THE FUTURE OF THE POSTGRES EXTENSION ECOSYSTEM

DAVID E. WHEELER
PGXN, TEMBO
LEGACY

Long history of extensibility

Two approaches

shared_preload_libraries

Pure SQL (including PLs)

A few intrepid adopters

PostGIS

BioPostgres

PL/R

PL/Proxy

pgTAP
POSTGRES 9.1

Dimitri Fontaine adds extensions

Key features:

- Compile and install
- CREATE/UPDATE/DROP EXTENSION
- pg_dump and pg_restore
“PostgreSQL today is not merely a database, it’s an application development platform.”

ME, 2010
**PGXN**

**FEATURES**

**Distribution**
- Source Code
- Registration
- Namespacing

**Discovery**
- Search
- Documentation
- Tags

**Installation**
- CLI
- Compile & install
- PGXS or Configure
PGXN

OUT OF SCOPE

Binary packaging

Binary distribution
  Defer to apt/yum

Developer tooling

Build tooling
  Defer to core (PGXS)
PGXN

HISTORY

Started 2010
Funded December 2010
pgxn.org launched April 2011
Client by Daniele Varrazzo April 2011
Dev utils by Dickson S. Guedes June 2011
“In classic SDLC fashion, the PGXN POC shipped as an MVP and was neglected.”

ME, JUST NOW
TIME PASSED...
UPTAKE

PGXN: 396 Extensions/369 Distributions
joelonsql/PostgreSQL-EXTENSIONs.md: 1,186
CPAN: 217,860 Perl modules/45,137 distributions
PyPI: 543,205 Python projects
RubyGems: 180,839 Ruby Gems
crates.io: 146,401 Rust Crates
POSTGRES EXTENSIONS:

Difficult to find and discover
Under-documented and difficult to understand
Maturity difficult to gauge
Hard to configure and install
No comprehensive binary packaging
Centralized source distribution insufficient
Insufficient developer tooling
FILLING THE GAPS
The Database Package Manager

For Postgres Trusted Language Extensions

Download and install
Install extensions via a SQL command

```
select dbdev.install(olirice-index_advisor);
```
Extension Registry

Get Trunk

Cargo cargo install pg-trunk

Trunk is an open-source package installer and registry for PostgreSQL extensions.
> extensions, **made easy**

npm for PostgreSQL

- Brew
- Curl

`brew install pgxman/tap/pgxman`
EMPHASES

Ease of use
Platform neutrality
Stats
Curation

Download statistics:
- 20 all time downloads
- 0 downloads in the last 30 days
- 1 download in the last 90 days
- 9 downloads in the last 180 days

Shell commands:
- `curl -sfL https://install.pgx.sh | sh`  
  pgxman successfully installed
- `pgxman install pgvector`  
  The following Debian packages will be installed:
  - postgresql-14-pgxman-pgvector=0.5.1
  - Do you want to continue? [Y/n]y

pgvector has been successfully installed

<table>
<thead>
<tr>
<th>Architecture</th>
<th>x86-64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Debian/Ubuntu</td>
</tr>
</tbody>
</table>
## MVPS

<table>
<thead>
<tr>
<th></th>
<th>Trunk</th>
<th>PGXMan</th>
<th>dbdev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pull request integration</td>
<td>Form-based submission</td>
<td>TLEs</td>
</tr>
<tr>
<td>Postgres 14–16</td>
<td>Postgres 12–16</td>
<td></td>
<td>CLI publishing</td>
</tr>
<tr>
<td>Debian/Ubuntu-only</td>
<td>Debian/Ubuntu-only</td>
<td></td>
<td>Postgres 11-16</td>
</tr>
<tr>
<td>AMD64-only</td>
<td>AMD64 &amp; ARM64</td>
<td></td>
<td>No C/pgrx</td>
</tr>
<tr>
<td>210 Extensions</td>
<td>Apt-only</td>
<td></td>
<td>pg_tle &amp;&amp; psql-http</td>
</tr>
<tr>
<td></td>
<td>81 Extensions</td>
<td></td>
<td>Count Unknown</td>
</tr>
</tbody>
</table>
STATE OF THE ECOSYSTEM
STATE OF AFFAIRS

Despite early promise...

No single source

Many incomplete attempts

Poor discovery

Difficult installation

Insufficient docs

Low adoption

But...
“There sure has been a lot of excitement around extensions lately.”

ME, AGAIN
PGXN V2

The New World Order

Started this year

Made possible by Tembo

Consider emerging patterns

Meditate on deficiencies

Engage deeply with the community

Design and implement new architecture

Serve the ecosystem for next decade
Collaborate to examine the ongoing work on PostgreSQL extension distribution, examine its challenges, identify questions, propose solutions, and agree on directions for execution.
WHO IS THE ECOSYSTEM?
EXTENSION AUTHORS
EXTENSION AUTHORS

Needs

Tools to create extensions

Learning materials

Standard patterns

Testing tools

Metadata management

Distribution & discovery

Packaging
EXTENSION AUTHORS

Delivery

Documentation
- How-tos
- Examples
- Tutorials
- References
- Book?

CLI for tasks
- Metadata
- Testing
- Scaffolding
- Release
- Documentation
- CI/CD Automation

Canonical Registry
- Discovery
- Binary Packaging
- Category Curation
- Incentives
- Quality Metrics
- Documentation standards
- Test coverage & matrices
## Install Management

```bash
❯ pgxn installs ls
```

<table>
<thead>
<tr>
<th>install</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>homebrew/14.12</td>
<td>/opt/homebrew/Cellar/postgresql@14/14.12</td>
</tr>
<tr>
<td>pgenv/16.3</td>
<td>/Users/david/.pgenv/pgsql-16.3</td>
</tr>
<tr>
<td>pgenv/17-devel</td>
<td>/Users/david/.pgenv/pgsql-17-devel</td>
</tr>
<tr>
<td>postgres.app/16.3</td>
<td>/Applications/Postgres.app/Contents/Versions/16/</td>
</tr>
</tbody>
</table>

```bash
❯ pgxn manage pgenv/16.3
Now managing extensions for pgenv/16.3
```
Develop CLI

```bash
pgxn develop new --extension --language c --vcs github bike
```

Scaffolding C extension “bike”…

- Create `bike.control`
- `META.json`
- `bike.control`
- `Makefile`
- `README.md`
- `src/bike.c`
- `sql/bike--0.0.1.sql`
- `doc/bike.md`
- `test/sql/base.sql`
- `test/expect/base.out`
- `.github/workflows/ci.yml`
- `.github/workflows/release.yml`

- Init repository
  - `git init bike`
Develop CLI

```bash
pgxn develop add-upgrade 0.0.2
```

Adding upgrade from 0.0.1 to 0.0.2...

- 🚫 Rename sql/bike--0.0.1.sql to sql/bike--0.0.2.sql
- ⚙️ Create sql/bike--0.0.1--0.0.2.sql
- 🚺 Update META.json
  
  bike.control
Develop CLI

❯ pgxn develop require 14

Require PostgreSQL 14...
  ➔ Update META.json

❯ pgxn develop require semver 0.32.0

Requiring semver 0.32.0...
  ➔ Update META.json
  semver.control

❯
Develop CLI

pgxn develop test --username postgres

Testing bike 0.0.2 in pgenv/16.3

```shell
make
gcc -c -o src/pair.o src/pair.c
gcc src/pair.o -o src/pair.so
make install
install -c -m 644 bike.control '/pgsql/share/extension/'
install -c -m 644 sql/bike--0.0.2.sql '/pgsql/share/bike/'
install -c -m 644 sql/bike--0.0.1--0.0.2.sql '/pgsql/share/bike/'
install -c -m 644 src/bike.so '/pgsql/lib/
make installcheck PGUSER=postgres
ok 1         - base                  15 ms
1..1
# All 1 tests passed.
```
Develop CLI

```bash
pgxn develop release
```

Releasing bike 0.0.2 on PGXN...
- ✅ Validate META.json... Okay
- 📦 Bundle bike-0.0.2.zip with git archive...
- 🚀 Publish bike-0.0.2.zip on PGXN...

🎉 Done: https://pgxn.org/theory/bike/0.0.2
Bump version in README to 0.32.1 #146

Summary

Triggered via push 2 months ago

theory pushed → 804f954 main

Jobs

- PostgreSQL 16
- PostgreSQL 15
- PostgreSQL 14
- PostgreSQL 13
- PostgreSQL 12
- PostgreSQL 11
- PostgreSQL 10
- PostgreSQL 9.6
- PostgreSQL 9.5
- PostgreSQL 9.4
- PostgreSQL 9.3
- PostgreSQL 9.2

Matrix: build

- 12 jobs completed
  - Show all jobs

Annotations

- 12 warnings

Release on PGXN and GitHub

- Set up job
- Initialize containers
- Check out the repo
- Bundle the Release
- Release on PGXN
- Generate Release Changes
- Create GitHub Release
- Upload Release Asset
- Post Check out the repo
- Stop containers
- Complete job

succeeded last month in 35s
PACKAGERS
EXTENSION PACKAGERS

Needs

Release notification
Build automation
Predictable download URLs
Flexible sourcing
Extensive metadata
Feedback loops
EXTENSION PACKAGERS

Delivery

Release feeds & notification Webhooks

Package on every release

Build from source or repackage community binaries

Metadata to automate building from source

Report build and test failures

Submit download stats to root registry
Name: bike
Version: 0.0.2
Source0: https://pgxn.org/dist/theory/bike/bike-%{version}.zip
BuildRoot: %{_tmppath}/%{name}-%{version}-%{release}-root-%(%%id_u -n)

%prep
%setup -q -n bike-%{version}

%build
pgxn package build

%install
pgxn package install --destdir $RPM_BUILD_ROOT
%{_fixperms} $RPM_BUILD_ROOT/*

%check
pgxn package installcheck --username postgres

%clean
rm -rf $RPM_BUILD_ROOT
Needs

- Reliable sourcing
- Extension management
- Inventory & audit reporting
- Packaging choice & standardization
- Trusted package sources
- Internal packaging options
- Mirroring & curation
- Upgrade paths
Delivery

Extension Management
- Fast binary installation
- No compilation required
- Use community binary packages
- Or community OS packages
- Or build own packages from binaries or source
- Extension reporting
- Registry access controls
- Simplify upgrades (dupe extensions)?

Binary Registry Federation
- Run own binary registry
- Package your own extensions
- Use the same tooling
- Gate it by authentication or network
- Subscribe to commercial registries with additional features

Binary Registry Mirroring
- Mirror upstream registry
- Fast local performance
## Package Management

```bash
❯ pgxn package ls
Packages installed in pgdb/16.3:

<table>
<thead>
<tr>
<th>package</th>
<th>installed</th>
<th>latest</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>core/citext</td>
<td>1.6.0</td>
<td>✅</td>
<td>data type for case-insensitive character strings</td>
</tr>
<tr>
<td>core/jsonb_plperl</td>
<td>1.0.0</td>
<td>✅</td>
<td>transform between jsonb and plperl</td>
</tr>
<tr>
<td>core/unaccent</td>
<td>1.1.0</td>
<td>✅</td>
<td>text search dictionary that removes accents</td>
</tr>
<tr>
<td>theory/semver</td>
<td>0.32.0</td>
<td>0.32.1</td>
<td>Semantic version data type</td>
</tr>
</tbody>
</table>
```
Package Management

```
pgxn upgrade -v theory/semver
```

Upgrading theory/semver 0.32.1 in pgdb-yum/16.3...
Found theory-semver-0.32.1+1.pg16+2.linux-arm64.zip
Unpacking theory-semver-0.32.1+1.pg16+2.linux-arm64.zip
Installing lib/semver.so in /usr/lib/postgresql/16/lib
Installing share/semver-0.32.1.sql in /usr/share/postgresql/16/extensions/
Installing share/semver.control in /usr/share/postgresql/16/extensions/
Installing doc/semver.md in /usr/share/doc/postgresql-doc-16

Done!

Also macOS, *BSD, Windows,
### Package Management

```bash
pgxn package ls
```

Packages installed in pgdb-yum/16.3:

| package          | installed | latest | description                                                      |
|------------------|-----------|--------|******************************************************************|
| core/citext      | 1.6.0     | ✅      | data type for case-insensitive character strings               |
| core/jsonb_plperl| 1.0.0     | ✅      | transform between jsonb and plperl                              |
| core/unaccent    | 1.1.0     | ✅      | text search dictionary that removes accents                     |
| theory/semver    | 0.32.0    | ✅      | Semantic version data type                                     |
This Release: semver 0.32.1
Date: 2023-08-01
Status: Stable
Other Releases: 0.32.1 — 2023-08-01

Abstract: A semantic version data type
Description: A Postgres data type for the Semantic Version format with support for btree and hash indexing.
Released By: theory
License: PostgreSQL
Resources: git → repo → bugs
Special Files: Changes → LICENSE → META.json → Makefile → semver.control
Packagers: PGDB Yum → PGDB Deb → Trunk → Homebrew → Chocolatey → PGXMan
Tags: semantic version → semver → version → version number
Needs

- Reliable index of extensions
- Effective search
- Curation & categorization
- Quality signals
- Version support info
- Quick & easy installation
- Informative documentation
**Delivery**

- Canonical Registry
  - Discovery
  - Quality Signals
  - Category curation
  - Links & easy install instructions

**Installation**

- One-line command
- Fast binary installation
- Many platforms
- Many packages (Postgres installs)
- OS packaging support

**Documentation**

- Getting started with extensions
- How to use community tools
- Beautiful extension documentation

**Usage**

- Documentation
- Tested examples
- Tutorials & references
<table>
<thead>
<tr>
<th>rating</th>
<th>package</th>
<th>version</th>
<th>date</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>hadi/json_fdw</td>
<td>1.0.0</td>
<td>2013-11-18</td>
<td>Foreign Data Wrapper for JSON files</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>theory/jsonschema</td>
<td>0.1.0</td>
<td>2024-04-30</td>
<td>JSON Schema validation functions</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>jma/yamltodb</td>
<td>0.10.0</td>
<td>2022-11-08</td>
<td>Database change management in YAML</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>shanekilkelly/bedquilt</td>
<td>2.1.0</td>
<td>2016-09-12</td>
<td>A json document store</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>furstenheim/is_jsonb_valid</td>
<td>0.0.1</td>
<td>2017-08-17</td>
<td>JSONB validation</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>postgrespro/jsonb_schema</td>
<td>1.0.0</td>
<td>2020-08-24</td>
<td>Decouple jsonb schema from data</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
<td>kungfury/jsoncdc</td>
<td>0.1.0</td>
<td>2017-11-23</td>
<td>Translates Postgres WAL to JSON</td>
</tr>
</tbody>
</table>

**Package Management**

`pgxn search json`
Installing theory/jsonschema 0.1.0 into pgenv/16.3…
Found theory-jsonschema-0.1.1+2.pg16+1.darwin-arm64.zip
Unpacking theory-jsonschema-0.1.1+2.pg16+1.darwin-arm64.zip
Installing lib/jsonschema.dylib in ~/.pgenv/pgsql-16.3/lib/
Installing share/jsonschema—0.1.0sql in ~/.pgenv/pgsql-16.3/share/extensions/
Installing share/jsonschema.control in ~/.pgenv/pgsql-16.3/share/extensions/
Installing doc/jsonschema.md in ~/.pgenv/pgsql-16.3/share/doc/

Done!
## Package Management

```bash
❯ pgxn package ls
```

Packages installed in pgenv/16.3:

<table>
<thead>
<tr>
<th>package</th>
<th>installed</th>
<th>latest</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>core/citext</td>
<td>1.6.0</td>
<td>✅</td>
<td>data type for case-insensitive character strings</td>
</tr>
<tr>
<td>core/jsonb_plperl</td>
<td>1.0.0</td>
<td>✅</td>
<td>transform between jsonb and plperl</td>
</tr>
<tr>
<td>core/unaccent</td>
<td>1.1.0</td>
<td>✅</td>
<td>text search dictionary that removes accents</td>
</tr>
<tr>
<td>theory/jsonschema</td>
<td>0.1.0</td>
<td>✅</td>
<td>JSON Schema validation functions</td>
</tr>
</tbody>
</table>
INDUSTRY
INDUSTRY

Needs

Embrace and extend

  Extension curation products
  Extension rating products

Linked from root registry metadata

Offer extensions for customers

Use APIs to provide UX for DBaaS products

Immutability
Industrial Sector

Delivery

Specialized Registries

Additional metadata, stats, reports, signal

Binary Registry Federation

Curate extensions for customers by copying only approved extensions to own registry

Use APIs to provide UX for DBaaS products

Report back stats

Separate mutable extension directory
Relies on Community APIs
THE VISION
PGXN V2 FEATURES

Canonical source registry
Search & discover all extensions
Streaming & federation
Quality metrics & reports
Binary packaging & distribution
  Multi-platform & multi-version
Simple installation & package management
ARCHITECTURE SKETCH
JOIN US

Background: https://wiki.postgresql.org/wiki/PGXN_v2
Architecture: https://wiki.postgresql.org/wiki/PGXN_v2/Architecture
Project: https://github.com/orgs/pgxn/projects/1
Tasks: https://github.com/pgxn/planning/issues
Writing: https://justathtory.com/tags/pgxn/
Slack: https://pgtreats.info/slack-invite
THE FUTURE OF THE POSTGRES EXTENSION ECOSYSTEM

DAVID E. WHEELER
PGXN, TEMBO

THANK YOU