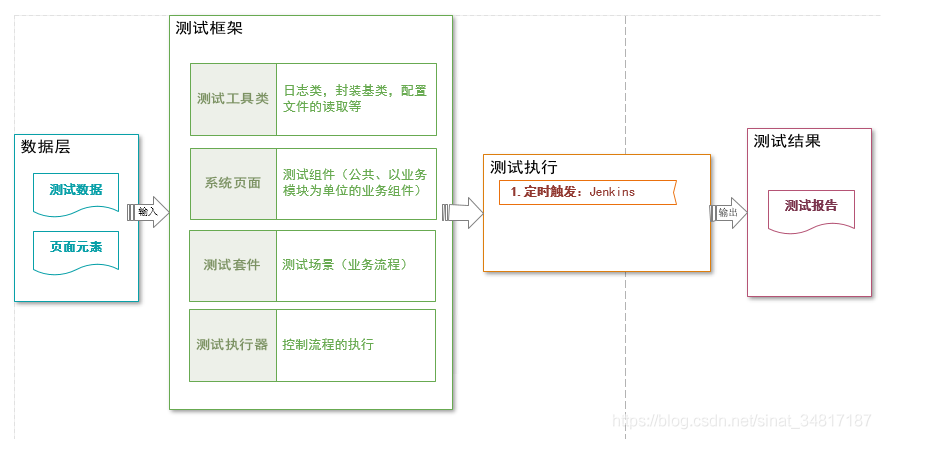
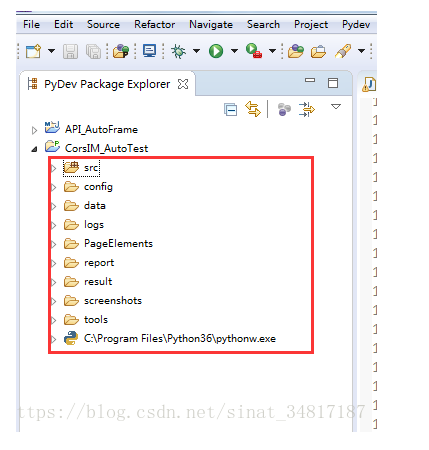
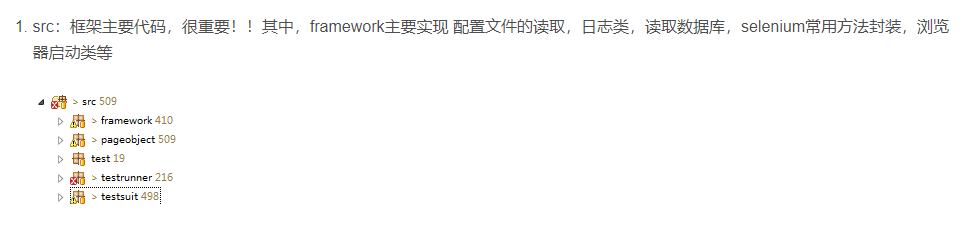
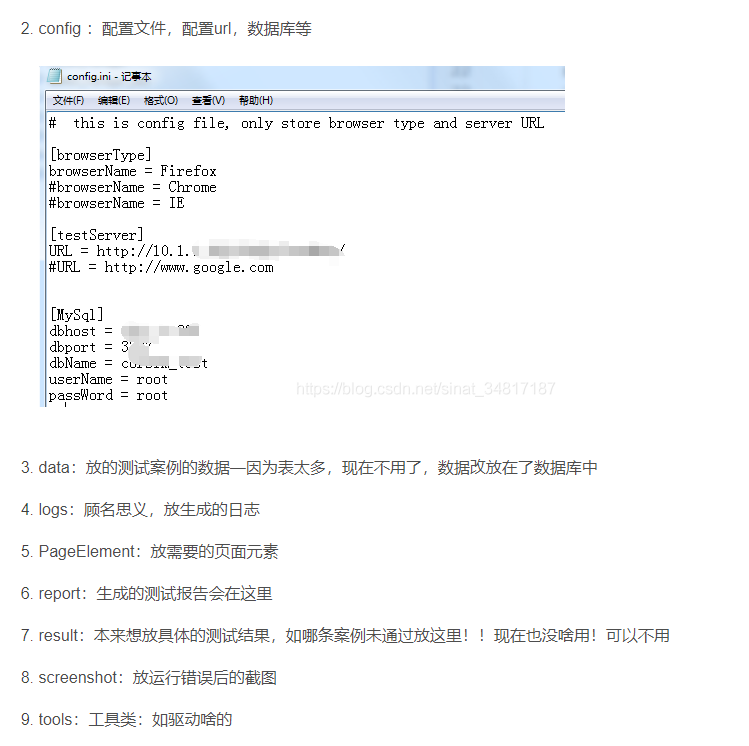
# 6/4/21, 8/31/21

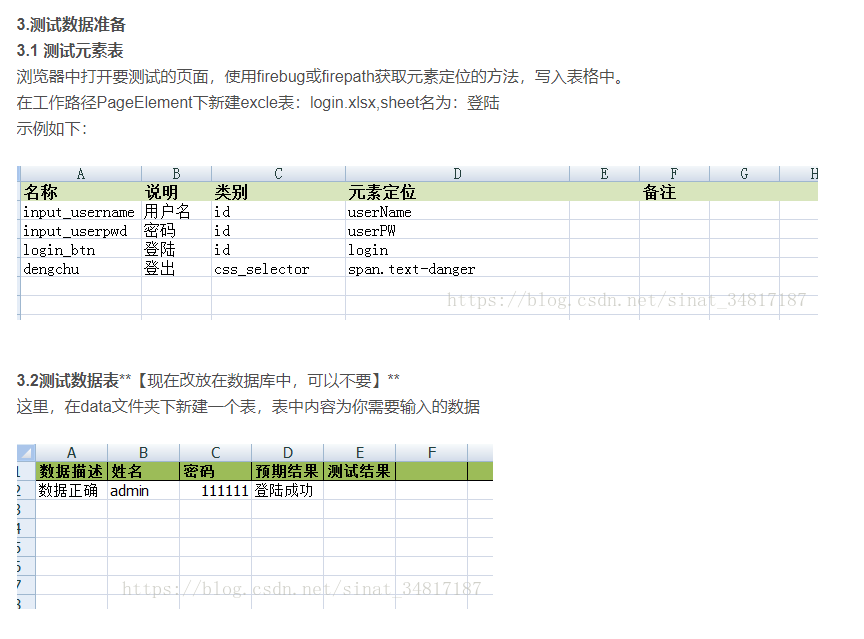
<https://github.com/chenjinwen-1994/framework_demo>





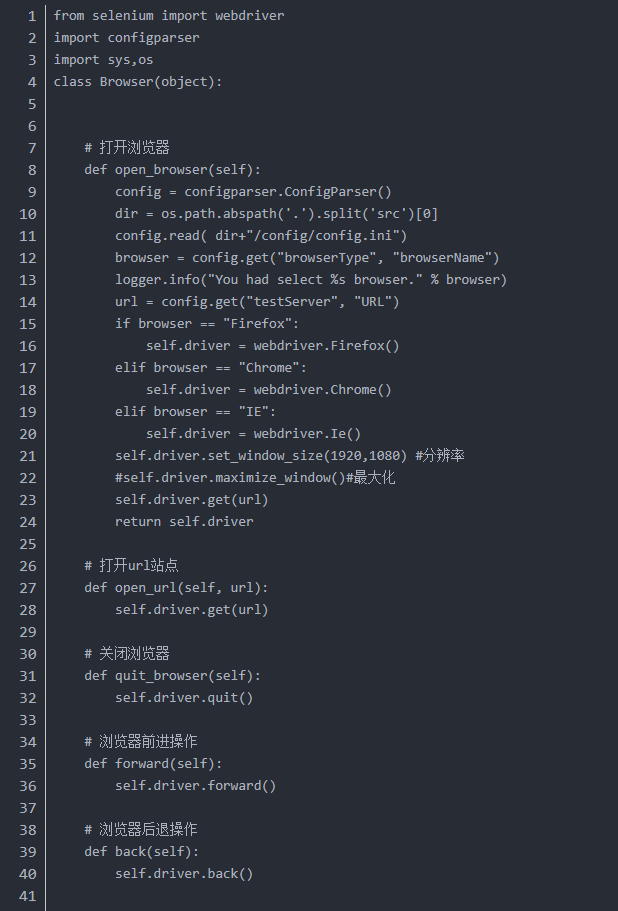


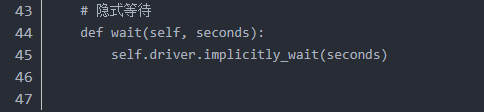




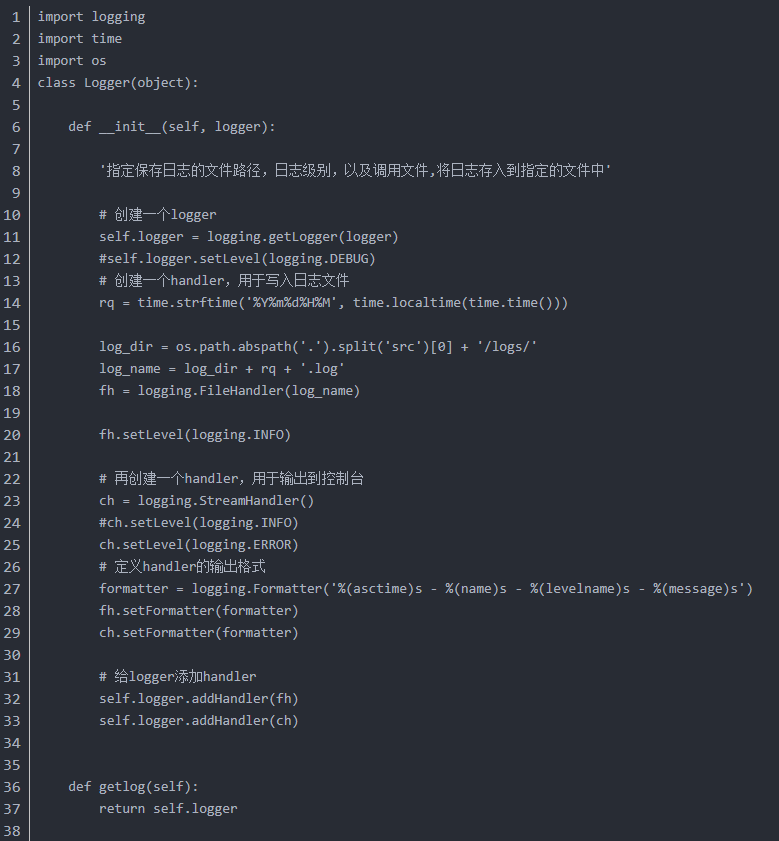








**4.3 框架之-----日志配置 logger类**



import logging  
import time  
import os  
class Logger(object):  
  
 def \_\_init\_\_(self, logger):  
 *'指定保存日志的文件路径，日志级别，以及调用文件,将日志存入到指定的文件中'* # 创建一个logger  
 self.logger = logging.getLogger(logger)  
 # self.logger.setLevel(logging.DEBUG)   
 # 创建一个handler，用于写入日志文件  
 rq = time.strftime('%Y%m%d%H%M', time.localtime(time.time()))  
  
 log\_dir = os.path.abspath('.').split('src')[0] + '/logs/'  
 log\_name = log\_dir + rq + '.log'  
 fh = logging.FileHandler(log\_name)  
  
 fh.setLevel(logging.INFO)  
  
 # 再创建一个handler，用于输出到控制台  
 ch = logging.StreamHandler()  
 # ch.setLevel(logging.INFO)   
 ch.setLevel(logging.ERROR)  
 # 定义handler的输出格式  
 formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')  
 fh.setFormatter(formatter)  
 ch.setFormatter(formatter)  
  
 # 给logger添加handler  
 self.logger.addHandler(fh)  
 self.logger.addHandler(ch)  
  
 def getlog(self):  
 return self.logger



