# Angular CLI Commands

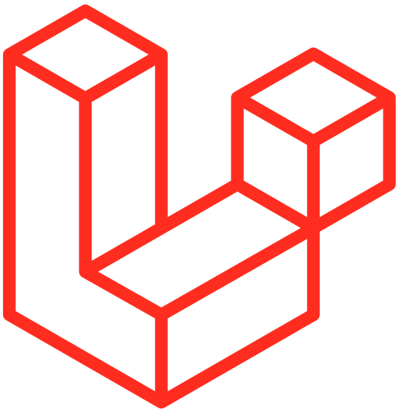
## Angular CLI Docs

| Angular CLI command | Description |
|---|---|
| **ng new** | *Creates a new Angular project with the specified name.* |
| **ng serve** | *Compiles and serves the application, rebuilding it on file changes.* |
| **ng build** | *Compiles the application into an output directory.* |
| **ng test** | *Runs unit tests in a project.* |
| **ng e2e** | *Runs end-to-end tests in a project.* |
| **ng generate** | *Generates a new component, directive, pipe, service, class, interface, enum, module, or guard.* |
| **ng add** | *Adds a new library or capability to a project* |
| **ng update** | *Updates the Angular framework and its dependencies to the latest version.* |
| **ng lint** | *Runs linting on the project's TypeScript code to check for syntax and style issues* |
| **ng help** | *Shows the Angular CLI help page* |

# Angular Modules

Angular Modules Docs

| Angular Modules | Description |
|---|---|
| **NgModule** | *An Angular decorator used to define a module that collects and organizes related components, directives, pipes, and services into functional units.* |
| **Declarations** | *An array of components, directives, and pipes that belong to a module and are available for use within its components* |
| **Imports** | *An array of other modules that a module depends on and whose components, directives, and pipes are available for use within its components.* |
| **Exports** | *An array of components, directives, and pipes that a module makes available to other modules that import it* |
| **Providers** | *An array of services, factories, or values that a module contributes to the global dependency injection system or makes available for injection within its components.* |
| **Bootstrap** | *An array of components that are automatically instantiated when an Angular application is launched* |
| **Entry components** | *A type of component that is not referenced in a template but is dynamically created at runtime, typically as a modal or dialog box* |
| **Lazy loading** | *A technique for loading modules on demand, as needed, to improve application startup time and reduce initial load times* |



# Laravel Artisan Commands

## Laravel docs

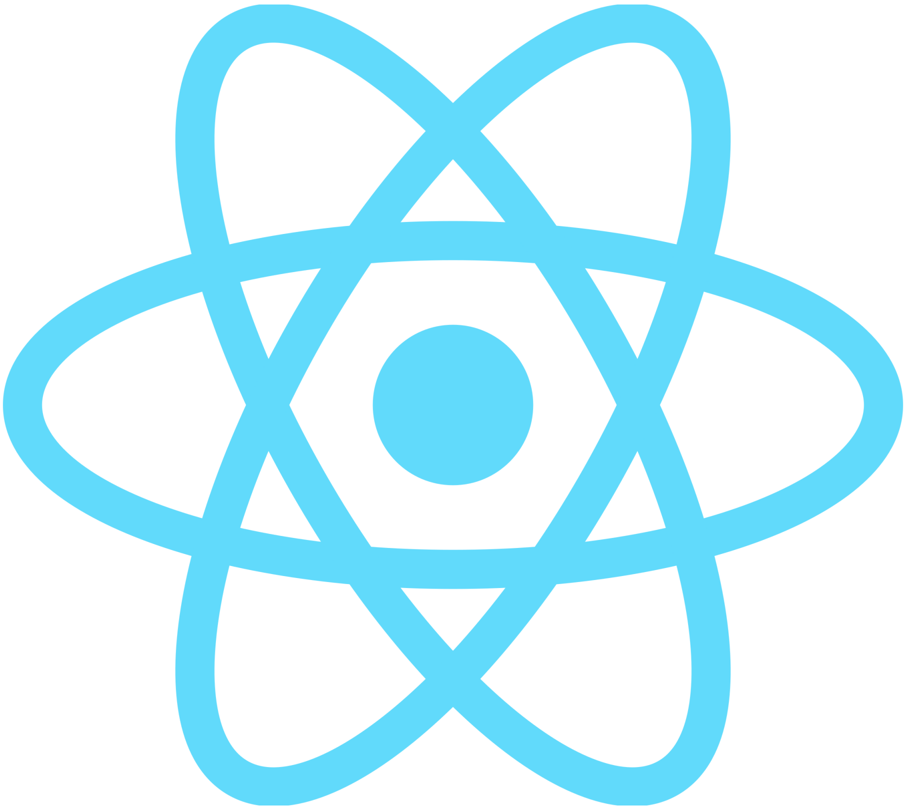
| Artisan commands | Description |
|---|---|
| **php artisan help** | *brings up a "help" screen which displays and describes the command's available arguments and options.* |
| **php artisan serve** | *Starts a development server for the Laravel application.* |
| **php artisan make:model** | *Generates a new model class* |
| **php artisan make:controller** | *Generates a new controller class* |
| **php artisan make:middleware** | *Generates a new middleware class* |
| **php artisan make:migration** | *Generates a new database migration* |
| **php artisan migrate** | *Runs all outstanding database migrations* |
| **php artisan migrate:rollback** | *Rolls back the last batch of executed database migrations* |
| **php artisan tinker** | *Launches an interactive REPL shell for the Laravel application* |
| **php artisan route:list** | *Displays a list of all registered routes for the application* |
| **php artisan make:auth** | *Generates authentication scaffolding, including views and controllers, for the Laravel application.* |

COMPOSER

# Composer Commands

## Composer docs

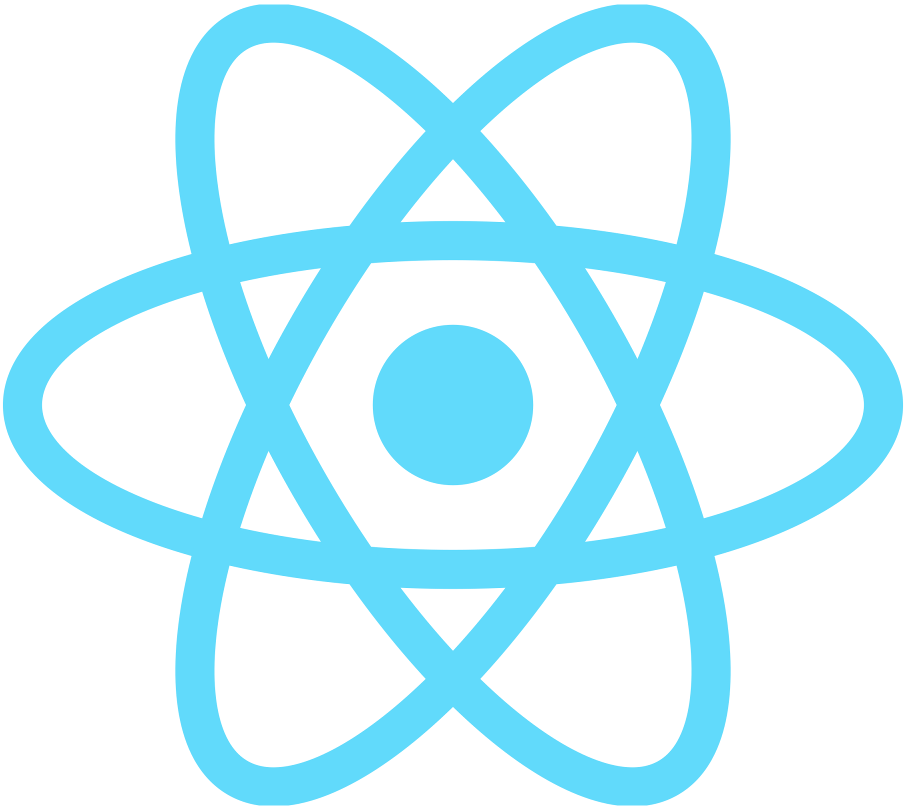
| Composer Commands | Description |
|---|---|
| **composer install** | *Installs all the packages listed in the* `composer.json` *file of a project* |
| **composer update** | *Updates all the packages listed in the* `composer.json` *file to their latest version.* |
| **composer require** | *Installs a new package in the project and adds it to the* `composer.json` *file.* |
| **composer remove** | *Removes a package from the project and updates the* `composer.json` *file.* |
| **composer show** | *Lists all installed packages, their versions, and their dependencies.* |
| **composer self-update** | *Updates the Composer installation itself to the latest version.* |



# React Commands

## React docs

| React Commands | Description |
|---|---|
| **npx create-react-app** | *Creates a new React application with the specified name.* |
| **npm start** | *Starts a development server for the React application* |
| **npm run build** | *Builds the production-ready version of the React application* |
| **npm test** | *Runs all tests in the React application* |
| **npm install** | *Installs a new package in the React application* |
| **npm uninstall** | *Removes a package from the React application* |
| **npm run eject** | *Removes the abstraction layer provided by Create React App and exposes the underlying configuration files, allowing for more advanced customizations.* |
| **npm run lint** | *Runs linting on the project's JavaScript code to check for syntax and style issues.* |
| **npm run format** | *Automatically formats the project's JavaScript code to adhere to a consistent coding style.* |



# React modules

## React docs

| React modules | Description |
|---|---|
| **React** | *The core library for building user interfaces in React* |
| **ReactDOM** | *A package that provides DOM-specific methods that can be used at the top level of a web app* |
| **React Router** | *A library for handling routing in React applications.* |
| **Axios** | *A library for making HTTP requests from a React application* |
| **PropTypes** | *A package that provides runtime type checking for React props* |
| **Redux** | *A predictable state container for JavaScript apps* |
| **React Redux** | *A package that provides the bindings between React and Redux* |
| **Moment.js** | *A library for parsing, validating, and manipulating dates and times in JavaScript.* |
| **React Native** | *A framework for building mobile applications using React* |
| **Enzyme** | *A testing utility for React that provides tools for working with React components* |

# Git Commands

## Git docs

| Git Commands | Description |
|---|---|
| **git init** | *Initializes a new Git repository in the current directory.* |
| **git clone** | *Clones an existing Git repository from the specified URL to a new directory on the local machine.* |
| **git add <file(s)>** | *Adds one or more files to the staging area in preparation for a commit* |
| **git commit -m ""** | *Commits changes to the local repository with a descriptive message.* |
| **git push** | *Pushes local commits to the specified branch on the remote repository.* |
| **git pull** | *Fetches and merges changes from the specified remote repository and branch.* |
| **git status** | *Displays the current status of the working directory and staging area.* |
| **git log** | *Displays a log of all previous commits in the repository* |
| **git branch** | *Lists all local branches in the repository* |
| **git checkout** | *Switches to the specified branch* |

npm

# NPM Commands

## NPM docs

| NPM Commands | Description |
|---|---|
| **npm init** | *Initializes a new npm package in the current directory, creating a `package.json` file.* |
| **npm install** | *Installs a package as a dependency for the current project.* |
| **npm install --save-dev** | *Installs a package as a development dependency for the current project.* |
| **npm uninstall** | *Removes a package from the current project's dependencies* |
| **npm update** | *Updates a package to the latest version in the current project* |
| **npm outdated** | *Lists any outdated packages that are currently installed in the current project.* |
| **npm publish** | *Publishes the current package to the npm registry for others to use.* |
| **npm search** | *Searches the npm registry for packages matching the specified query.* |
| **npm run** | *Runs the specified script defined in the `scripts` section of the `package.json` file.* |
| **npm audit** | *Checks for any known security vulnerabilities in the packages installed in the current project.* |

# Vite Commands

## Vite Docs

| Vite Commands | Description |
|---|---|
| **vite init** | *Initializes a new Vite project, creating a `package.json` file and installing dependencies.* |
| **vite dev** | *Starts a development server with hot reloading enabled* |
| **vite build** | *Builds the production-ready version of the project in the `dist` directory* |
| **vite serve** | *Serves the production-ready version of the project using a static file server* |
| **vite create** | *Creates a new Vite project using a predefined template.* |
| **vite optimize** | *Runs the Vite optimizer to optimize the production build.* |
| **vite inspect** | *Prints debugging information about the current Vite configuration and plugins.* |
| **vite plugin** | *Manages Vite plugins, with subcommands for adding, removing, and listing plugins.* |
| **vite ssr** | *Builds and serves the server-rendered version of the project* |
| **vite test** | *Runs tests for the project using a testing framework like Jest or Mocha.* |

node
JS®

# NodeJS Commands

## NodeJS docs

| NodeJS Commands | Description |
|---|---|
| **node** | *Starts a Node.js REPL (Read-Eval-Print Loop) shell* |
| **node** | *Runs the specified JavaScript file with Node.js.* |
| **node --version** | *Displays the version of Node.js currently installed on the system* |
| **node --help** | *Displays a list of available Node.js command line options* |
| **node package.json** | *Runs scripts defined in the `scripts` section of the `package.json` file.* |
| **node -e ""** | *Evaluates the specified script using Node.js* |
| **node --inspect** | *Enables debugging for the specified file* |
| **node --experimental-modules** | *Enables ECMAScript modules for the specified file.* |
| **node --max-old-space-size=** | *Sets the maximum memory size for Node.js when running the specified file.* |
| **node --trace-warnings** | *Enables tracing for warnings when running the specified file.* |

jQuery

write less, do more.

# JQuery Commands

## JQuery Docs

| JQuery Commands | Description |
|---|---|
| **$(selector)** | *Selects an HTML element using a CSS-style selector* |
| **$(selector).click(handler)** | *Binds a click event handler to the selected element(s)* |
| **$(selector).hide()** | *Hides the selected element(s)* |
| **$(selector).show()** | *Displays the selected element(s)* |
| **$(selector).addClass(class)** | *Adds a CSS class to the selected element(s)* |
| **$(selector).removeClass(class)** | *Removes a CSS class from the selected element(s)* |
| **$(selector).toggleClass(class)** | *Toggles a CSS class on the selected element(s).* |
| **$.get(url, data, callback)** | *Sends an HTTP GET request to the specified URL and invokes the callback function with the response data.* |
| **$.post(url, data, callback)** | *Sends an HTTP POST request to the specified URL and invokes the callback function with the response data.* |
| **$(selector).attr(attribute, value)** | *Sets the specified attribute to the specified value on the selected element(s).* |