ACIT 3855 – Service Oriented Architectures – Winter 2024

Lesson 5 Pre-Reading

In Lesson 5 we will be building our third service, a Processing Service, and updating our Storage Service with additional RESTful API endpoints. So for both of these services we will be either creating or updating OpenAPI specifications.

Please review the following OpenAPI 3.0 tutorial:

https://app.swaggerhub.com/help/tutorials/openapi-3-tutorial

We will be adding some GET endpoints with parameters to our APIs, so make sure you are familiar with the syntax for adding those to our OpenAPI specifications.

For Lab 5, you will be creating a Processing Service that incorporates all the topics we have covered so far:

- OpenAPI 3.0 Specification
- Python Application using the Python connexion framework
- Yaml based external configuration
- Logging using the Python logging module
- MySQL Database using SQLAlchemy ORM
- Synchronous Communication using the Python requests module

In addition, the Processing Service will include periodic processing using the Python apscheduler module (https://apscheduler.readthedocs.io/en/latest/userguide.html). So you should make sure you understand what this code will do within our connexion application:

```python
def populate_stats():
    """ Periodically update stats """
    print(“This is my periodic processing”)


def init_scheduler():
    sched = BackgroundScheduler(daemon=True)
    sched.add_job(populate_stats,
                  'interval',
                  seconds=app_config['scheduler']['period_sec'])
    sched.start()


if __name__ == "__main__":
    # run our standalone gevent server
    init_scheduler()
    app.run(port=8100)
```