Logging 模块 简介

极客时间

扫码试看/订阅

《Selenium自动化测试实战》视频课程

日志的作用

- 程序调试
- 了解程序运行是否正常
- 故障分析与问题定位
- 用户行为分析

日志的等级

- DEBUG 最详细的日志信息，典型应用场景是 问题诊断
- INFO 信息详细程度仅次于 DEBUG，通常只记录关键节点信息，用于确认一切都是按照我们预期的那样进行工作
- WARNING 当某些不期望的事情发生时记录的信息（如，磁盘可用空间较低），但是此时应用程序还是正常运行的
- ERROR 由于一个更严重的问题导致某些功能不能正常运行时记录的信息
- CRITICAL 当发生严重错误，导致应用程序不能继续运行时记录的信息

logging 模块的使用

- 第一种方式是使用 logging 提供的模块级别的函数
- 第二种方式是使用 Logging 日志系统的四大组件

logging 模块定义常用函数

- logging.debug(msg, *args, **kwargs) 创建一条严重级别为 DEBUG 的日志记录
- logging.info(msg, *args, **kwargs) 创建一条严重级别为 INFO 的日志记录
- logging.warning(msg, *args, **kwargs) 创建一条严重级别为 WARNING 的日志记录
- logging.error(msg, *args, **kwargs) 创建一条严重级别为 ERROR 的日志记录
- logging.critical(msg, *args, **kwargs) 创建一条严重级别为 CRITICAL 的日志记录
- logging.log(level, *args, **kwargs) 创建一条严重级别为 level 的日志记录
- logging.basicConfig(**kwargs) 对 root logger 进行一次性配置

logging 模块的四大组件

- loggers 提供应用程序代码直接使用的接口
- handlers 用于将日志记录发送到指定的目的位置
- filters 提供更细粒度的日志过滤功能，用于决定哪些日志记录将会被输出（其它的日志记录将会被忽略）
- formatters 用于控制日志信息的最终输出格式

简单的日志输出

- 实例演示

Logging 日志格式输出

logging.basicConfig() 函数说明

参数名称	描述
filename	指定日志输出目标文件的文件名，指定该设置项后日志信心就不会被输出到控制台了
filemode	指定日志文件的打开模式，默认为'a'。需要注意的是，该选项要在filename指定时才有效
format	指定日志格式字符串，即指定日志输出时所包含的字段信息以及它们的顺序。logging模块定义的格式字段下面会列出。
datefmt	指定日期/时间格式。需要注意的是，该选项要在format中包含时间字段%(asctime)s时才有效
level	指定日志器的日志级别
stream	指定日志输出目标stream，如sys.stdout、sys.stderr以及网络stream。需要说明的是，stream和filename不能同时提供，否则会引发 `ValueError` 异常
style	Python 3.2中新添加的配置项。指定format格式字符串的风格，可取值为'%'、'{'和'$'，默认为'%'
handlers	Python 3.3中新添加的配置项。该选项如果被指定，它应该是一个创建了多个Handler的可迭代对象，这些handler将会被添加到root logger。需要说明的是：filename、stream和handlers这三个配置项只能有一个存在，不能同时出现2个或3个，否则会引发ValueError异常。

logging 模块的格式字符串

字段/属性名称	使用格式	描述
asctime	%(asctime)s	日志事件发生的时间--人类可读时间，如：2003-07-08 16:49:45,896
created	%(created)f	日志事件发生的时间--时间戳，就是当时调用time.time()函数返回的值
relativeCreated	%(relativeCreated)d	日志事件发生的时间相对于logging模块加载时间的相对毫秒数（目前还不知道干嘛用的）
msecs	%(msecs)d	日志事件发生事件的毫秒部分
levelname	%(levelname)s	该日志记录的文字形式的日志级别（'DEBUG', 'INFO', 'WARNING', 'ERROR', 'CRITICAL'）
levelno	%(levelno)s	该日志记录的数字形式的日志级别（10, 20, 30, 40, 50）
name	%(name)s	所使用的日志器名称，默认是'root'，因为默认使用的是 rootLogger
message	%(message)s	日志记录的文本内容，通过 `msg % args` 计算得到的
pathname	%(pathname)s	调用日志记录函数的源码文件的全路径
filename	%(filename)s	pathname的文件名部分，包含文件后缀
module	%(module)s	filename的名称部分，不包含后缀
lineno	%(lineno)d	调用日志记录函数的源代码所在的行号
funcName	%(funcName)s	调用日志记录函数的函数名
process	%(process)d	进程ID
processName	%(processName)s	进程名称，Python 3.1新增
thread	%(thread)d	线程ID
threadName	%(thread)s	线程名称

logging 模块实例演示

Logging 模块四大组件

四大组件

组件名称	对应类名	功能描述
日志器	Logger	提供了应用程序可一直使用的接口
处理器	Handler	将logger创建的日志记录发送到合适的目的输出
过滤器	Filter	提供了更细粒度的控制工具来决定输出哪条日志记录，丢弃哪条日志记录
格式器	Formatter	决定日志记录的最终输出格式

Logger 类相关方法

方法	描述
Logger.setLevel()	设置日志器将会处理的日志消息的最低严重级别
Logger.addHandler() 和 Logger.removeHandler()	为该logger对象添加 和 移除一个handler对象
Logger.addFilter() 和 Logger.removeFilter()	为该logger对象添加 和 移除一个filter对象
Logger.debug(), Logger.info(), Logger.warning(), Logger.error(), Logger.critical()	创建一个与它们的方法名对应等级的日志记录
Logger.exception()	创建一个类似于Logger.error()的日志消息
Logger.log()	需要获取一个明确的日志level参数来创建一个日志记录

Handler 类

Handler对象的作用是（基于日志消息的level）将消息分发到handler指定的位置（文件、网络、邮件等）

方法	描述
Handler.setLevel()	设置handler将会处理的日志消息的最低严重级别
Handler.setFormatter()	为handler设置一个格式器对象
Handler.addFilter() 和 Handler.removeFilter()	为handler添加 和 删除一个过滤器对象

Handler 相关子类

Handler	描述
logging.StreamHandler	将日志消息发送到输出到Stream，如std.out, std.err或任何file-like对象。
logging.FileHandler	将日志消息发送到磁盘文件，默认情况下文件大小会无限增长
logging.handlers.RotatingFileHandler	将日志消息发送到磁盘文件，并支持日志文件按大小切割
logging.hanlders.TimedRotatingFileHandler	将日志消息发送到磁盘文件，并支持日志文件按时间切割
logging.handlers.HTTPHandler	将日志消息以GET或POST的方式发送给一个HTTP服务器
logging.handlers.SMTPHandler	将日志消息发送给一个指定的email地址
logging.NullHandler	该Handler实例会忽略error messages，通常被想使用logging的library开发者使用来避免'No handlers could be found for logger XXX'信息的出现。

实例演示

为项目添加日志

极客时间

扫码试看/订阅

《Selenium自动化测试实战》视频课程