

# SQLx

SQLx is a SQL framework for Rust, which allows to write SQL queries and check them on compile time. It also comes with a migration logic to update your database.

## Install SQLx

To install the SQLx, run the following command in the terminal

```
cargo install sqlx-cli
```

In your Rust project, install sqlx using the following command

```
cargo add sqlx --features postgres,runtime-tokio

# Instead of postgres, you can also use the following databases:
# postgres, mysql, mssql, sqlite

# Also instead of runtime-tokio, you can use:
# runtime-tokio, runtime-async-std

# For actix-web projects, use runtime-tokio
```

## Database Migration

To create a migration, you need to set the environment variable `DATABASE_URL`, then execute the following command to create the database:

```
# Create the database
sqlx database create
```

To create a migration step, execute the following command. Replace "name" with a description in snake case, which describes the migration step.

```
# Add a migration, replace name
sqlx migrate add -r name

# Or using sequential numbers instead of timestamps
sql migrate add -r -s name
```

The flag `-r` will create two migration files, one with the pattern `timestamp_name.up.sql` and one with the name `timestamp_name.down.sql` - the one is the migration and the second is the revert script. Without it, it will only create a `timestamp_name.sql` file. You can also use the flag `-s`, then instead of a timestamp a sequence will be used (`0001_name.up.sql`).

```
# Execute migration
sqlx migrate run
```

## Migrate per code

You can embed the migrations into the application, so that the migration will be executed on startup:

```
let database_url = std::env::var("DATABASE_URL").expect("DATABASE_URL must be set");

let pool = PgPoolOptions::new()
    .max_connections(5)
    .connect(&database_url)
    .await
    .expect("Failed to connect to database");

sqlx::migrate!()
    .run(&pool)
    .await
    .expect("Failed to run migrations");
```

---

Revision #5

Created 2026-03-06 08:19:01 UTC by Matthias

Updated 2026-03-06 11:40:17 UTC by Matthias