
aiolambda

Release 0.2.12

Dec 24, 2018

Contents:

| | | |
|----------|-----------------------------------|-----------|
| 1 | Quickstart | 1 |
| 1.1 | Requirements: | 1 |
| 1.2 | Create <i>project</i> : | 1 |
| 1.3 | API specs: | 1 |
| 1.4 | Code | 2 |
| 1.5 | Useful commands: | 2 |
| 2 | Template init | 3 |
| 3 | Template update | 5 |
| 4 | Database | 7 |
| 5 | AMQP | 9 |
| 6 | Roadmap | 11 |
| 7 | Aiolambda-cli | 13 |
| 8 | Indices and tables | 15 |

1.1 Requirements:

- **docker**
- **python3**
- **pip3**
- **aiolambda-cli:** *pip install -U aiolambda*

1.2 Create *project*:

Use *aiolambda-cli* to initialize project dir:

```
$ aiolambda-cli init jwt
```

1.3 API specs:

Define microservice [OpenAPI specs](#), for example:

```
/auth/{user_id}:  
  get:  
    tags:  
      - auth  
    summary: Return JWT token  
    operationId: jwt.handlers.auth_handler  
    parameters:  
      - name: user_id  
        description: User unique identifier
```

(continues on next page)

(continued from previous page)

```
in: path
required: true
example: 12
schema:
  type: integer
responses:
  '200':
    description: JWT token
    content:
      'application/json':
        schema:
          type: string
```

1.4 Code

Define your custom handlers referenced in *openAPI specs*. For example, *auth_handler*:

```
async def auth_handler(user_id: str) -> Response:
    return compose(
        generate_token,
        logger.debug,
        return_200)(user_id)
```

1.5 Useful commands:

- *make lint*
- *make run*
- *make test*

See example *jwt* [README.md](#):

CHAPTER 2

Template init

TODO

CHAPTER 3

Template update

CHAPTER 4

Database

TODO

CHAPTER 5

AMQP

TODO

- **Add more examples:**
 - **crud_users: postgres.** Postgres calls. *Document project: [page:database:postgres]
 - **user_settings: mq.** Send and subscribe AMQP messages. *Document project: [page:amqp]
 - **jwt: ci integration.** Basic CI integration with travis. *Document project: [page:ci:travisci]
 - **jwt: container integration.** Basic container integration. *Document project: [page:containers:docker:]
 - **jwt: functional implementations.** Async compose, types: errors, **ROP** documentation. *Document project: [page:functional]
- **Change command definitions:**
 - **inputs:** Command defined with env vars, not cli commands (only use cli commands for short commands [big execution number of times]).
 - **updates:** Makes with inline, replace based ansible filters. (same things: number_of_postgres versions, etc. . . with default values)
 - **.env:** Recomend .env for auto updates and load envvars in each directory (*projects*).
- **Contributions:**
 - **databases:**
 - * redis . . .
 - * mongo . . .
 - **containers:**
 - * firecracker . . .
 - **code analysis:**
 - * code climate
 - **automated api tests:**

- * **dredd**
- aiolambda functional

CHAPTER 7

Aiolambda-cli

```
Usage:
  aiolambda-cli init [--db=<name>] [--mq] [--ci] <project-name>
  aiolambda-cli [-h]
Options:
  -h --help          Show this screen.
  --version          Show version.
  --db=<name>        Database to include: [postgresql].
  --mq              Include RabbitMQ/AMQP client library.
  --ci              Include Travis-CI configuration.
```


CHAPTER 8

Indices and tables

- `genindex`
- `search`