Composer PHP package manager

CONCEPTS

Commit composer.lock for applications: install then gives every machine the same versions.

composer.json your constraints
↓ `update / require`
composer.lock exact versions
↓ `install`
vendor/ packages + autoload.php

install reads the lock file; update resolves composer.json and rewrites the lock.

PROJECT

`composer init`
Create composer.json interactively

`composer create-project <vendor/pkg> <dir>`
Start a new project from a package

INSTALL & REMOVE

`composer install`
Install exactly what the lock says

`composer install --no-dev`
Skip require-dev packages

`composer install -o`
Optimize the autoloader

`composer remove <vendor/pkg>`
Remove a dependency

REQUIRE & UPDATE

`composer require <vendor/pkg>`
Add a package and install it

`composer require <vendor/pkg>:<constraint>`
Add with a version constraint

`composer require --dev <vendor/pkg>`
Add a dev-only dependency

REQUIRE & UPDATE (cont.)

`composer update`
Update everything within constraints

`composer update <vendor/pkg>`
Update only one package

`composer update <vendor/pkg> -W`
Also update its dependencies

INSPECT

`composer show`
List installed packages

`composer show <vendor/pkg>`
Details of one package

`composer outdated`
List packages with newer versions

`composer outdated -D`
Only direct dependencies

`composer why <vendor/pkg>`
Why is this installed?

AUTOLOAD & SCRIPTS

`composer dump-autoload`
Regenerate the autoloader

`composer run-script <name>`
Run a script from composer.json

`composer <name>`
Shortcut for run-script

SECURITY & AUDIT

`composer audit`
Check for known vulnerabilities

GLOBAL & OPTIONS

`composer --version`
Show the Composer version

`composer help <command>`
Help for a command

`composer list`
List all commands