nautilus Documentation

Release v0.2.4

Alec Aivazis

February 09, 2016

Contents:

CHAPTER 1

Getting Started

CHAPTER 2

Services

Services are the building block of nautilus clouds. Very simply, a nautilus service is a standalone process that responds to actions sent over the global event queue, maintains and mutates some internal state according to the action, and provides a summary of that internal state via a GraphQL API.

Nautilus provides extendible implementations of common services as well as a base Service class which all act as good starting points for your own services:

CHAPTER 3

Action Handlers

Action handlers describe how your service mutates its internal state in response to the arrival of an action from the queue. They are defined as a function of two arguments: `type` and `payload`. `Type` is a string that classifies the event and `payload` is a dictionary representing the data associated with the event. For example,

```
def action_handler(action_type, payload):
    # if the payload represents a new recipe to add to the list
    if action_type == 'create_recipe':
        # create a new instance of the recipe model
        recipe = Recipe(**payload)
        # save the new model
        recipe.save()
    # otherwise if the payload is the id of a recipe to be deleted
    elif action_type == 'delete_recipe':
        # find the matching recipe
        recipe = Recipe.query.first(Recipe.id == payload)
        # remove the recipe from the database
        recipe.remove()
```

3.1 Combining Action Handlers

As your services get more complex, you'll want to split your action handler into separate functions which each get called. Nautilus provides a function called `combineActionHandlers` which serves just this case:

```
from nautilus.network import combineActionHandlers

def action_handler1(type, payload):
    print("first handler fired!")

def action_handler2(type, payload):
    print("second handler fired!")

combined_handler = combineActionHandlers(
    action_handler1,
    action_handler2
)
```

3.2 Provided Action Handlers

Nautilus provides some action handlers to mix with your own services when creating custom solutions.

3.2.1 Factories

The following are functions that take a paramter and return an an action creator.

CHAPTER 4

Service API

For the most part, the generation of a Graphql schema takes care of itself thanks to the excellent Graphene community. To reduce the overall boilerplate, Nautilus provides a schema factory that generates a schema to match a model:

4.1 Summarizing an External Service in A Schema

Although a service should never rely on information that it does not maintain, there are very rare cases (like the api gateway) where it is necessary to show another service's data in a schema. In this case, nautilus provides a special base class for the object type that represents remote data.

4.2 Designating a GraphQL Equivalent of an SQLAlchemy Type

Due to the early age of the Graphene/GraphQL community, not all SQLAlchemy types have conversions already specified. If you encounter such a type, consider posting a PR. If that's not your style, you can always register it yourself:

```
from nautilus.models import BaseModel
from nautilus.api import convert_sqlalchemy_type
from sqlalchemy import Column
from sqlalchemy.dialects.postgresql import UUID
from graphene.core.types.scalars import String

class Product(BaseModel):
    id = Column(UUID, primary_key = True)

@convert_sqlalchemy_type.register(UUID)
def convert_column_to_string(type, column):
    return String(description = column.doc)
```

CHAPTER 5

Module Index

5.1 Subpackages

5.1.1 nautilus.admin package

Module contents

5.1.2 nautilus.api package

Subpackages

nautilus.api.fields package

Submodules

nautilus.api.fields.connection module

nautilus.api.fields.list module

Module contents

nautilus.api.filter package

Submodules

nautilus.api.filter.helpers module

Module contents

nautilus.api.objectTypes package

Submodules

nautilus.api.objectTypes.serviceObjectType module

Module contents

Submodules

nautilus.api.helpers module

Module contents

5.1.3 nautilus.auth package

Subpackages

nautilus.auth.primitives package

Submodules

nautilus.auth.primitives.passwordHash module

nautilus.auth.primitives.user module

Module contents

Submodules

nautilus.auth.backend module

nautilus.auth.decorators module

Module contents

5.1.4 nautilus.conventions package

Submodules

nautilus.conventions.actions module

nautilus.conventions.models module

nautilus.conventions.schema module

nautilus.conventions.services module

Module contents

5.1.5 nautilus.models package

Subpackages

nautilus.models.mixins package

Submodules

nautilus.models.mixins.crudNotificationCreator module

nautilus.models.mixins.hasID module

nautilus.models.mixins.hasPassword module

nautilus.models.mixins.user module

Module contents

nautilus.models.types package

Submodules

nautilus.models.types.password module

nautilus.models.types.s3File module

Module contents

Submodules

nautilus.models.base module

nautilus.models.util module

Module contents

5.1.6 nautilus.network package

Subpackages

nautilus.network.actionHandlers package

Submodules

nautilus.network.actionHandlers.createHandler module

nautilus.network.actionHandlers.crudHandler module

nautilus.network.actionHandlers.deleteHandler module

nautilus.network.actionHandlers.updateHandler module

Module contents

nautilus.network.consumers package

Submodules

nautilus.network.consumers.ActionConsumer module

nautilus.network.consumers.BasicConsumer module

Module contents

nautilus.network.registry package

Module contents

nautilus.network.tests package

Submodules

nautilus.network.tests.test_util module

Module contents

Submodules

nautilus.network.dispatch module

nautilus.network.util module

Module contents

5.1.7 nautilus.services package

Submodules

nautilus.services.connectionService module

nautilus.services.modelService module

nautilus.services.service module

nautilus.services.serviceManager module

Module contents

5.2 Module contents

CHAPTER 6

Indices and tables

- genindex
- modindex
- search