Update laravel from 7 to 8

To keep your Laravel project up to date and benefit from the latest features and improvements, it's essential to update your dependencies regularly. This guide will walk you through updating specific packages in your composer.json file and executing the necessary composer update command.

Step 1: Modify composer.json
Open your composer.json file, typically located in the root directory of your Laravel project. Locate the "require" section and update the versions of the following dependencies:

```
"require": {

  "guzzlehttp/guzzle": "^7.0.1",

  "facade/ignition": "^2.3.6",

  "laravel/framework": "^8.0",

  "laravel/ui": "^3.0",

  "nunomaduro/collision": "^5.0",

  "phpunit/phpunit": "^9.0"

},
```

Note: The caret symbol (^) before each version number allows Composer to install future minor and patch updates while keeping the major version fixed.

Step 2: Execute Composer Update
Open your terminal or command prompt, navigate to your Laravel project's root directory, and run the following command to update the dependencies:

```
composer update
```

Composer will download the latest versions of the specified packages and update the composer.lock file with the new versions and their dependencies.

Step 3: Verify and Test
After updating the dependencies, it's crucial to verify that your application is still running correctly. Run your Laravel project and ensure that all functionality, including tests, works as expected. Pay particular attention to any backward-incompatible changes that might affect your application.

Final Words
Keeping your Laravel project and its dependencies up to date is essential for security, performance, and compatibility with the latest features. By following the steps above, you can ensure that your project benefits from the most recent improvements while maintaining a stable and reliable codebase. Remember to review the documentation of updated packages to take advantage of new features and consider any potential changes required in your codebase.