zygoat

Apr 01, 2020

Contents

CHAPTER 1

Getting Started

1.1 What is Zygoat?

`zygoat` is an opinionated tool for creating `NextJS`/`Django`/`PostgreSQL` applications. It comes preloaded with a set of project components for spawning a fully functional application, including the deployment stack and docker development configuration.

`zygoat` can also be used to create your own project generation tools, using a set of well-defined and arbitrarily nested components.

1.2 Installation

From `pypi`

```
pip install --user zygoat
```

This adds the `zg` executable to your path, which allows you to interact with `zygoat`'s pre-existing project configuration.

1.3 Usage

```
Usage: zg [OPTIONS] COMMAND [ARGS]...

Options:
-v, --verbose
--help          Show this message and exit.

Commands:
create  Create components without initializing a new project
```

(continues on next page)

(continued from previous page)

```
delete  Calls the delete phase on all included build components
init    Creates a new zygoat settings file and exits
list    Lists all of the running phase names
new     Creates a new settings file and all components
update  Calls the update phase on all included build components
```

CHAPTER 2

Configuration

2.1 zygoat_settings.yml

Upon project creation, `zygoat` will create a `zygoat_settings.yml` file. Currently, this file only contains the name of the project being created and a list of components to exclude.

Example file:

```
name: goat-hugging-instructions
exclude:
  - DockerCompose
  - Backend__DockerFile
  - Frontend__DockerFile
```

CHAPTER 3

Deploying a Zygoat Application

Zygoat leaves the deployment options up to the user, and provides enough flexibility to tolerate most setups built on top of it. The only dependency is that the `frontend` application has an environment variable during build time called `BACKEND_URL` that points to the backend deployment.

3.1 Recommended Deployment Options

We recommend deploying the backend Django application using Zappa serverless, and deploying the frontend NextJS application using Zeit Now.

3.2 Common Setup

The official Zappa documentation recommends using this guide to configure your django deployment and attaching a database to it. Make sure that you're allowed to call out to it from the frontend by configuring CORS properly. The Zygoat NextJS configuration will proxy the CSRF token and cookies back and forth between API requests.

Something that is not mentioned in the Zappa documentation is that if you want an outbound internet connection (such as if your Django app consumes a 3rd party API) you have to configure two private subnets in AWS, where the default route points to a NAT Gateway that routes out through one or more public subnets.

A private subnet in AWS is any subnet where the default route in the route table (`0.0.0.0/0`) points to a NAT Gateway, whereas in a public subnet it points to an internet gateway resource. Ensure that the private subnets are the only ones attached to your Lambda function in your Zappa configuration, and that it is in a security group that allows access to the database.

CHAPTER 4

Advanced Usage

4.1 Components

4.1.1 Base Component Implementation Details

4.1.2 Included types of components

Base Component

File Component

4.1.3 What is a component?

A component is any well structured Python object that implements any out of a set of lifecycle hooks. These hooks are as follows:

- `create` - installs the component from a completely blank state
- `update` - if the component has been changed in the generator since the project was created, this hook will bring it up to date
- `delete` - removes the component from the project
- `list` - prints out the component's unique identifier, for excluding in the config file

A component also contains a method with signature `call_phase(phase, force_create=False)` which will run the phase on this component and all sub components. If the phase is not defined, it moves on to the sub components instead.

Note: When running the `delete` phase, sub-components are called first followed by self. In addition, components should be run in reverse at the top level and are in the default project.

You should never need to override the base implementation of `call_phase`, but if you do, ensure that you maintain that minimum standard of functionality.

4.2 Utility Functions

4.2.1 Files

4.2.2 Shell

4.2.3 Backend

4.3 How Zygoat Generators are Structured

A `zygoat` project generator consists of a set of components, which can be arbitrarily nested. `zygoat` provides a set of utility classes that cover common use cases for various components. The main two are the base `Component` class and the `FileComponent` class for copying files from a resource package to the generated project.

`zygoat` also provides a `SettingsComponent` that uses `redbaron` to allow you to programmatically modify Python files, mostly used for updating Django settings during project creation.