Modern JavaScript Apps with Neutrino.js

leandro.frigerio@devugees.org

14.02.2018

Neutrino.js

Neutrino.js is a companion tool which lets you build web (and Node.js, React) applications with shared presets or configurations.

It intends to make the process of initializing and building projects much simpler by providing minimal development dependencies.

Why Neutrino?

„We created Neutrino to solve problems we faced creating front-end applications across teams within Mozilla's Release & Productivity organization.

Neutrino is currently in use by several Mozilla projects including TaskCluster, Treeherder, and Unified Logviewer. We maintain and support Neutrino because it is something we ourselves need and use, and we hope that everyone who uses it will also benefit.“

Eli Perelman

Neutrino.js author

Eli Perelman is a JavaScript and Node.js Obsessionalist™, currently working on web tooling for Mozilla's Release & Productivity team. In his spare time he enjoys working on open source and experimenting with electronic music production. Child of the '90s web. Skeptic.

http://eliperelman.com

Neutrino.js

Neutrino uses **Webpack** to build web, Node.js and React projects by providing complete build presets which can be shared across targets and projects.

You can use Neutrino base presets to get started building a variety of projects, create your own presets by extending the Neutrino core ones to be shared across your own projects or even by the community. Presets can even be manipulated on a project-by-project basis to handle almost any build situation your preset doesn't cover.

Why you should use Neutrino?

Starting a new JS project often brings along a significant effort into preparing your environment prior to starting development on the project.

Many developers have a preference for using cutting-edge features and modern tooling workflows to make the process enjoyable.

Unfortunately, this approach can often have quite a bit of learning curve as people spend time researching best practices, following complex configuration tutorials, and experimenting with boilerplates.

Why we should use Neutrino?

Neutrino **combines** the build and configuration power of Webpack and bolts on the capability to build JavaScript-based projects with presets.

A **preset** is the fundamental building block of Neutrino. With it you can supplement a project with a number of features including how it is compiled, ensuring code quality, and even quickly add testing. By breaking up configuration into composable units we can foster an ecosystem where customizing a project becomes plug and play.

Another boilerplate?

„The proliferation of boilerplate and metapackages is one thing we are trying to reduce. These types of projects are great, and do serve a purpose. But what if you wanted to make a configuration change across all your projects? You must make config changes in many places, including the original boilerplate, whereas presets give you the power to confine these changes to a single package. Some of these projects also make a tradeoff between ease of set up and black-boxing the configuration. Once you decide to make a configuration change, you are forced to maintain the entire configuration and its dependencies in perpetuity.

We believe Neutrino represents a good balance between ease of set up and future extensibility.“

https://neutrino.js.org/presets/#why-not-a-boilerplate-or-alternative

Scarfolding

Scarfolding… you know…

Easy setup

- Zero upfront configuration necessary to start developing and building a Web or Node or React web app.
- Modern Babel compilation adding JSX, ES modules, last 2 major browser versions, async functions, and object rest spread syntax.
- Support for hot module replacement.
- Webpack loaders for importing HTML, CSS, images, icons, and fonts directly from JavaScript.

Easy setup

- Webpack Dev Server during development.
- Automatic creation of static HTML pages, no templating necessary.
- Production-optimized bundles with Babili minification and easy chunking.
- Easily extensible to customize your project as needed, no blackboxes or ejecting required.

Code Quality

It’s just as easy to add linting.

With Neutrino and the Airbnb style keeping your code quality high is as simple as adding presets and following conventions.

You can follow the same guidelines to add testing to the project. Just choose a testing preset and you are on your way.

Resources

Documentation
https://neutrino.js.org

Development
https://github.com/mozilla-neutrino/neutrino-dev

Requirements

- Node.js v6 LTS, v8, v9
- Yarn v1.2.1+, or npm v5.4+
- Neutrino v8