



# Richard Nyström



---

## Simple API with Nginx and PostgreSQL

26 Jul 2013

How to build a simple REST API using only Nginx and PostgreSQL.

Sometimes it's overkill to use a web framework if you only need to develop a very simple REST API. It turns out that Nginx can be used to develop a full fledged REST API and PostgreSQL can easily be used for persistence.

In this blog post I'm going to show you how to create a simple CRUD API for articles.

### Setup

I recommend that you use the [OpenResty \(http://openresty.org\)](http://openresty.org) to install Nginx. It contains the standard Nginx core and lots of 3rd-party Nginx modules including the [Postgres upstream module \(https://github.com/FRiCKLE/nginx\\_postgres/\)](https://github.com/FRiCKLE/nginx_postgres/) that allows Nginx to communicate with a PostgreSQL database. OpenResty is not an Nginx fork, just a software bundle so there's nothing to worry about.

This is how I installed and compiled OpenResty on my Mac:

```
brew install pcre

tar xzvf ngx_openresty-1.4.1.1.tar.gz

cd ngx_openresty-1.4.1.1/

./configure \
--with-cc-opt="-I/usr/local/Cellar/pcre/8.33/include" \
--with-ld-opt="-L/usr/local/Cellar/pcre/8.33/lib" \
--with-http_postgres_module
```

Remember to change pcre version number to the one you have installed. It might differ.

For this blog post I used PostgreSQL 9.2. You can install PostgreSQL from [Homebrew \(http://brew.sh\)](http://brew.sh) or use [Postgres.app \(http://postgresapp.com\)](http://postgresapp.com).

## Create the database

```
CREATE DATABASE articledb WITH OWNER username ENCODING 'UTF8';

CREATE TABLE articles (
    id serial PRIMARY KEY,
    title varchar(50) NOT NULL,
    body varchar(32000) NOT NULL,
    created_at timestamp DEFAULT current_timestamp
);

# add a few rows:
INSERT INTO articles (title, body) VALUES ('Test title 1', 'Test body 1');
INSERT INTO articles (title, body) VALUES ('Test title 2', 'Test body 2');
INSERT INTO articles (title, body) VALUES ('Test title 3', 'Test body 3');
```

## The complete nginx.conf

```

worker_processes 8;

events {}

http {
    upstream database {
        postgres_server 127.0.0.1 dbname=articedb user=username password=yourpass;
    }

    server {
        listen      8080;
        server_name localhost;

        location /articles {
            postgres_pass database;
            rds_json on;
            postgres_query    HEAD GET  "SELECT * FROM articles";

            postgres_escape $title $arg_title;
            postgres_escape $body  $arg_body;
            postgres_query
                POST "INSERT INTO articles (title, body) VALUES($title, $body) RETURNING *";
            postgres_rewrite  POST changes 201;
        }

        location ~ /articles/(?<id>\d+) {
            postgres_pass database;
            rds_json on;
            postgres_escape $escaped_id $id;
            postgres_query    HEAD GET  "SELECT * FROM articles WHERE id=$escaped_id";
            postgres_rewrite  HEAD GET  no_rows 410;

            postgres_escape $title $arg_title;
            postgres_escape $body  $arg_body;
            postgres_query
                PUT "UPDATE articles SET title=$title, body=$body WHERE id=$escaped_id RETURNING *";
            postgres_rewrite  PUT no_changes 410;

            postgres_query    DELETE  "DELETE FROM articles WHERE id=$escaped_id";
            postgres_rewrite  DELETE  no_changes 410;
            postgres_rewrite  DELETE  changes 204;
        }
    }
}

```

# Test drive the API

*Get all articles:*

```
curl http://localhost:8080/articles
```

*Create a new article:*

```
curl -X POST http://localhost:8080/articles?title=Article1&body=body1
```

*Update article:*

```
curl -X PUT http://localhost:8080/articles/1?title=Article2&body=body2
```

*Delete article:*

```
curl -X DELETE http://localhost:8080/articles/1
```

---

AROUND THE WEB

WHAT'S THIS?

**Lifefactopia**

How to Buy Must-Have Products for Next to Nothing

**Healthy LifeStyle**

Controversial "Skinny Pill" Sweeps the Nation

**CNN**

CNN Money: How Young Millionaires Invest

**RealDose Nutrition**

4 Hormones All Skinny People Have In Balance

ALSO ON RICHARD NYSTRÖM

[PostgreSQL indexing in Rails](#) 0 comments

[Custom fields in Rails using PostgreSQL](#)

## Tagging in Rails 4 using PostgreSQL arrays

13 comments

## Use UUIDs in Rails 4 with PostgreSQL 4 comments

9 Comments

Richard Nyström

Login

Sort by Best

Share Favorite



Join the discussion...



**Richard Nyström** Mod · a year ago

I just added commenting to my blog. There is already a lot of comments on Hacker News about the blog post: <https://news.ycombinator.com/i...>

4 ^ | v · Reply · Share



**Francisco Zelaya** → Richard Nyström · a year ago

How do you compare Nginx to Apache?

^ | v · Reply · Share



**Francisco Zelaya** → Richard Nyström · a year ago

May I suggest that you add some snapshots to your end work, that will help visualize better your nice work.

^ | v · Reply · Share



**Manuel Gonzalez** · a year ago

Awesome post, thank you.

1 ^ | v · Reply · Share



**Mike Quentel** · 8 months ago

Many thanks for sharing this...pre-reqs for OpenResty on my instance of Ubuntu Desktop: I had to also install libpq, in addition to pcre. Also, had to include in the ngx\_openresty-1.4.3.6 configure flag --with-luajit

^ | v · Reply · Share



**Larsie** · 9 months ago

I just learnt that Postgres has some internal JSON support, but no HTTP access. Is that correct? How would your example relate to the JSON support in postgres? Does it use it or just store the JSON as a text field?

^ | v · Reply · Share



**eugeneware** · a year ago

Hi, thanks for the great example. You use HTTP GET variables in your example. How could



Hi, thanks for the great example. You use HTTP GET variables in your example. How could you modify your example to pass HTTP POST variables through to postgres\_query?

^ | v • Reply • Share ›



**Hakim Benoudjit** • a year ago

Thanks for the tutorial!

---

## Contact me

[Github \(https://github.com/ricn\)](https://github.com/ricn) • [Twitter \(https://twitter.com/richardnystrom\)](https://twitter.com/richardnystrom) • [Linkedin \(http://se.linkedin.com/in/richardnystrom\)](http://se.linkedin.com/in/richardnystrom) • E-mail (<mailto:ricny046@gmail.com>)

---