bundle size after exact code splitting

What are PWA principles?

- **Progressive** — Progressive enhancement principles using dynamic import(s).
- **Faster after initial loading** — That's why important to reduce "bundle" size
- **Connectivity independent** — Persistent data-storage and cached requests;
- **Fresh** — Meteor's incremental upgrades;
- **Safe** — Served via HTTPS to prevent snooping;
- **Discoverable** — by manifest.json and service worker;
- **Re-engageable** — Ability to use push notifications;
- **Installable** — Provides homescreen icons without the use of an App Store.

How to measure the size of the bundle?

```
meteor --production --extra-packages bundle-visualizer
```

Upgrade routes

To load codebase per route, I will move all static `import`s from the application's bundle to routes definition inside `waitOn()` hook.

Return dynamic `import()` from `waitOn()` hook to wait for dynamically imported codebase to fully load.

Persistent local FilesCollection

```
// CREATE LOCAL COLLECTION BY PASSING null
const _files = new Mongo.Collection(null);
// SET THE SAME NAME AS ON THE SERVER
_files._name = 'uploadedFiles';

const files = new FilesCollection({
  collection: _files // Pass instance of *Mongo.Collection*
});
```

Links

- Docs and snippets repository

Thank you

METEOR

pathable