

Admin Site

msaSDK - Admin Site

*Admin Site is an efficient and easily extensible MSAApp (FastAPI) admin framework.
Inspired by Django-admin, and for a fast track through PoC's, MVP's to Production.*

`msaSDK - Admin Site` is an efficient `Python 3.7+` based framework on top of `msaSDK` & `fastapi` & `amis`, and build with standard Python type hints. The original intention of the development is to improve the application ecology and to quickly generate a visual dashboard for API's and web application. It was build to support very quick PoC's, MVP's in AI Applications. It can be extended with Auth and Login for an MVP Phase with the Auth Module by just define a setting to True.

Features

- **Build connected distributed applications faster:** Ready for Distributed Application Runtime (Dapr).
- **Consistency:** With sometimes 10s or 100s of Micro Services, the SDK helps to easy version control and provides a stable Dapr Basis.
- **High performance:** Based on `FastAPI`. Enjoy all the benefits.
- **High efficiency:** Perfect code type hints. Higher code reusability.
- **Support asynchronous and synchronous hybrid writing:** `ORM` is based on `SQLModel` & `Sqlalchemy`. Freely customize database type. Asynchronous mode. Strong scalability.
- **Front-end separation:** The front-end is rendered by `Amis`, the back-end interface is automatically generated by `msaSDK - Admin`. The interface is reusable.
- **Strong scalability:** The background page supports `Amis` pages and ordinary `html` pages. Easily customize the interface freely.
- **Automatic api documentation:** Automatically generate Interface documentation by `FastAPI`. Easily debug and share interfaces.

Dependencies

- `FastAPI`
- `SQLModel` combined with `SQLAlchemy` and `Pydantic`, with all their features .
- `Amis`: Vue.js / React - JSON based Frontend

Composition

msaSDK - Admin consists of three core modules, of which, amis, crud, which can be used as separate modules, and MSAApp as the core SDK part.

- amis: Based on the pydantic data model building library of baidu amis. To generate/parse data rapidly.
- crud: Based on FastAPI & SQLAlchemy. To quickly build Create, Read, Update, Delete common API interfaces.
- admin: Inspired by Django-Admin. Combine amis with crud. To quickly build Web Admin dashboard.

Simple Example

```
# -*- encoding: utf-8 -*-
"""
Copyright (c) 2022 - U2D.ai / S.Welcker
"""
__version__ = "0.0.1"

from typing import Optional
from sqlalchemy import SQLAlchemy
from msaSDK.admin.utils.fields import Field
from msaSDK.models.service import get_msa_app_settings
from msaSDK.service import MSAApp
from msaSDK.utils.scheduler import MSATimers, MSATimerEnum

# example async function to be executed by a timer
async def test_timer_min():
    app.logger.info("msaSDK Test Timer Async Every Minute")

# example sync/blocking function to be executed by a timer
def test_timer_five_sec():
    app.logger.info("msaSDK Test Timer Sync 5 Second")

# SQLAlchemy class to be used for auto API CRUD and/or Admin Site Web UI
class TestArticle(SQLAlchemy, table=True):
    id: Optional[int] = Field(default=None, primary_key=True, nullable=False)
    title: str = Field(title='ArticleTitle', max_length=200)
    description: Optional[str] = Field(default='', title='ArticleDescription',
max_length=400)
    status: bool = Field(None, title='status')
    content: str = Field(title='ArticleContent')

# SQLAlchemy class to be used for auto API CRUD and/or Admin Site Web UI
class TestCategory(SQLAlchemy, table=True):
    id: Optional[int] = Field(default=None, primary_key=True, nullable=False)
    title: str = Field(title='ArticleTitle', max_length=200)
```

```

    description: Optional[str] = Field(default='', title='ArticleDescription',
max_length=400)
    status: bool = Field(None, title='status')
    content: str = Field(title='ArticleContent')

# get the MSA app setting, clear the cache, set some settings
get_msa_app_settings.cache_clear()
settings = get_msa_app_settings()
settings.title = "SPK.ai - MSA/SDK MVP"
settings.version = "SPK.0.0.1"
settings.debug = True

# Create some timers with a MSATimer instance, define the interval and set the handler
my_timers: MSATimers = MSATimers()
my_timers.create_timer(MSATimerEnum.every_minute, test_timer_min)
my_timers.create_timer(MSATimerEnum.on_the_5_second, test_timer_five_sec)

# Create the main app instance, like FastAPI but provide a Setting Definition Instance
# Optional the Classes of your SQLAlchemy
# Define if the optional Admin Site gets mounted automatically, if False you need to
Mount in your own Startup MSAUIEvent Handler
# Optional the MSATimers instance
app = MSAApp(settings=settings, timers=my_timers, auto_mount_site=True,
            sql_models=[TestArticle, TestCategory],
            contact={"name": "msaSDK", "url": "http://u2d.ai", "email":
"stefan@u2d.ai"},
            license_info={"name": "MIT", "url": "https://opensource.org/licenses/MIT",
})

# use the internal logger of app
app.logger.info("Initialized " + settings.title + " " + settings.version)

# Optional use startup event
@app.on_event("startup")
async def startup():
    app.logger.info("msaSDK Own Startup MSAUIEvent")

# Optional use shutdown event
@app.on_event("shutdown")
async def shutdown():
    app.logger.info("msaSDK Own Shutdown MSAUIEvent")

```

Interface Preview

Home Screen with System Info

- Open `http://127.0.0.1:8090/admin/` in your browser:

The screenshot shows the MSA SDK Admin interface. On the left is a blue sidebar with navigation links: Home, TestArticle, TestCategory, API Docs, and Profiler. The main content area displays three tables:

Service Info					
Title	SPK.ai - MSA/SDK MVP		Version	SPK.0.0.1	Language
			Debug	True	

MSA SDK							
System	Linux	Python	3.9.14	Version	0.0.1	License	MIT

System							
Booted	2022-09-07 09:30:50	IP	127.0.1.1	MAC Address	45:af:c9:4d:95:67	Host Name	swelcker-XPS-15-7590
CPU Physical	8	CPU Logical	16	CPU in Use	5	CPU Affinity	16
Mem Total	65.6 GB	Mem Usage	23.5%	Mem Free	50.22 GB	PID	183310
OS Name	Linux	OS Version	#51-Ubuntu SMP Thu Aug 11 07:51:15 UTC 2022	OS Release	5.15.0-47-generic	Status	running

CRUD of SQLModels Screen

The screenshot shows the MSA SDK Admin interface for the 'TestCategory' screen. The sidebar is the same as the previous screenshot. The main content area includes a filter section with input fields for 'id', 'ArticleTitle', 'ArticleDescription', 'status' (a toggle switch), and 'ArticleContent'. Below the filter are 'Clear', 'Reset', and 'Search' buttons. A table displays two results, each with a checkbox, 'id', 'ArticleTitle', 'ArticleDescription', 'status' (toggle), and 'ArticleContent'. A modal dialog titled 'Click to control columns visibility' is open, showing checkboxes for 'id', 'ArticleTitle', 'ArticleDescription', 'status', and 'ArticleContent'. The modal also has 'Display columns', 'Sort', 'Cancel', and 'Confirm' buttons.

OpenAPI Interactive Documentation (Swagger) Screen

- Open `http://127.0.0.1:8090/#/admin/docs` in your browser:

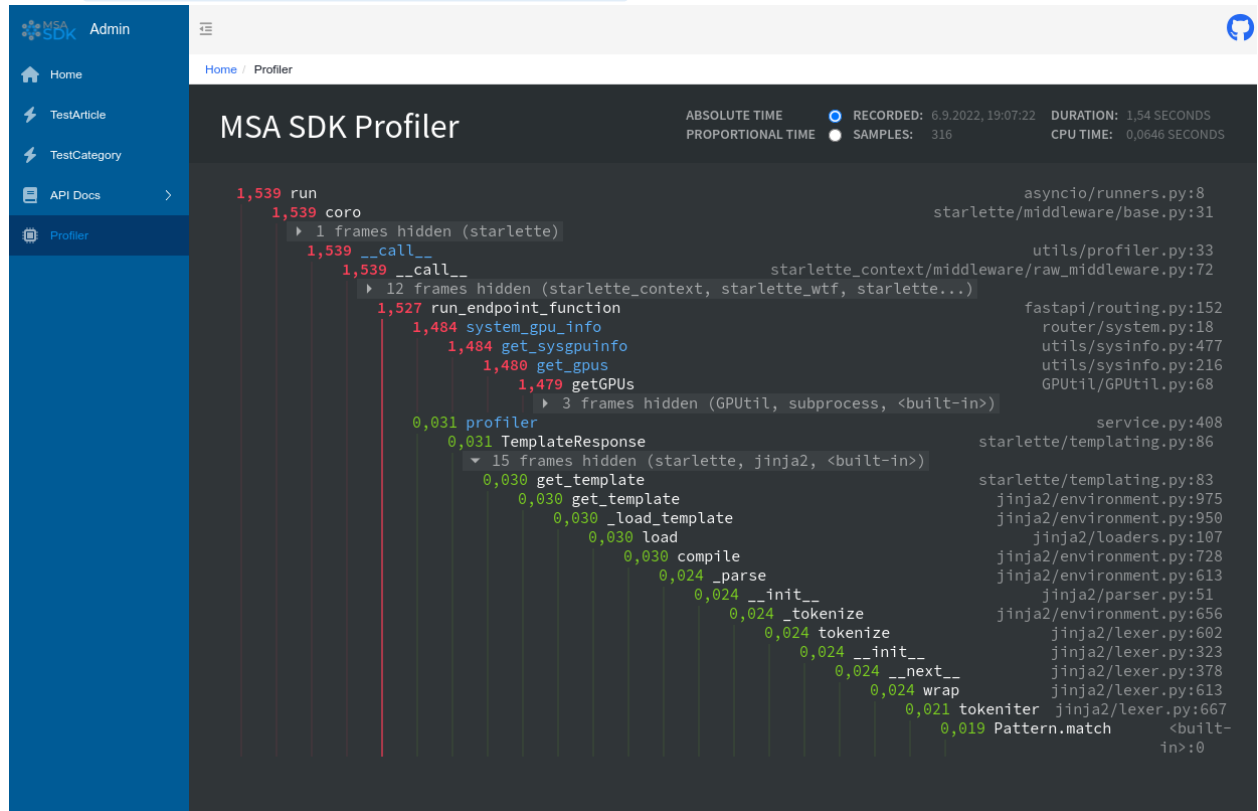
The screenshot displays the msaSDK Admin interface. On the left is a dark blue sidebar with navigation links: Home, TestArticle, TestCategory, API Docs (expanded), OpenAPI, Redocs, and Profiler. The main content area shows the OpenAPI documentation for the 'testcategory' endpoint. It lists several endpoints with their methods and descriptions:

- testcategory**
 - POST `/testcategory/list` List
 - GET `/testcategory/item/{item_id}` Read
 - PUT `/testcategory/item/{item_id}` Update
 - DELETE `/testcategory/item/{item_id}` Delete
 - POST `/testcategory/item` Create
- service**
 - GET `/healthcheck` Get Healthcheck
 - GET `/scheduler` Get Scheduler Status
 - GET `/status` Get Services Status
 - GET `/definition` Get Services Definition
 - GET `/settings` Get Services Settings
 - GET `/metrics` Metrics
- system**
 - GET `/sysinfo` System Info
 - GET `/sysgpuinfo` System Gpu Info
 - GET `/syserror` System Test Error
- openapi**
 - GET `/schema` Get Services Openapi Schema
 - GET `/info` Get Services Openapi Info
- pages**
 - GET `/profiler` Profiler

At the bottom of the page, there is a copyright notice: Copyright © 2022 by u2d.ai msa-sdk-admin v0.0.1

Profiler Screen

- Open <http://127.0.0.1:8090/#/admin/profiler> in your browser:



License Agreement

- **msaSDK** Based on **MIT** open source and free to use, it is free for commercial use, but please clearly show the copyright information about msaSDK - Auth Admin in the display interface.

Last update: September 13, 2022

Created: September 10, 2022