

付録 A

各種ソフトウェアのアップグレード

この付録では本書の学習に必要な各種ソフトウェアをアップグレードする手順を解説します。具体的には、Erlang/OTP、Elixir、Phoenix、nvm、Node.jsなどを本書で使用するバージョンに更新します。

A.1 Erlang/OTP 21.1 のインストールと有効化

ターミナルで次のコマンド群を順に実行してください。

```
$ kerl update releases
$ kerl build 21.1 21.1
$ kerl install 21.1 ~/erlang/21.1
$ source ~/erlang/21.1/activate
```

インターネット接続状況にもよりますが、通常 Erlang/OTP のソースコードをダウンロードするのに 1 時間程度かかります。

■コラム: *kerl* コマンド

kerl は、バージョン番号を指定して Erlang/OTP をインストールしたり、複数の Erlang/OTP を切り替えて使用したりするためのツールです。『初級①』の第 2 章で導入しました。

なお、『初級①』の第 2 章は初版刊行後に大幅改訂されています。『Elixir/Phoenix 初級①』の読者サポートページ（下記 URL）から第 2 章の PDF ファイルを入手してください。

<https://www.oiax.jp/books/elixir-phoenix-volume1.html>

A.2 Bash 設定ファイルの変更

macOS の場合

『初級①』から学習を続けている場合、`.bash_profile` の末尾付近に次のような記述があるはずですが、

```
export PATH="$HOME/erlang/19.3/bin:$PATH"
```

テキストエディタでこの行を削除してください。

そして、ターミナルで次のコマンドを実行してください。

```
$ echo 'export PATH="$HOME/erlang/21.1/bin:$PATH"' >> ~/.bash_profile
```

Ubuntu の場合

『初級①』から学習を続けている場合、`.bashrc` の末尾付近に次のような記述があるはずですが、

```
export PATH="$HOME/erlang/19.3/bin:$PATH"
```

テキストエディタでこの行を削除してください。

A.3 Elixir、Hex、rebar のインストール

そして、ターミナルで次のコマンドを実行してください。

```
$ echo 'export PATH="$HOME/erlang/21.1/bin:$PATH"' >> ~/.bashrc
```

A.3 Elixir、Hex、rebar のインストール

ターミナルで次のコマンド群を順に実行してください。

```
$ kiex install 1.7.4
$ kiex use 1.7.4 --default
$ mix local.hex --force
$ mix local.rebar --force
```

kiex はバージョン番号を指定して Elixir をインストールしたり、複数の Elixir を切り替えて使用したりするためのツールです。『初級①』の第 2 章で導入しています。*mix* は Elixir に標準で添付されているツールです。Elixir のライブラリをインストールしたり、データベース関連のさまざまなタスクを実行するために利用します。

A.4 バージョン番号の確認

ターミナルで次のコマンドを実行してください。

```
$ mix hex.info
```

次のような結果が出力されます。

```
Hex:      0.18.2
Elixir:   1.7.4
OTP:      21.1

Built with: Elixir 1.7.4 and OTP 19.3
```

3 行目に表示されている数字が Erlang/OTP のバージョン番号です。Elixir のバージョン番号は 2 行目に出力されています。最後の行に出力される Elixir

付録A 各種ソフトウェアのアップグレード

と OTP のバージョン番号は、これらのバージョン番号と異なる場合があります。

A.5 Phoenix のインストール

ターミナルで次のコマンド群を順に実行してください。

```
$ TEMP=https://github.com/phoenixframework/archives/raw/master  
$ mix archive.install $TEMP/phoenix_new-1.2.4.ez --force
```

バージョン番号を確認しましょう。

```
$ mix phoenix.new --version
```

次のような結果が出力されれば OK です。

```
Phoenix v1.2.4
```

2018 年 12 月 2 日現在、Phoenix の最新版は 1.4.0 ですが、本書はバージョン 1.2 に基づいて書かれておりますので、少なくとも本書の学習を進める間は 1.2 系を使用してください。ただし、Phoenix 1.2 系の最新版である 1.2.5 を使用した場合、現在の環境では依存パッケージ群を正常にインストールできません。そこで、ひとつ前のバージョン 1.2.4 を利用してください。Phoenix はバージョン 1.2 から 1.3 でディレクトリ構成やモジュールの命名規則が大きく変わっています。バージョン 1.3 以降の Phoenix をインストールしてしまうと学習を進めるための大きな障害となります。

A.6 *nvm* のアップグレード

macOS の場合

ターミナルで以下のコマンドを順に実行してください。

```
$ touch ~/.bash_profile
```

```
$ cd /tmp
$ NVMURL=https://rawgit.com/creationix/nvm/v0.33.11/install.sh
$ wget --no-check-certificate $NVMURL
$ bash install.sh
$ rm -f install.sh
$ source ~/.bash_profile
```

4行目で `Unable to establish SSL connection.` というエラーが出ることがあります。その場合は、次のコマンドを実行して `wget` をバージョンアップしてから、もう一度やり直してみてください。

```
$ brew upgrade wget
```

Ubuntu の場合

ターミナルで以下のコマンドを順に実行してください。

```
$ touch ~/.bashrc
$ cd /tmp
$ NVMURL=https://rawgit.com/creationix/nvm/v0.33.11/install.sh
$ wget --no-check-certificate $NVMURL
$ bash install.sh
$ rm -f install.sh
$ source ~/.bashrc
```

A.7 Node.js のインストール

ターミナルで次のコマンドを実行してください。

```
$ nvm install --lts
```

オプション `--lts` は長期サポート (Long Time Support) 版を選択するためのものです。2018年12月2日現在、最新の Node.js 推奨版 v10.14.1 がインス

付録 A 各種ソフトウェアのアップグレード

トールされます。

Node.js のバージョンを確認するには、次のコマンドを実行します。

```
$ node --version
```

バージョン番号として v10.14.1 のような文字列が出力されれば OK です。

付録 B

PostgreSQL のセットアップ

この付録では、PostgreSQL のインストール、初期化、起動など Phoenix 開発に必要な準備作業の手順を解説します。手順は macOS と Ubuntu で大きく異なります。

【重要な注意事項】 この付録に記載されたコマンドは、特記ない限り一般ユーザーの権限で (sudo なしで) 実行してください。むやみに sudo を付けてコマンドを実行するとさまざまな問題を引き起こします。

B.1 macOS の場合

PostgreSQL のインストール

ターミナルで以下のコマンドを順に実行します。

```
$ brew install postgresql
```

2018 年 12 月 2 日現在、PostgreSQL 10.6 がインストールされます。

付録B PostgreSQLのセットアップ

ここで `bash: brew: command not found` というメッセージが出力された場合は、Homebrew のインストールが必要です。『Elixir/Phoenix 初級①』の付録 C を参照してください。付録 C の PDF 版は、本書の読者サポートページ (<http://www.oiax.jp/books/elixir-phoenix-volume2.html>) から入手可能です。

PostgreSQL サーバーの起動

ターミナルで次のコマンドを実行します。

```
$ pg_ctl start -D /usr/local/var/postgres/
```

すると、次のようなメッセージが表示されます。

```
server starting
...
Log: database system is ready to accept connections
Log: autovacuum launcher started
```

最後に Enter キーを入力してください。

なお、以前に古いバージョンの PostgreSQL を使用していた場合、サーバー起動時に次のようなエラーメッセージが出る場合があります。

```
FATAL: database files are incompatible with server
DETAIL: The data directory was initialized by PostgreSQL version 9.5, >
which is not compatible with this version 10.5.
```

この場合は、以下のコマンドを実行してからサーバーの起動をやり直してください。

```
$ mv /usr/local/var/postgres /usr/local/var/postgres.bak
$ initdb /usr/local/var/postgres -E utf8
```

今後、Mac へのログイン時に PostgreSQL サーバーを自動で起動したい場合は、ターミナルで次のコマンドを実行してください。


```
$ brew services start postgresql
```

次のようなメッセージが表示されれば成功です。

```
==> Successfully started `postgresql` (label: homebrew.mxcl.postgresql)
```

逆に、PostgreSQL サーバーの自動起動設定を解除したい場合は、PostgreSQL サーバーが起動している状態でターミナルで次のコマンドを実行してください。

```
$ brew services stop postgresql
```

次のようなメッセージが表示されれば成功です。

```
==> Successfully stopped `postgresql` (label: homebrew.mxcl.postgresql)
```

phoenix ユーザーの追加

```
$ createuser -dP phoenix
```

createuser は PostgreSQL サーバーにユーザーを追加するコマンドです。オプション `-d` は「データベース作成権限を与える」という意味で、オプション `-P` は「パスワードを設定する」という意味です。

ここで、次のようにパスワードの入力を求められます。

```
Enter password for new role:
```

これは、PostgreSQL サーバーに phoenix ユーザーとしてログインするためのパスワードです。ここでは phoenix と入力することにします。さらに、

```
Enter it again:
```

と表示されたら、同じく phoenix と入力してください。

■ コラム: ユーザー追加に失敗する場合

`createuser` コマンドが次のようなエラーメッセージを出して失敗することがあります:

```
createuser: could not connect to database postgres: FATAL: role "xxxx" >  
does not exist
```

ただし、xxxxの部分には、現在あなたがMacにログインしているユーザー名が入っています。

このエラーは、一台のMacに複数ユーザーを作成し、ユーザーを切り替えて使っている場合に主に発生します。例えば、あなたのMacにaliceとbobというふたつのユーザーが存在していて、aliceユーザーとしてHomebrewを用いてPostgreSQLをインストールした場合、bobユーザーとしてMacにログインしている状態で`createuser`コマンドを用いると上記のようなエラーが出ます。

問題を解決するには、aliceユーザーとしてMacにログインし直して、次のコマンドを実行してください。

```
$ createuser -s bob
```

オプション `-s` は「スーパーユーザー権限を与える」という意味です。この結果、bobユーザーとして`createuser`コマンドを実行できるようになります。

PostgreSQL サーバーの停止

PostgreSQL サーバーを停止する前に、Phoenix アプリケーションを停止してください。

ターミナルで次のコマンドを実行します。

```
$ pg_ctl stop -D /usr/local/var/postgres/
```

次のようなメッセージが表示されれば、サーバーの停止が完了しています。

```
waiting for server to shut down...LOG: received smart shutdown request
LOG: autovacuum launcher shutting down
LOG: shutting down
LOG: database system is shut down
done
server stopped
```

ただし、Mac からログアウトするときに PostgreSQL サーバーは自動で停止するので、通常はサーバーを明示的に停止する必要はありません。

B.2 Ubuntu 18.04 の場合

PostgreSQL のインストール

ターミナルで次のコマンドを実行します。

```
$ sudo apt-get install -y postgresql libpq-dev
```

macOS の場合と異なり、インストール後に自動で PostgreSQL サーバーが起動します。また、Ubuntu の起動時にも PostgreSQL サーバーが自動的に起動します。

念のため、バージョン番号を調べます。

```
$ psql --version
```

次のように出力されれば OK です（バージョン番号は異なるかもしれません）。

```
psql (PostgreSQL) 10.6
```

`psql` コマンドを用いて、PostgreSQL サーバーに接続できることを確認します。

```
$ sudo -u postgres psql
```

接続に成功すると、次のように PostgreSQL コンソールが開きます。

```
psql (10.6 (Ubuntu 10.6-0ubuntu0.18.04.1))
Type "help" for help.

postgres=#
```

Ctrl-D を押すと PostgreSQL コンソールが終了します。

phoenix ユーザーの追加

```
$ sudo -u postgres createuser -dP phoenix
```

createuser は PostgreSQL サーバーにユーザーを追加するコマンドです。オプション `-d` は「データベース作成権限を与える」という意味で、オプション `-P` は「パスワードを設定する」という意味です。

ここで、(sudo コマンドの実行に必要なパスワードの他に) 次のようにパスワードの入力を求められます。

```
Enter password for new role:
```

これは、PostgreSQL サーバーに phoenix ユーザーとしてログインするためのパスワードです。ここでは phoenix と入力することにします。さらに、

```
Enter it again:
```

と表示されたら、同じく phoenix と入力してください。

PostgreSQL サーバーの起動と停止

PostgreSQL サーバーは Ubuntu の起動時に自動的に起動し、シャットダウン時に自動的に停止します。

自動起動の設定を解除するには、次のコマンドを実行します。

```
$ sudo systemctl disable postgresql
```

自動起動する設定に戻すには、次のコマンドを実行します。

```
$ sudo systemctl enable postgresql
```

手動で起動するには、次のコマンドを実行します。

```
$ sudo systemctl start postgresql
```

手動で停止するには、次のコマンドを実行します。

```
$ sudo systemctl stop postgresql
```


付録 C

MySQL のセットアップ

この付録では、MySQL のインストール、初期化、起動など Phoenix 開発に必要な準備作業の手順を解説します。手順は macOS と Ubuntu で大きく異なります。

【重要な注意事項】 この付録に記載されたコマンドは、特記ない限り一般ユーザーの権限で (sudo なしで) 実行してください。むやみに sudo を付けてコマンドを実行するとさまざまな問題を引き起こします。

C.1 macOS の場合

MySQL 5.7 のインストール

ターミナルで以下のコマンドを順に実行します。

```
$ brew install mysql
```

ここで `bash: brew: command not found` というメッセージが出力された場合は、Homebrew のインストールが必要です。『Elixir/Phoenix 初級①』の付録 C を参照してください。

MySQL サーバーの起動

ターミナルで次のコマンドを実行します。

```
$ mysql.server start
```

すると、次のようなメッセージが表示されます。

```
Starting MySQL
.. SUCCESS!
```

MySQL サーバーの停止

ターミナルで次のコマンドを実行します。

```
$ mysql.server stop
```

すると、次のようなメッセージが表示されます。

```
Shutting down MySQL
.. SUCCESS!
```

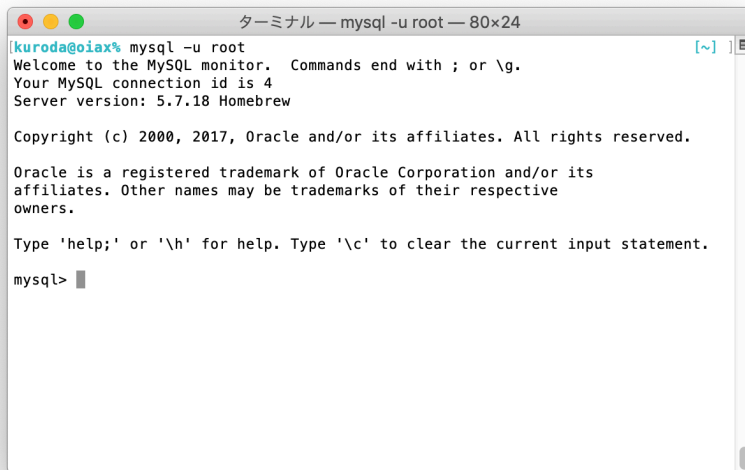
phoenix ユーザーの追加

MySQL コンソールにログインします。

```
$ mysql -u root
```

Homebrew で MySQL をインストールした場合、root ユーザーとしてパスワードなしでログインできます。

ターミナルが図 C.1 のような表示になればログイン成功です。



```
ターミナル — mysql -u root — 80x24
[kuroda@oiax% mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.18 Homebrew

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

図 C.1 MySQL コンソール

プロンプト (mysql>) に対して、次のように入力してください。

```
GRANT ALL on *.* TO phoenix@localhost IDENTIFIED BY 'phoenix';
FLUSH PRIVILEGES;
```

Ctrl-D を押して、MySQL コンソールを抜けてください。そして、**phoenix** ユーザーとして MySQL コンソールにログインします。root ユーザーとしてログインする場合と異なり、パスワード入力をするためオプション **-p** を付けてください。

```
$ mysql -u phoenix -p
```

プロンプトに対して、次のように入力してください。

```
CREATE DATABASE dummy;
```

付録C MySQLのセットアップ

次のメッセージが出力されれば、成功です。

```
Query OK, 1 row affected (0.00 sec)
```

さらに、プロンプトに次のように入力して、データベース `dummy` を削除します。

```
DROP DATABASE dummy;
```

削除に成功すれば次のように入力されます。

```
Query OK, 0 row affected (0.00 sec)
```

Ctrl-D を押して、MySQL コンソールを抜けてください。

MySQL サーバーの自動起動設定

Mac にログインしたときに自動で MySQL サーバーを起動する設定にするには、ターミナルで次のコマンドを実行します。

```
$ brew services start mysql
```

すると、次のようなメッセージが表示されます。

```
==> Successfully started `mysql` (label: homebrew.mxcl.mysql)
```

逆に、MySQL サーバーの自動起動設定を解除するには、ターミナルで次のコマンドを実行します。

```
$ brew services stop mysql
```

すると、次のようなメッセージが表示されます。

```
==> Successfully stopped `mysql` (label: homebrew.mxcl.mysql)
```

C.2 Ubuntu 18.04 の場合

MySQL 5.7 のインストール

ターミナルで以下のコマンドを順に実行します。

```
$ sudo apt-get -y install mysql-server libmysqlclient-dev
```

macOS の場合と異なり、インストール後に自動で MySQL サーバーが起動します。その後は、Ubuntu の起動時にも MySQL サーバーが自動的に起動します。

phoenix ユーザーの追加

MySQL コンソールにログインします。パスワードを尋ねられますが、何も入力せずに Enter キーを押してください。

```
$ sudo mysql -u root -p
```

ターミナルが図 C.2 のような表示になればログイン成功です。

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.24-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

図 C.2 MySQL コンソール

プロンプト (mysql>) に対して、次のように入力してください。

付録C MySQLのセットアップ

```
GRANT ALL on *.* TO phoenix@localhost IDENTIFIED BY 'phoenix';  
FLUSH PRIVILEGES;
```

Ctrl-D を押して、MySQL コンソールを抜けてください。そして、phoenix ユーザーとして MySQL コンソールにログインします（コマンドの先頭に `sudo` を付けずに実行してください）。

```
$ mysql -u phoenix -p
```

プロンプトに対して、次のように入力してください。

```
CRATE DATABASE dummy;
```

次のメッセージが出力されれば、成功です。

```
Query OK, 1 row affected (0.00 sec)
```

さらに、プロンプトに次のように入力して、データベース `dummy` を削除します。

```
DROP DATABASE dummy;
```

削除に成功すれば次のよう出力されます。

```
Query OK, 0 row affected (0.00 sec)
```

Ctrl-D を押して、MySQL コンソールを抜けてください。

MySQL サーバーの起動と停止

MySQL サーバーは Ubuntu の起動時に自動的に起動し、シャットダウン時に自動的に停止します。

自動起動の設定を解除するには、次のコマンドを実行します。

```
$ sudo systemctl disable mysql.service
```

自動起動する設定に戻すには、次のコマンドを実行します。

```
$ sudo systemctl enable mysql.service
```

手動で起動するには、次のコマンドを実行します。

```
$ sudo systemctl start mysql.service
```

手動で停止するには、次のコマンドを実行します。

```
$ sudo systemctl stop mysql.service
```