在Python的Framework中实现定时任务，我们有许多中选择。

我们可以使用在application外的工具，例如使用服务器的cron服务，来定时的运行脚本。也可以在application内直接通过手写代码来实现定时效果（一个持续的循环）。

同时，根据工作的复杂行，我们可以选定轻量化的组件APScheduler组件，同时如果是大型系统我们也可以使用celery + rabbitmq的机制实现。

APScheduler(Advanced Python Scheduler) 轻量化的python定时任务组件：

APScheduler的四大组件：

Trigger: 触发器，每个任务都有自己的触发器，用于决定这个任务需要在什么时间点运行。在初始化配置以后，触发器是无状态的。有3中trigger, date(指定时间运行一次)， interval(间隔多少时间执行), cron(一天中的某几个时间点执行)

任务储存器（job stores）：任务储存器由于储存被调度的任务。有许多中不同的任务储存器，默认的存储在内存中，也可以选用其他的任务储存器将任务存在数据库中。默认为[MemoryJobStore](https://apscheduler.readthedocs.io/en/latest/modules/jobstores/memory.html#apscheduler.jobstores.memory.MemoryJobStore)。

执行体：用于执行被调度的任务。默认为[ThreadPoolExecutor](https://apscheduler.readthedocs.io/en/latest/modules/executors/pool.html#apscheduler.executors.pool.ThreadPoolExecutor) 。

调度器（Scheduler）: 在一个app中通常只有一个调度器。并且我们通过scheduler去配置其他组件，而不是直接于其他组件交互。默认为BackgroundScheduler.