

# Hex

## Package Manager

@emjii



**2 758**  
packages  
available



**12 848**  
package  
versions



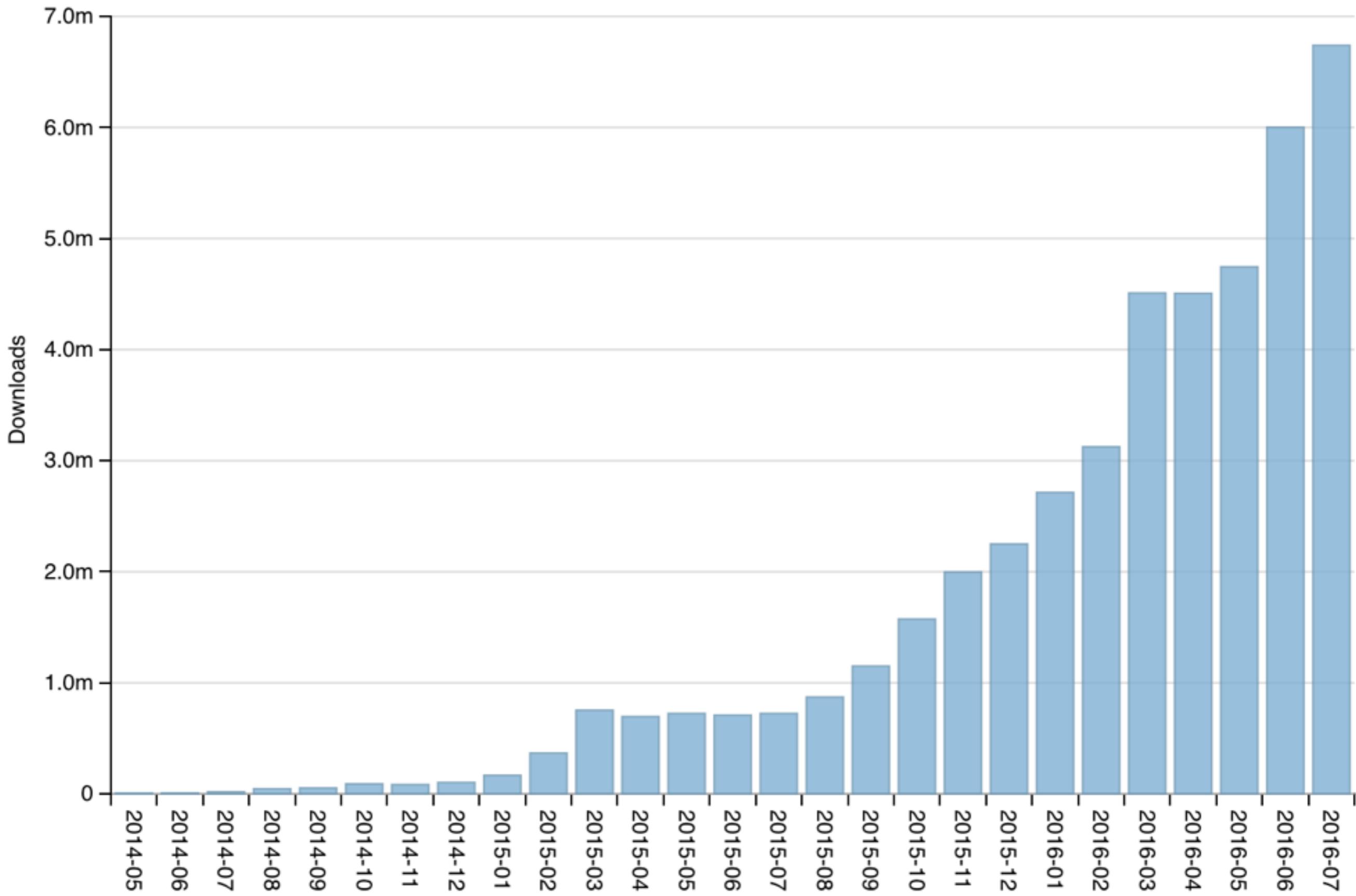
**169 715**  
downloads  
yesterday



**863 740**  
downloads  
last 7 days



**50 115 066**  
downloads  
all time



# Mix

# Mix

- Generate new projects
- Compile
- Run tests
- Handle dependencies
- Whatever else you can think of

# mix.exs

```
defmodule MyProject.Mixfile do
  use Mix.Project
  def project do
    [app: :my_project,
     version: "0.1.0",
     elixir: "~> 1.0"]
  end
end
```

# Dependencies

```
defmodule MyProject.Mixfile do
  use Mix.Project
  def project do
    [app: :my_project,
     version: "0.1.0",
     deps: deps()]
  end
  defp deps do
    [{:poolboy, github: "devinus/poolboy"},
     {:ecto, "~> 2.0"}]
  end
end
```

# Dependencies

- `$ mix deps.get / deps.update / ...`
- Converge and sort
- Repeatable builds
- Rebar & Makefile dependencies

# mix.lock

```
%{"cowboy": {:hex, :cowboy, "1.0.4",
  "a324a8df9f2316c833a470d918aaaf73ae894278b8aa6226ce7a9bf699388f878", ...},
  "cowlib": {:hex, :cowlib, "1.0.2",
  "9d769a1d062c9c3ac753096f868ca121e2730b9a377de23dec0f7e08b1df84ee", ...},
  "ranch": {:hex, :ranch, "1.2.1",
  "a6fb992c10f2187b46ffd17ce398ddf8a54f691b81768f9ef5f461ea7e28c762", ...},
  "poolboy": {:git, "git://github.com/devinus/poolboy.git",
  "2b8d9805306d36f22330f432ae6472f1f2625c30", ...}}}
```

# Dependencies

- `$ mix deps.get / deps.update / ...`
- Converge and sort
- Repeatable builds
- Rebar & Makefile dependencies

# Architecture

# Specifications

- Package structure
- Endpoints
- Registry
- HTTP API
- Fully documented

# Package structure

**ecto-2.0.0.tar**

**VERSION**

**metadata.config**

**CHECKSUM**

**contents.tar.gz**

# Package metadata

```
{<<"app">>,<<"ecto">>}.
{<<"build_tools">>,[<<"mix">>]}.
{<<"description">>,
 <<"Ecto is a domain specific language for writing
queries and interacting with databases in Elixir.">>}.
{<<"elixir">>,<<"~> 1.2">>}.
{<<"files">>,...}.
{<<"licenses">>,[<<"Apache 2.0">>]}.
{<<"links">>,[{<<"GitHub">>,<<"https://github.com/
elixir-ecto/ecto">>}]}.
{<<"maintainers">>,...}.
{<<"name">>,<<"ecto">>}.
{<<"requirements">>,...}.
{<<"version">>,<<"2.0.0">>}.
```

# Repository

- `/registry.ets.gz`
- `/registry.ets.gz.signed`
- `/tarballs/PACKAGE-VERSION.tar`

# Registry

- Single ETS file

```
{"ecto",      [[["0.1.0", "0.1.1", "0.1.2", ...]]]}  
{"postgrex",  [[["0.1.0", "0.2.0", "0.2.1", ...]]]}  
  
{{"ecto", "0.1.0"}, [[[{"postgrex", "~> 0.1.0"},  
                      ["poolboy", "~> 1.1.0"],  
                      ...]]}}
```

# Dependency resolution

1. Add deps from mix.exs to pending requests
2. Take next pending, find latest matching release
  - 2a. Compare against activated packages
  - 2b. If no matching, backtrack
3. Activate the package
4. Add children to pending
5. Save state for backtracking
6. Goto 2

# Systems

- Website & HTTP API
  - Heroku
- Repository
  - Amazon S3 + Fastly
- Hexdocs
  - Amazon S3 + Fastly

# Community

# The package manager for the Erlang ecosystem



## Using with Elixir

Simply specify your Mix dependencies as two-item tuples like `{:ecto, "~> 0.1.0"}` and Elixir will ask if you want to install Hex if you haven't already. After installed, you can run `$ mix local` to see all available Hex tasks and `$ mix help TASK` for more information about a specific task.



## Using with Erlang

Download `rebar3`, put it in your `PATH` and give it executable permissions. Now you can specify Hex dependencies in your `rebar.config` like `{deps, [hackney]}`.



2 749  
packages  
available



12 816  
package  
versions



174 539  
downloads  
yesterday



848 098  
downloads  
last 7 days



49 945 351  
downloads  
all time

[MOST DOWNLOADED](#)[NEW PACKAGES](#)[RECENTLY UPDATED](#)

# Packages

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2749 Results Found  
(Sorted by downloads)

Sort By ▾

**poison** 2.2.0 1 722 072  
downloads

An incredibly fast, pure Elixir JSON library

**cowboy** 1.0.4 1 649 193  
downloads

Small, fast, modular HTTP server written in Erlang.

**cowlib** 1.3.0 1 630 774  
downloads

Support library for manipulating Web protocols.

**ranch** 1.2.1 1 425 031  
downloads

Socket acceptor pool for TCP protocols.

**plug** 1.2.0 1 413 081  
downloads

A specification and conveniences for composable modules between web applications

**decimal** 1.1.2 1 382 547  
downloads

# cowboy 1.0.4

Small, fast, modular HTTP server written in Erlang.

## Maintainers

Loïc Hoguin

## Links

[GitHub](#)

## License

ISC

 1 048 007  
downloads  
this version

 4 010  
downloads  
yesterday

 21 092  
downloads  
last 7 days

 1 649 193  
downloads  
all time

## Versions

[1.0.4](#) November 17, 2015

[1.0.3](#) September 13, 2015

[1.0.2](#) July 2, 2015

[1.0.1](#) July 2, 2015

[1.0.0](#) August 1, 2014

## Config

`mix.exs`

`{:cowboy, "~> 1.0"}`



`rebar.config`

`{cowboy, "1.0.4"}`



`erlang.mk`

`dep_cowboy = hex 1.0.4`



## Build Tools

`rebar` `make`

## Dependencies

`cowlib`  $\sim >$  1.0.0

`ranch`  $\sim >$  1.0

# Build tools

- Mix, Rebar3 & erlang.mk
- Compatibility between languages and tools

# Hexdocs

- Single place to find docs for all packages
- Automatically published
- Versioned
- Static file hosting
- Can get much improvement

# Ecto

&lt;/&gt;

Ecto is split into 4 main components:

- [Ecto.Repo](#) - repositories are wrappers around the data store. Via the repository, we can create, update, destroy and query existing entries. A repository needs an adapter and credentials to communicate to the database
- [Ecto.Schema](#) - schemas are used to map any data source into an Elixir struct. We will often use them to map tables into Elixir data but that's one of their use cases and not a requirement for using Ecto
- [Ecto.Changeset](#) - changesets provide a way for developers to filter and cast external parameters, as well as a mechanism to track and validate changes before they are applied to your data
- [Ecto.Query](#) - written in Elixir syntax, queries are used to retrieve information from a given repository. Queries in Ecto are secure, avoiding common problems like SQL Injection, while still being composable, allowing developers to build queries piece by piece instead of all at once

In the following sections, we will provide an overview of those components and how they interact with each other. Feel free to access their respective module documentation for more specific examples, options and configuration.

If you want to quickly check a sample application using Ecto, please check the [getting started guide](#) and the accompanying sample application.

## Repositories

[Ecto.Repo](#) is a wrapper around the database. We can define a repository as follows:

[sbroker](#)  
[sbroker\\_codel\\_queue](#)  
[sbroker\\_drop\\_queue](#)  
[sbroker\\_legacy](#)  
[sbroker\\_time](#)  
[sbroker\\_timeout\\_queue](#)  
[sscheduler](#)

# Sojourn-time based active queue management process broker

**See also:** [sbroker](#), [squeue](#), [squeue\\_codel](#), [squeue\\_codel\\_timeout](#), [squeue\\_naive](#), [squeue\\_timeout](#).

`sbroker` is a process match maker for two groups of processes. Active queue management is supplied by `squeue`, which is a queue module with a similar API to OTP's `queue`.

Four queue management callback modules are provided for use with `squeue`: `squeue_naive`, `squeue_timeout`, `squeue_codel` and `squeue_codel_timeout`.

*Generated by EDoc, Jul 4 2015, 11:42:34.*

# Hexdocs

- Single place to find docs for all packages
- Automatically published
- Versioned
- Static file hosting
- Can get much improvement

# escripts

```
$ mix escript.install git  
  git://github.com/phoenixframework/phoenix_new.git  
  
$ mix escript.install hex phoenix_new 1.0.0
```

# What do we get?

- Single place to find libraries, docs
- More collaboration, less forks
- Promote cross BEAM-lang usage

# Security

# What to secure

- Servers
- Hex installer
- HTTP API
- Registry integrity
- Package integrity
- User keys

# HTTPS in OTP

- Do your own hostname verification
- Handle partial certificate chains
- Bring your own certificate store
- httpc issues
- [github.com/deadtrickster/ssl\\_verify\\_fun.erl](https://github.com/deadtrickster/ssl_verify_fun.erl)
- [github.com/certifi/erlang-certifi](https://github.com/certifi/erlang-certifi)

# Integrity

- Registry signed with hex.pm online private key
- Registry includes checksums
- Lock files include checksums

# Authentication

- bcrypt hashed password
- API keys
- Keys encrypted locally
- Protection against package worm  
([www.kb.cert.org/vuls/id/319816](http://www.kb.cert.org/vuls/id/319816))

# Typo squatting

- [incolumitas.com/2016/06/08/  
typosquatting-package-managers/](https://incolumitas.com/2016/06/08/typosquatting-package-managers/)
- Nightly job checks package name  
similarity

# left-pad

- User removed highly depended on npm package
- Admins re-published package without users consent
- Lesson: Limit what users can do

# User-signed packages

- Public key infrastructure
- Web of trust
- Not a silver bullet

# Future

# Hex 1.0

- Specifications
- New registry format
- Stable API

# Current registry

- Single ETS file
- 930kb compressed
- Grows with ~20kb/week
- Forward compatibility is tricky

# New registry

- One file per package
- Index files /names and /versions
- Protocol buffers ensure forward compatibility
- Complicates dependency resolution
- Use local ETS file as cache

# Namespaces

- Allows private packages
- May not be enabled for public packages on hex.pm

# Package retirement

- For security issues, deprecations, renames, etc.
- Gives you a warning when fetching
- Package still available

# Current issues

- Package signing
- Libraries missing on hex.pm
- Private repositories
- ?

?

@emjii