dotenv

Automatically load your project ENV variables from `.env` file when you `cd` into project root directory.

Storing configuration in the environment is one of the tenets of a twelve-factor app. Anything that is likely to change between deployment environments, such as resource handles for databases or credentials for external services, should be extracted from the code into environment variables.

To use it, add `dotenv` to the plugins array in your zshrc file:

```
plugins=(... dotenv)
```

Usage

Create `.env` file inside your project root directory and put your ENV variables there.

For example:

```
export AWS_S3_TOKEN=d84a83539134f28f412c652b09f9f98eff96c9a
export SECRET_KEY=7c6c72d959416d5aa368a409362ec6e2ac90d7f
export MONGO_URI=mongodb://127.0.0.1:27017
export PORT=3001
```

`export` is optional. This format works as well:

```
AWS_S3_TOKEN=d84a83539134f28f412c652b09f9f98eff96c9a
SECRET_KEY=7c6c72d959416d5aa368a409362ec6e2ac90d7f
MONGO_URI=mongodb://127.0.0.1:27017
PORT=3001
```

You can even mix both formats, although it's probably a bad idea.

Settings

ZSH_DOTENV_FILE

You can also modify the name of the file to be loaded with the variable `ZSH_DOTENV_FILE`. If the variable isn't set, the plugin will default to use `.env`. For example, this will make the plugin look for files named `.dotenv` and load them:

```
# in ~/.zshrc, before Oh My Zsh is sourced:
ZSH_DOTENV_FILE=.dotenv
```

ZSH_DOTENV_PROMPT

Set `ZSH_DOTENV_PROMPT=false` in your zshrc file if you don't want the confirmation message. You can also choose the `Always` option when prompted to always allow sourcing the .env file in that directory. See the next section for more details.

ZSH_DOTENV_ALLOWED_LIST, ZSH_DOTENV_DISALLOWED_LIST

The default behavior of the plugin is to always ask whether to source a dotenv file. There's a **Y**es, **N**o, **A**lways and N**e**ver option. If you choose Always, the directory of the .env file will be added to an allowed list; if you choose Never, it will be added to a disallowed list. If a directory is found in either of those lists, the plugin won't ask for confirmation and will instead either source the .env file or proceed without action respectively.

The allowed and disallowed lists are saved by default in `$ZSH_CACHE_DIR/dotenv-allowed.list` and `$ZSH_CACHE_DIR/dotenv-disallowed.list` respectively. If you want to change that location, change the `$ZSH_DOTENV_ALLOWED_LIST` and `$ZSH_DOTENV_DISALLOWED_LIST` variables, like so:

```
# in ~/.zshrc, before Oh My Zsh is sourced:
ZSH_DOTENV_ALLOWED_LIST=/path/to/dotenv/allowed/list
ZSH_DOTENV_DISALLOWED_LIST=/path/to/dotenv/disallowed/list
```

The file is just a list of directories, separated by a newline character. If you want to change your decision, just edit the file and remove the line for the directory you want to change.

NOTE: if a directory is found in both the allowed and disallowed lists, the disallowed list takes preference, *i.e.* the .env file will never be sourced.

Version Control

It's strongly recommended to add `.env` file to `.gitignore`, because usually it contains sensitive information such as your credentials, secret keys, passwords etc. You don't want to commit this file, it's supposed to be local only.

Disclaimer

This plugin only sources the `.env` file. Nothing less, nothing more. It doesn't do any checks. It's designed to be the fastest and simplest option. You're responsible for the `.env` file content. You can put some code (or weird symbols) there, but do it on your own risk. `dotenv` is the basic tool, yet it does the job.

If you need more advanced and feature-rich ENV management, check out these awesome projects:

- direnv
- zsh-autoenv