## NAME

**npm-view** − View registry info

### Synopsis

npm view [<@scope>/]<name>[@<version>] [<field>[.<subfield>]...]

aliases: info, show, v

### Description

This command shows data about a package and prints it to stdout.

As an example, to view information about the **connect** package from the registry, you would run:

npm view connect

The default version is **"latest"** if unspecified.

Field names can be specified after the package descriptor. For example, to show the dependencies of the **ronn** package at version **0.3.5**, you could do the following:

npm view ronn@0.3.5 dependencies

You can view child fields by separating them with a period. To view the git repository URL for the latest version of **npm**, you would run the following command:

npm view npm repository.url

This makes it easy to view information about a dependency with a bit of shell scripting. For example, to view all the data about the version of **opts** that **ronn** depends on, you could write the following:

npm view opts@$(npm view ronn dependencies.opts)

For fields that are arrays, requesting a non−numeric field will return all of the values from the objects in the list. For example, to get all the contributor email addresses for the **express** package, you would run:

npm view express contributors.email

You may also use numeric indices in square braces to specifically select an item in an array field. To just get the email address of the first contributor in the list, you can run:

npm view express contributors[0].email

Multiple fields may be specified, and will be printed one after another. For example, to get all the contributor names and email addresses, you can do this:

npm view express contributors.name contributors.email

"Person" fields are shown as a string if they would be shown as an object. So, for example, this will show the list of **npm** contributors in the shortened string format. (See npm help **package.json** for more on this.)

npm view npm contributors

If a version range is provided, then data will be printed for every matching version of the package. This will show which version of **jsdom** was required by each matching version of **yui3**:

npm view yui3@'>0.5.4' dependencies.jsdom

To show the **connect** package version history, you can do this:

npm view connect versions

### Configuration

<!−− AUTOGENERATED CONFIG DESCRIPTIONS START −−> <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

#### json

- Default: false
- Type: Boolean

Whether or not to output JSON data, rather than the normal output.

- In **npm pkg set** it enables parsing set values with JSON.parse() before saving them to your **package.json** .

Not supported by all npm commands. <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

**workspace**

- Default:
- Type: String (can be set multiple times)

Enable running a command in the context of the configured workspaces of the current project while filtering by running only the workspaces defined by this configuration option.

Valid values for the **workspace** config are either:

- Workspace names
- Path to a workspace directory
- Path to a parent workspace directory (will result in selecting all workspaces within that folder)

When set for the **npm init** command, this may be set to the folder of a workspace which does not yet exist, to create the folder and set it up as a brand new workspace within the project.

This value is not exported to the environment for child processes. <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

**workspaces**

- Default: null
- Type: null or Boolean

Set to true to run the command in the context of **all** configured workspaces.

Explicitly setting this to false will cause commands like **install** to ignore workspaces altogether. When not set explicitly:

- Commands that operate on the **node_modules** tree (install, update, etc.) will link workspaces into the **node_modules** folder. − Commands that do other things (test, exec, publish, etc.) will operate on the root project, *unless* one or more workspaces are specified in the **workspace** config.

This value is not exported to the environment for child processes. <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

**include−workspace−root**

- Default: false
- Type: Boolean

Include the workspace root when workspaces are enabled for a command.

When false, specifying individual workspaces via the **workspace** config, or all workspaces via the **workspaces** flag, will cause npm to operate only on the specified workspaces, and not on the root project. <!−− automatically generated, do not edit manually −−> <!−− see lib/utils/config/definitions.js −−>

<!−− AUTOGENERATED CONFIG DESCRIPTIONS END −−>

**Output**

If only a single string field for a single version is output, then it will not be colorized or quoted, to enable piping the output to another command. If the field is an object, it will be output as a JavaScript object literal.

If the −−**json** flag is given, the outputted fields will be JSON.

If the version range matches multiple versions then each printed value will be prefixed with the version it applies to.

If multiple fields are requested, then each of them is prefixed with the field name.

**See Also**

- npm help search
- npm help registry
- npm help config
- npm help npmrc
- npm help docs