文档正在建设中，部分内容可能缺失或缺少翻译，详情查看 文档更新日志

环境变量

如何设置环境变量?

Git 源码或 create-nocobase-app 安装方式

在项目根目录下的 `.env` 文件里设置环境变量，修改环境变量之后需要 kill 应用进程，重新启动。

Docker 安装方式

修改 `docker-compose.yml` 配置，在 `enviroment` 参数里设置环境变量。示例：

```yaml
services:
  app:
    image: nocobase/nocobase:latest
    environment:
      - APP_ENV=production
```

也可以使用 `env_file`，即可在 `.env` 文件中设置环境变量。示例：

```yaml
services:
  app:
    image: nocobase/nocobase:latest
    env_file: .env
```

修改环境变量之后，需要重建 app 容器。

```
docker-compose up -d app
```

全局环境变量

TZ

用于设置应用的时区，默认为操作系统时区。

https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

> ⚠ WARNING
>
> 与时间相关的操作会依据该时区进行处理，修改 TZ 可能会影响数据库里的日期值，详情查看「日期 & 时间概述」

APP_ENV

应用环境，默认值 `development` ，可选项包括：

- `production` 生产环境
- `development` 开发环境

```
APP_ENV=production
```

APP_KEY

应用的密钥，用于生成用户 token 等，修改为自己的应用密钥，并确保不对外泄露

> ⚠ WARNING
>
> 如果 APP_KEY 修改了，旧的 token 也会随之失效

```
APP_KEY=app-key-test
```

APP_PORT

应用端口，默认值 `13000`

```
APP_PORT=13000
```

API_BASE_PATH

NocoBase API 地址前缀，默认值 `/api/`

```
API_BASE_PATH=/api/
```

API_BASE_URL

CLUSTER_MODE

> `v1.6.0+`

多核（集群）启动模式，如配置了该变量，会透传至 `pm2 start` 命令中作为 `-i <instances>` 的参数。可选项与 pm2 `-i` 参数一致（参考 PM2: Cluster Mode），包括：

- `max` ：使用 CPU 最大核数
- `-1` ：使用 CPU 最大核数 -1

- `<number>` ：指定核数

默认值为空，代表不开启。

> ⚠ 注意
>
> 该模式需要配合集群模式相关的插件使用，否则应用的功能可能出现异常。

更多可参考：集群模式。

PLUGIN_PACKAGE_PREFIX

插件包名前缀，默认为： `@nocobase/plugin-,@nocobase/preset-` 。

例如，添加 `hello` 插件到 `my-nocobase-app` 项目，插件的完整包名则为 `@my-nocobase-app/plugin-hello` 。

PLUGIN_PACKAGE_PREFIX 可以配置为：

```
PLUGIN_PACKAGE_PREFIX=@nocobase/plugin-,@nocobase-preset-,@my-nocobase-app/plugin-
```

则插件名称和包名对应关系如下：

- `users` 插件的包名为 `@nocobase/plugin-users`
- `nocobase` 插件的包名为 `@nocobase/preset-nocobase`
- `hello` 插件的包名为 `@my-nocobase-app/plugin-hello`

DB_DIALECT

数据库类型，默认值 `sqlite` ，可选项包括：

- `sqlite`

- `mariadb`
- `mysql`
- `postgres`

```
DB_DIALECT=mysql
```

DB_STORAGE

数据库文件路径（使用 SQLite 数据库时配置）

```
# 相对路径
DB_STORAGE=storage/db/nocobase.db
# 绝对路径
DB_STORAGE=/your/path/nocobase.db
```

DB_HOST

数据库主机（使用 MySQL 或 PostgreSQL 数据库时需要配置）

默认值 `localhost`

```
DB_HOST=localhost
```

DB_PORT

数据库端口（使用 MySQL 或 PostgreSQL 数据库时需要配置）

- MySQL、MariaDB 默认端口 3306
- PostgreSQL 默认端口 5432

```
DB_PORT=3306
```

DB_DATABASE

数据库名（使用 MySQL 或 PostgreSQL 数据库时需要配置）

```
DB_DATABASE=nocobase
```

DB_USER

数据库用户（使用 MySQL 或 PostgreSQL 数据库时需要配置）

```
DB_USER=nocobase
```

DB_PASSWORD

数据库密码（使用 MySQL 或 PostgreSQL 数据库时需要配置）

```
DB_PASSWORD=nocobase
```

DB_TABLE_PREFIX

数据表前缀

```
DB_TABLE_PREFIX=nocobase_
```

DB_UNDERSCORED

数据库表名、字段名是否转为 snake case 风格，默认为 `false` 。如果使用 MySQL（MariaDB）数据库，并且 `lower_case_table_names=1` ，则 DB_UNDERSCORED 必须为 `true`

⚠ WARNING

当 `DB_UNDERSCORED=true` 时，数据库实际的表名和字段名与界面所见的并不一致，如 `orderDetails` 数据库里的是 `order_details`

DB_LOGGING

数据库日志开关，默认值 `off`，可选项包括：

- `on` 打开
- `off` 关闭

```
DB_LOGGING=on
```

NOCOBASE_PKG_USERNAME

Service 平台用户名，用于自动下载和更新插件

NOCOBASE_PKG_PASSWORD

Service 平台密码，用于自动下载和更新插件

LOGGER_TRANSPORT

日志输出方式，多个用 `,` 分隔。开发环境默认值 `console`，生产环境默认值 `console,dailyRotateFile`。可选项：

- `console` - `console.log`
- `file` - 文件
- `dailyRotateFile` - 按天滚动文件

```
LOGGER_TRANSPORT=console,dailyRotateFile
```

LOGGER_BASE_PATH

基于文件的日志存储路径，默认为 `storage/logs` 。

```
LOGGER_BASE_PATH=storage/logs
```

LOGGER_LEVEL

输出日志级别，开发环境默认值 `debug`，生产环境默认值 `info`。可选项：

- `error`
- `warn`
- `info`
- `debug`
- `trace`

```
LOGGER_LEVEL=info
```

数据库日志输出级别为 `debug`，由 `DB_LOGGING` 控制是否输出，不受 `LOGGER_LEVEL` 影响。

LOGGER_MAX_FILES

最大保留日志文件数。

- `LOGGER_TRANSPORT` 为 `file` 时，默认值为 `10`。
- `LOGGER_TRANSPORT` 为 `dailyRotateFile`，使用 `[n]d` 代表天数。默认值为 `14d`。

```
LOGGER_MAX_FILES=14d
```

LOGGER_MAX_SIZE

按大小滚动日志。

- `LOGGER_TRANSPORT` 为 `file` 时，单位为 `byte`，默认值为 `20971520 (20 * 1024 * 1024)`.
- `LOGGER_TRANSPORT` 为 `dailyRotateFile`，可以使用 `[n]k`，`[n]m`，`[n]g`. 默认不配置。

```
LOGGER_MAX_SIZE=20971520
```

LOGGER_FORMAT

日志打印格式，开发环境默认 `console`，生产环境默认 `json`. 可选项:

- `console`
- `json`
- `logfmt`
- `delimiter`

```
LOGGER_FORMAT=json
```

参考：日志格式

CACHE_DEFAULT_STORE

使用缓存方式的唯一标识，指定服务端默认缓存方式，默认值 `memory`，内置可选项：

- `memory`
- `redis`

```
CACHE_DEFAULT_STORE=memory
```

CACHE_MEMORY_MAX

内存缓存项目最大个数，默认值 `2000` 。

```
CACHE_MEMORY_MAX=2000
```

CACHE_REDIS_URL

Redis连接，可选。示例： `redis://localhost:6379`

```
CACHE_REDIS_URL=redis://localhost:6379
```

TELEMETRY_ENABLED

启动遥测数据收集，默认为 `off` .

```
TELEMETRY_ENABLED=on
```

TELEMETRY_METRIC_READER

启用的监控指标采集器，默认为 `console` . 其他值需要参考对应采集器插件注册的名字，如 `prometheus` . 多个使用 `,` 分隔。

```
TELEMETRY_METRIC_READER=console,prometheus
```

TELEMETRY_TRACE_PROCESSOR

启用的链路数据处理器，默认为 `console` . 其他值需要参考对应处理器插件注册的名字。多个使用 `,` 分隔。

```
TELEMETRY_TRACE_PROCESSOR=console
```

实验性环境变量

APPEND_PRESET_LOCAL_PLUGINS

用于附加预置的未激活插件，值为插件包名（package.json 的 name 参数），多个插件英文逗号分隔。

> INFO
>
> 1. 需要确保插件已经下载到本地，并且在 `node_modules` 目录里可以找到，更多内容查看 插件的组织方式。
> 2. 添加了环境变量后，需要在初始化安装 `nocobase install` 或升级 `nocobase upgrade` 后才会在插件管理器页面里显示。

```
APPEND_PRESET_LOCAL_PLUGINS=@my-project/plugin-foo,@my-project/plugin-bar
```

APPEND_PRESET_BUILT_IN_PLUGINS

用于附加内置并默认安装的插件，值为插件包名（package.json 的 name 参数），多个插件英文逗号分隔。

> INFO
>
> 1. 需要确保插件已经下载到本地，并且在 `node_modules` 目录里可以找到，更多内容查看 插件的组织方式。
> 2. 添加了环境变量后，需要在初始化安装 `nocobase install` 或升级 `nocobase upgrade` 时会自动安装或升级插件。

```
APPEND_PRESET_BUILT_IN_PLUGINS=@my-project/plugin-foo,@my-project/plugin-bar
```

临时环境变量

安装 NocoBase 时，可以通过设置临时的环境变量来辅助安装，如：

```
yarn cross-env \
  INIT_APP_LANG=zh-CN \
  INIT_ROOT_EMAIL=demo@nocobase.com \
  INIT_ROOT_PASSWORD=admin123 \
  INIT_ROOT_NICKNAME="Super Admin" \
  nocobase install

# 等同于
yarn nocobase install \
  --lang=zh-CN  \
  --root-email=demo@nocobase.com \
  --root-password=admin123 \
  --root-nickname="Super Admin"

# 等同于
yarn nocobase install -l zh-CN -e demo@nocobase.com -p admin123 -n "Super Admin"
```

INIT_APP_LANG

安装时的语言，默认值 `en-US` ，可选项包括：

- `en-US`
- `zh-CN`

```
yarn cross-env \
  INIT_APP_LANG=zh-CN \
  nocobase install
```

INIT_ROOT_EMAIL

Root 用户邮箱

```
yarn cross-env \
  INIT_APP_LANG=zh-CN \
  INIT_ROOT_EMAIL=demo@nocobase.com \
  nocobase install
```

INIT_ROOT_PASSWORD

Root 用户密码

```
yarn cross-env \
  INIT_APP_LANG=zh-CN \
  INIT_ROOT_EMAIL=demo@nocobase.com \
  INIT_ROOT_PASSWORD=admin123 \
  nocobase install
```

INIT_ROOT_NICKNAME

Root 用户昵称

```
yarn cross-env \
  INIT_APP_LANG=zh-CN \
  INIT_ROOT_EMAIL=demo@nocobase.com \
  INIT_ROOT_PASSWORD=admin123 \
  INIT_ROOT_NICKNAME="Super Admin" \
  nocobase install
```

< 集群模式

插件的安装与升级 >