@usepa-ngst/dev-require

This module was created to allow local version of modules instead of npm published versions to make development easier. It turns out this isn’t going to be used because embedding it everywhere require is used is a bit much. Also doesn’t work with static import so moved to cli tool @ngst/dev-link. Documentation is below just for reference.

@usepa-ngst/dev-require module exposes a factory function to create an object with a function called require() which accepts the package name and possible options. There is a config file that maps package names to physical location on local computer. This config file is local and should be added to .gitignore so that it is not checked in because we don’t want local version of files to be used on servers. This probably wouldn’t be a problem though because local version will only be used if the config file location is setting using this environment variable:

devRequireConfigPath

Don’t set the env var on servers or anywhere you don’t want to use local files set up in config file. The local config file maps package name to local path like:

```
let config = {
    packages: {
        '@aevans04/test-node-monorepo-a':{path:`${monoRepoPath}/package-a`,disabled:false},
        '@aevans04/test-node-monorepo-b':{path:`${monoRepoPath}/package-b`,disabled:false},
        '@aevans04/test-node-monorepo-c':{path:`${monoRepoPath}/package-c`,disabled:false},
    },
    alternates: {
        alt1: {
            merge: false,
            packages: {
                '@aevans04/test-node-monorepo-a':{path:`${monoRepoPath}/package-a`,disabled:true},
                '@aevans04/test-node-monorepo-b':{path:`${monoRepoPath}/package-b`,disabled:false},
            }
        },
        alt2: {
            merge: true,
            packages: {
                '@aevans04/test-node-monorepo-a':{path:`${monoRepoPath}/package-a`,disabled:true},
            }
        }
    },
};
```

The mapping can also be disabled if you want to use the remote npm version of file. Typically the main default packages are used, but alternate configurations can be set up. If these alternate configs are used by caller of the dev-require require function, they will replace the default config. But if merge=true then the alternate config will be merged with the default config. In the example above, the default config will use package-a, package-b and package-c from local files. The alt1 config will only use package-b and the alt2 config will use package-b and package-c.

In order to make the use of dev-require simpler in your project, you can include a local module like this:

```
//can probably use npm link for this for just comment and uncomment because can't use dev-require
for dev-require!
//if hard coding like this might have to update the path of local dev-require below
//let devRequireFactory = require('../dev-require');
let devRequireFactory = require('@usepa-ngst/dev-require');

let devRequire = devRequireFactory();

function devRequireWrapper(packageName) {
    let packageInstance = devRequire.require(packageName,{alternateName:'alt1'}) ||
require(packageName);

    return packageInstance
}

module.exports = devRequireWrapper;
```

Much easier to just include this so that you can call the devRequireWrapper function wherever a module needs to be required. A lot less code and can reuse the devRequire instance.