

Vue.js App Performance Optimization

190228 한현정



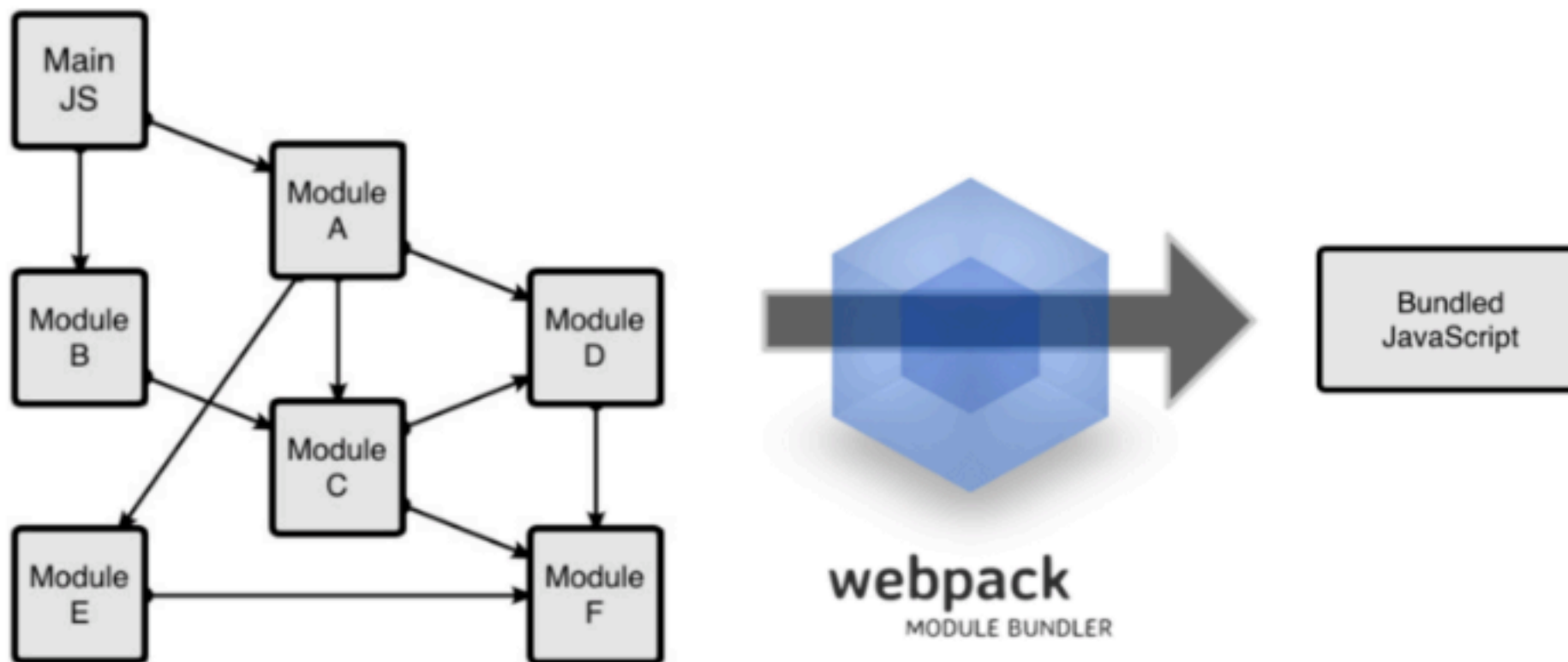
왜?

**느린 로딩 === 적은 사용자
이탈을 고민하는 시간은 단 1초**

느린 로딩 무엇?

app.js가 너무 크네요 (..)

app.js는 왜 커졌을까?

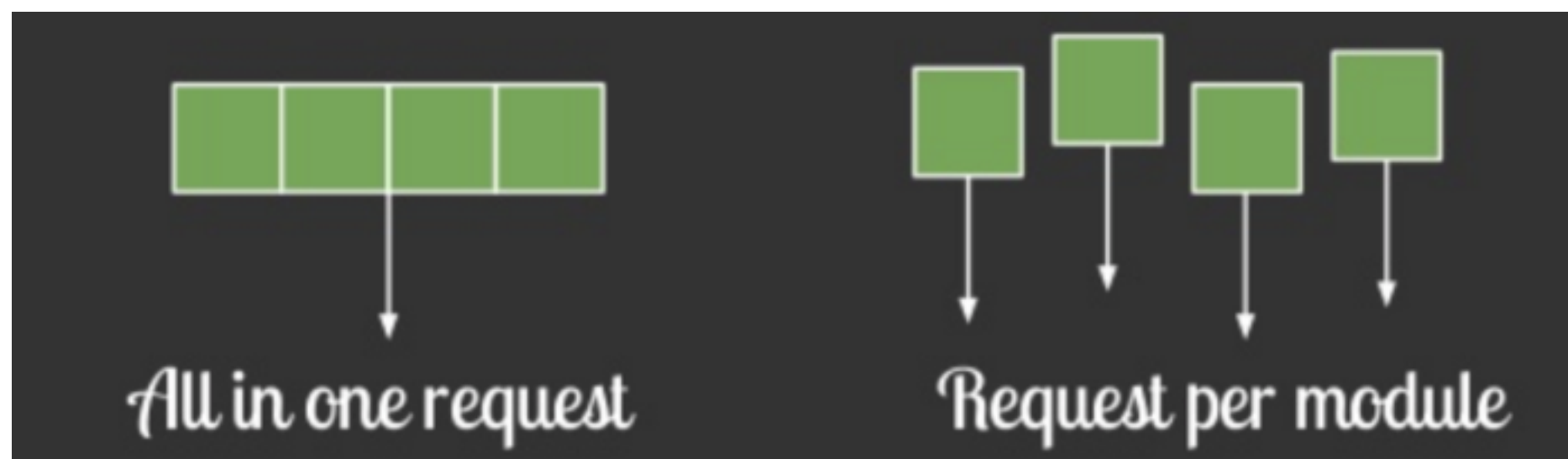


webpack의 dependency graph

Lazy or on demand loading

유저가 정말 필요로 하는 파일만
정말 필요로 할 때 다운받게 하자

: 처음에 모든 파일을 다 번들링해서 가져오지 않아요



Code splitting

how?

: es6 **Dynamic import**를 이용
webpack이 새로운 entry point로 인식,
쪼개서 번들링해줍니다

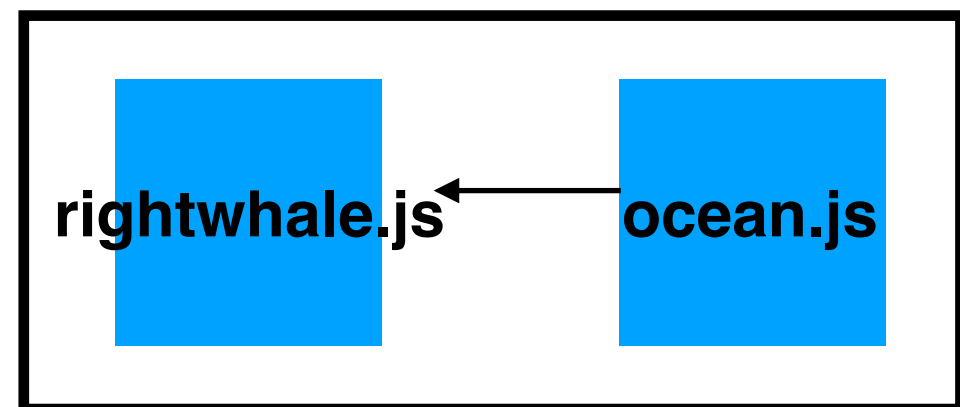
Code splitting

```
//ocean.js
import RightWhale from './whales/rightwhale.js'
const hiBeluga = () => import('./whales/beluga.js');

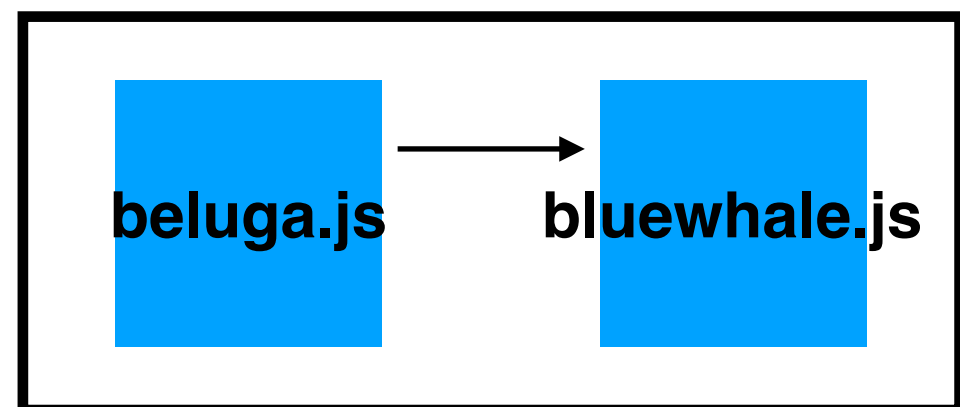
hiBeluga()
  .then(byeBeluga => byeBeluga());

//beluga.js
import BlueWhale from './whales/bluewhale.js'
```

ocean_bundle.js



beluga_bundle.js



Route-based Code splitting

[Home](#) | [About](#) | [Beluga](#) | [Blue Whale](#) | [Humpback Whale](#) | [Killer Whale](#) | [Right Whale](#) | [Sperm Whale](#)



Chasing Whales

For a guide and recipes on how to configure / customize this project, check out the [vue-cli documentation](#).

Installed CLI Plugins

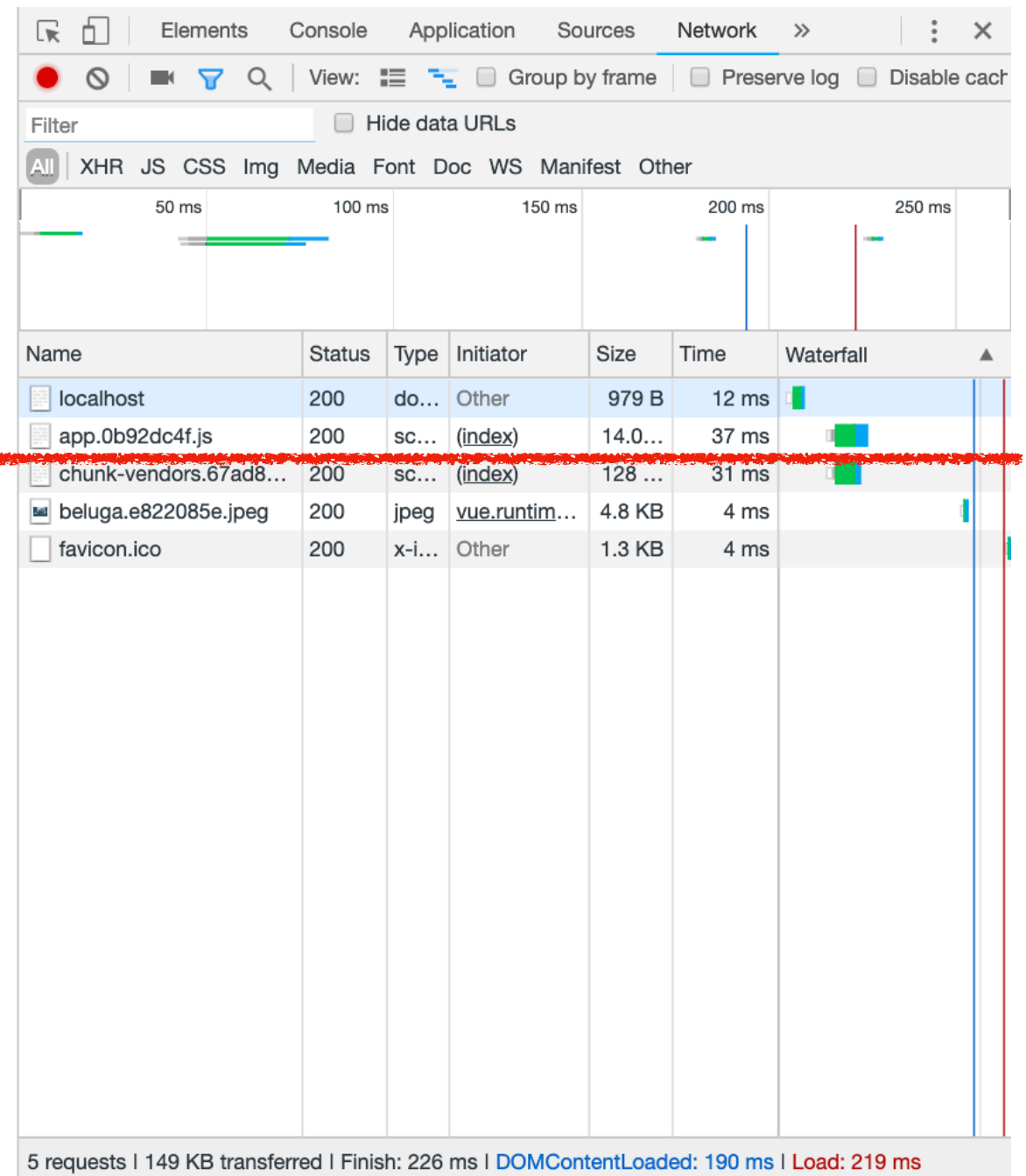
[babel](#) [typescript](#)

Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

Ecosystem

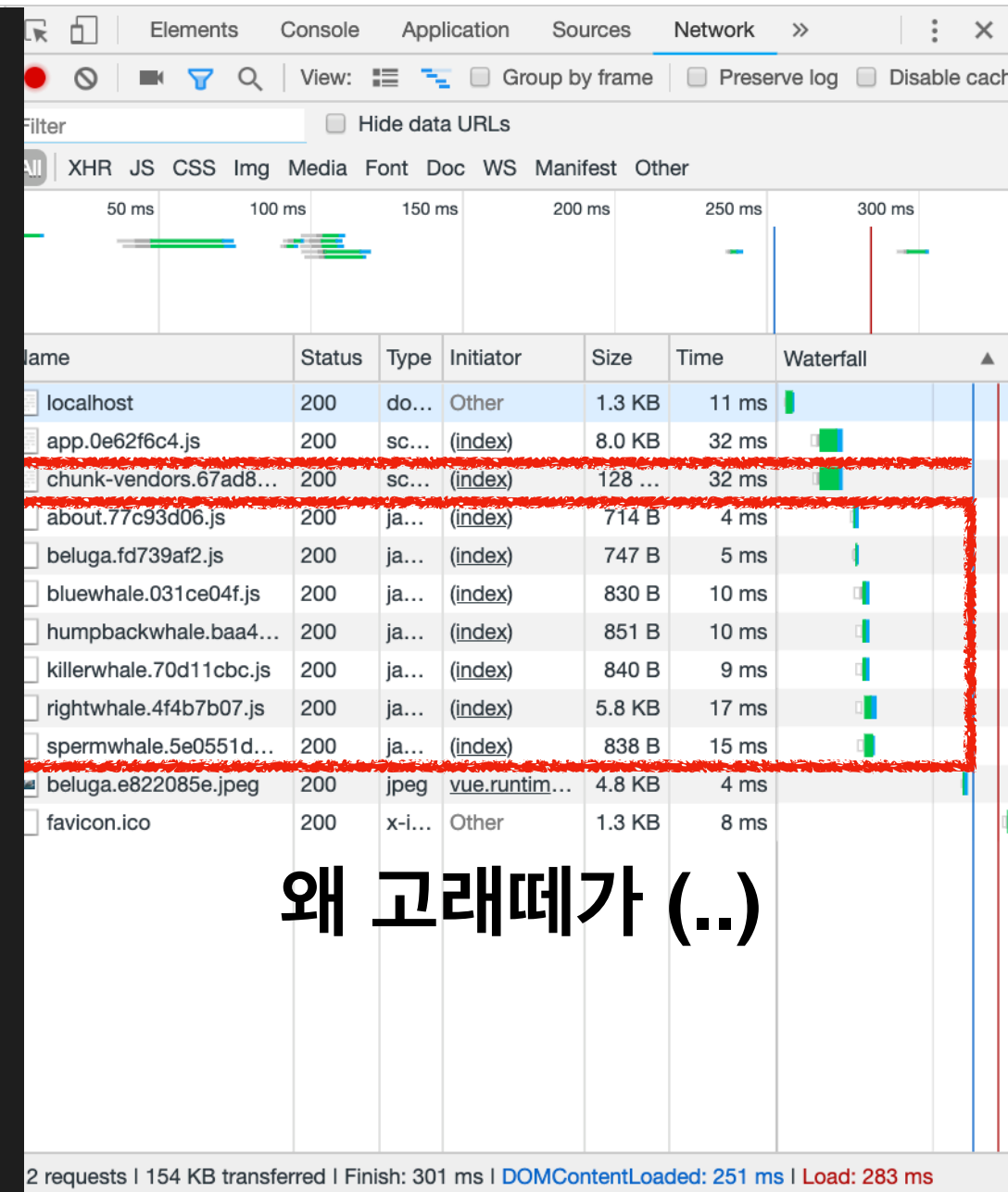
[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)



Route-based Code splitting

```
export default new Router({
  base: process.env.BASE_URL,
  routes: [
    {
      path: '/',
      name: 'home',
      component: Home,
    },
    {
      path: '/about',
      name: 'about',
      component: () => import(/* webpackChunkName: "about" */ './views/About.vue'),
    },
    {
      path: '/beluga',
      name: 'beluga',
      component: () => import(/* webpackChunkName: "beluga" */ './views/Beluga.vue'),
    },
    {
      path: '/bluewhale',
      name: 'bluewhale',
      component: () => import(/* webpackChunkName: "bluewhale" */ './views/BlueWhale.vue'),
    },
    {
      path: '/humpbackwhale',
      name: 'humpbackwhale',
      component: () => import(/* webpackChunkName: "humpbackwhale" */ './views/HumpbackWhale.vue'),
    },
    {
      path: '/killerwhale',
      name: 'killerwhale',
      component: () => import(/* webpackChunkName: "killerwhale" */ './views/KillerWhale.vue'),
    },
    {
      path: '/rightwhale',

```



Vue-cli 3: Code splitting

쪼개놓은 파일을 굳이 prefetch해주는 친절함 (..)

괜찮아요, config 파일 수정하면 되니까 (..)

```
// vue.config.js
module.exports = {
  chainWebpack: config => {
    // remove the prefetch plugin
    config.plugins.delete('prefetch')

    // or:
    // modify its options:
    // config.plugin('prefetch').tap(options => {
    //   options[0].fileBlacklist = options[0].fileBlacklist || []
    //   options[0].fileBlacklist.push(/myasyncRoute(.)+?.js$/)
    //   return options
    // })
  }
}
```

Vue-cli 3: Code splitting

[Home](#) | [About](#) | [Beluga](#) | [Blue Whale](#) | [Humpback Whale](#) | [Killer Whale](#) | [Right Whale](#) | [Sperm Whale](#)



Chasing Whales

For a guide and recipes on how to configure / customize this project, check out the [vue-cli documentation](#).

Installed CLI Plugins

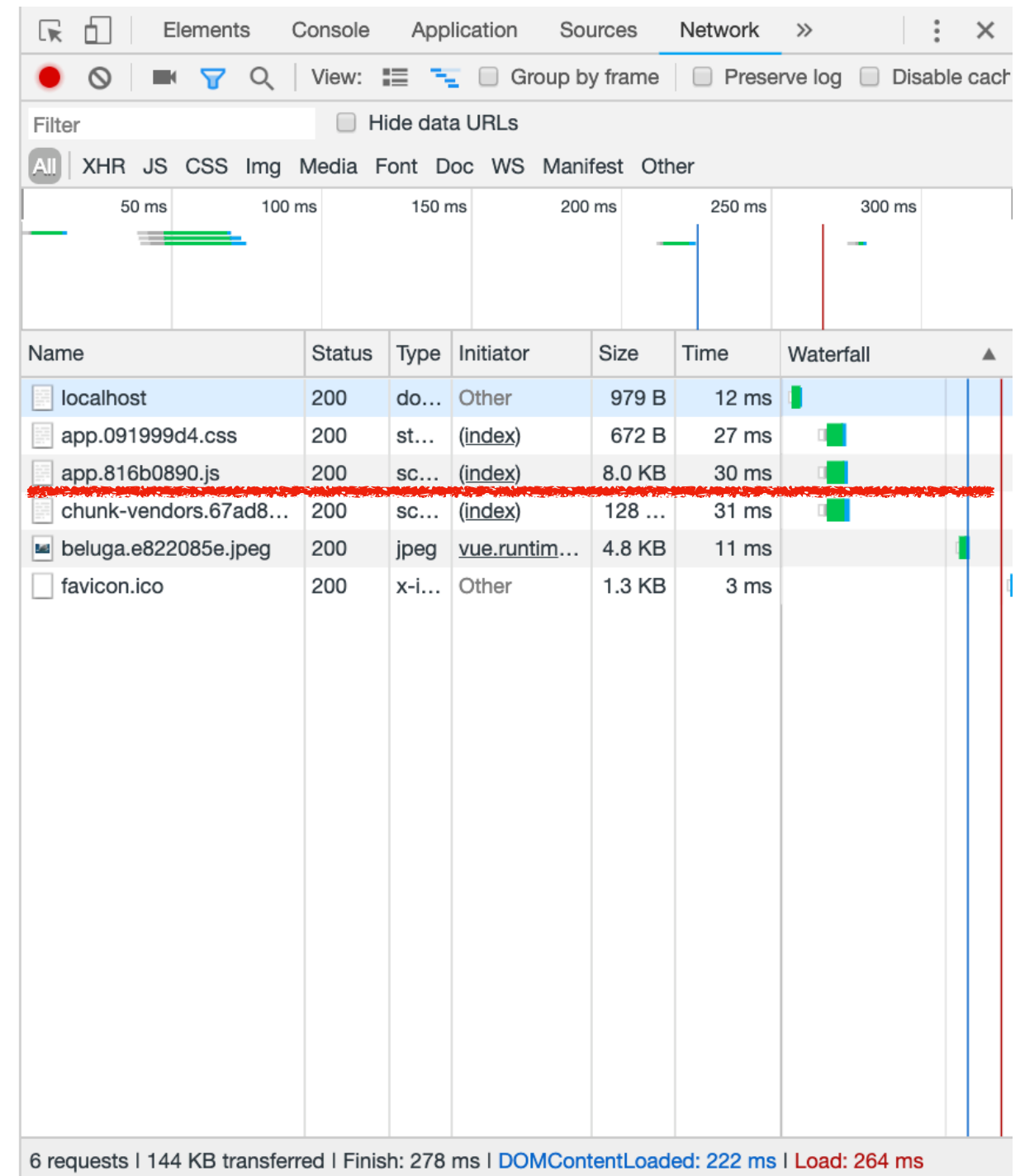
[babel](#) [typescript](#)

Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

Ecosystem

[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)



Anti-pattern

흔히 봤던 vendor.js (..)
 라이브러리 하나만 필요해도 다 받을 테니까요!

: 그러니 번들링할 포인트를 더 상세하게 설계하고,
 공통으로 사용하는 dependency끼리
 번들링 해주는 splitChunksPlugin를 쓰자

감사합니다

