

 ethpm / ethpm-spec PublicEthereum Package Manager <http://ethpm.github.io/ethpm-spec/>

☆ 159 stars    🍴 30 forks

☆ Star

👁 Watch ▼

&lt;&gt; Code

🔗 Issues 15

🔗 Pull requests 12

🎬 Actions

📁 Projects

📖 Wiki

🛡 Security

🔗 master ▼

...



njgheorghita Merge pull request #174 from fubuloubu/fix/extra-newlines ...

5 days ago

🕒 338

[View code](#)

# EthPM Package Specification

## Overview

[chat](#) [on gitter](#)

This repository comprises the formal specification and documentation source for the **EthPM package manifest** data format.

This data format is designed to be produced/consumed by Ethereum development tools. As such, this repository is intended for tool developers wishing to integrate with EthPM.

Package manifests are JSON-encoded, tightly-packed, with objects' keys sorted in lexicographic order. Package manifests may live on disk, but are intended to be produced programmatically and uploaded directly to a content-addressable storage network (e.g. [IPFS](#)). A package manifest describes a single package, including package name, version, dependencies, and references to distributed source files.

## Repository Contents

- [Schema Description](#)
  - [package.spec.json](#)
- [Use Case Examples](#)
  - [Examples Sources](#)
- [Glossary](#)
- [ethPM URI](#) JSON tests ( `./tests/fixtures/ethpm_uri_tests.json` )

## Examples / Use Cases

---

**Package:** owned (prettified)

```
{
  "name": "owned",
  "version": "1.0.0",
  "manifest": "ethpm/3",
  "meta": {
    "license": "MIT",
    "authors": [
      "Piper Merriam <pipermerriam@gmail.com>"
    ],
    "description": "Reusable contracts which implement a privileged 'owner' mode",
    "keywords": [
      "authorization"
    ],
    "links": {
      "documentation": "ipfs://QmUYcVzTfSwJoigggMxoe2g5STWAgJdisQsqcXHws7b1FW"
    }
  },
  "sources": {
    "contracts/Owned.sol": {
      "type": "solidity",
      "urls": [
        "ipfs://Qme4otpsS88NV8yQi8TfTP89EsQC5bko3F5N1yhRoi6cwGV"
      ],
      "installPath": "./contracts/Owned.sol"
    }
  }
}
```

Please see [Use Cases](#) for documented examples of different kinds of packages with varying levels of complexity. Source for use case examples can be found in the [examples/](#) directory of this repository.

Each example directory contains..

- `1.0.0.json` : ethpm v2 manifest (deprecated)
- `1.0.0-pretty.json` : ethpm v2 manifest (pretty printed) (deprecated)
- `v3.json` : ethpm v3 manifest
- `v3-pretty.json` : ethpm v3 manifest (pretty printed)
- `contracts/` : Directory containing the source contracts for the example
- `metadata/` : Directory containing example ethpm compliant [compiler metadata output](#) for each example contract

## Specification

The EthPM package manifest format is formally specified as a [JSON-Schema](#).

Please see [Package Specification](#) for a natural-language description of this schema, or see [package.spec.json](#) for the machine-readable version.

☰ README.md

Requirements: Python 3, `pip`, `make` Fork and clone this repo to get started. Then, activate a [virtual environment](#) in the cloned repo's directory and run `pip install -r requirements.txt`

## Building Sphinx docs locally

1. `cd docs`
2. `make html`

Docs are written in [reStructuredText](#) and built using the [Sphinx](#) documentation generator.

## Running tests locally

1. `pytest tests/`

## Test fixture schema

Each test fixture contains a `package` field with a raw, json encoded string of the manifest.

Each test fixture contains a `testCase` field that indicates whether the associated `package` is `invalid` or `valid`.

Each invalid test fixture contains an `errorInfo` field.

- The `errorPointer` field, which is a [jsonpointer](#) pointing towards the cause of the invalid error, is included for `invalid` tests.
- The `reason` field, which is a human readable description of the error, is included for `invalid` tests.
- The `errorCode` field, which is a machine readable description of the error, is included for `invalid` tests according to the following table.

N0001 - Invalid "manifest" field. N0002 - Invalid "name" field. N0003 - Invalid "version" field. N0004 - Invalid "sources" field. N0005 - Invalid "contractTypes" field. N0006 - Invalid "deployments" field. N0007 - Invalid "compilers" field. N0008 - Invalid "buildDependencies" field. N0009 - Invalid "meta" field.

## Releases

 2 tags


## Packages

No packages published

## Contributors 12



## Environments 1

 github-pages Active

## Languages

● Python 60.5% ● Shell 39.5%