PyConChina 2019

Build a DSL editor using Python

崔衡

目录

1 背景知识

一些知识介绍

背景知识

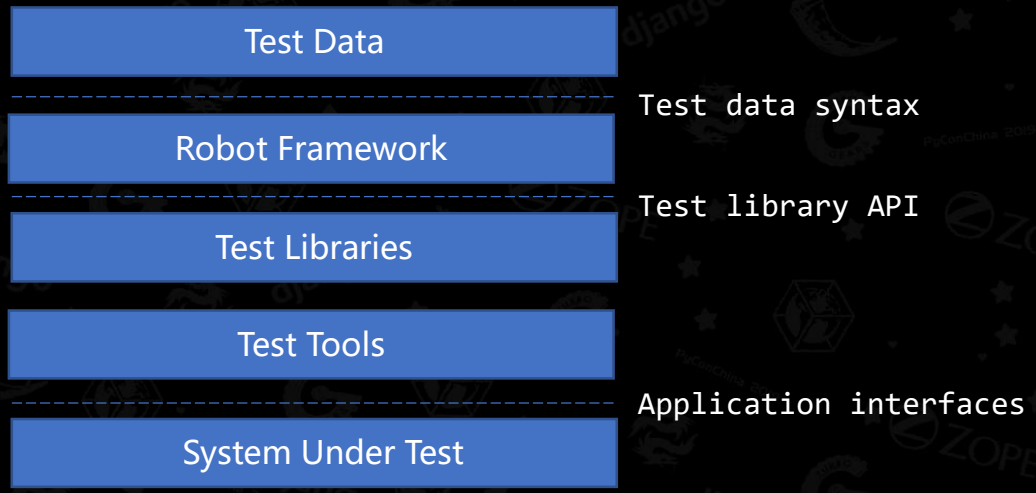
- DSL : Domain Specific Language
- Robot Framework : 基于Python编写的自动化测试框架
- Qt : 跨平台(桌面、服务器、移动端)的C++框架, 比如Linux的KDE桌面
- PyQt : Qt的Python绑定,如Python的开源IDE - Eric
- Qscintilla : C++编辑器Scintilla的Qt实现

2 Robot Framework

通过例子快速了解RF

Robot Framework

- RF是一个Data Driven测试框架, 用于支持自动化测试
- 支持Web、Console、移动端多平台的测试
- Clear、Easy、Modular



Robot Framework Architecture

Robot Framework


```python
# -*- coding: utf-8 -*-
from typing import Tuple
import paramiko


class SSHClient:
    def __init__(self, host: str, port: int, user: str, password: str):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.ssh = paramiko.SSHClient()

    def connect(self) -> Tuple[bool, str]:
        pass

    def close(self):
        pass

    def run_command(self, cmd: str) -> Tuple[bool, str, str]:
        pass
```

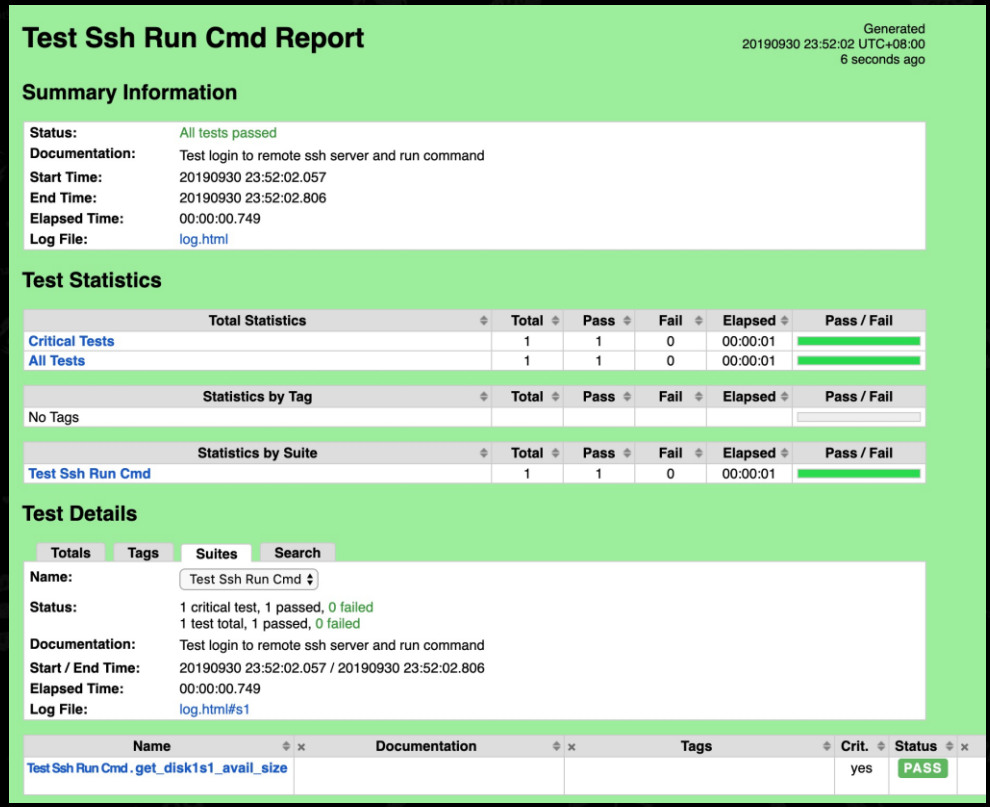
Robot Framework

- pip install robotframework

```
*** Settings ***
Documentation  Test login to remote ssh server and run command
Library  network.ssh_client.SSHClient  192.168.20.12  22  admin  hello-world  WITH NAME  s1
Library  BuiltIn

*** Variables ***
${getSizeCmd}      df -h|grep /dev/disk1s1|awk '{print $4}'|sed 's/.$//'

*** Test Cases ***
get_disk1s1_avail_size
 login
  # get disk available size in GBytes
  ${success}  ${message}  ${size} =   s1.run_command      ${getSizeCmd}
  log   disc size is ${size}  info
  close

*** Keywords ***
login
  s1.connect

close
  s1.close
```

Robot Framework

robot --outputdir output --pythonpath . test_get_ssh.robot



Test Ssh Run Cmd Report

Generated
20190930 23:52:02 UTC+08:00
6 seconds ago

Summary Information

Status:	All tests passed
Documentation:	Test login to remote ssh server and run command
Start Time:	20190930 23:52:02.057
End Time:	20190930 23:52:02.806
Elapsed Time:	00:00:00.749
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:01	
All Tests	1	1	0	00:00:01	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Test Ssh Run Cmd	1	1	0	00:00:01	

Test Details

Totals | Tags | Suites | Search

Name:	Test Ssh Run Cmd
Status:	1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed
Documentation:	Test login to remote ssh server and run command
Start / End Time:	20190930 23:52:02.057 / 20190930 23:52:02.806
Elapsed Time:	00:00:00.749
Log File:	log.html#s1

Name	Documentation	Tags	Crit.	Status
Test Ssh Run Cmd . get_disk1s1_avail_size			yes	PASS

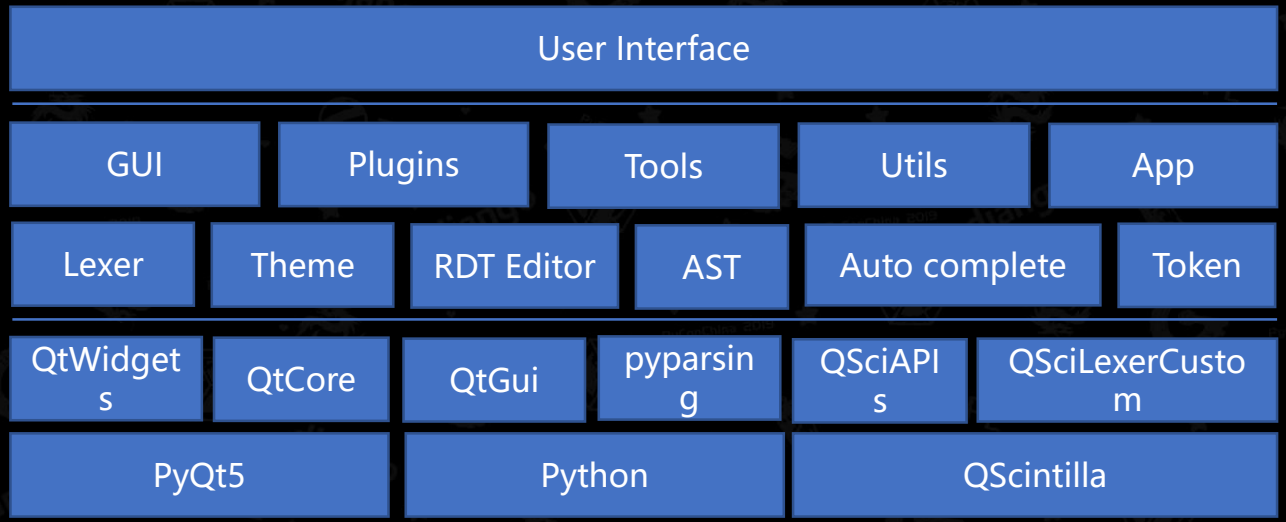
RF is great, but …

- 基于Tabular的编程, Tab vs Space
- 半角 vs 全角
- 资源文件导入
- Have Fun

3 如何实现

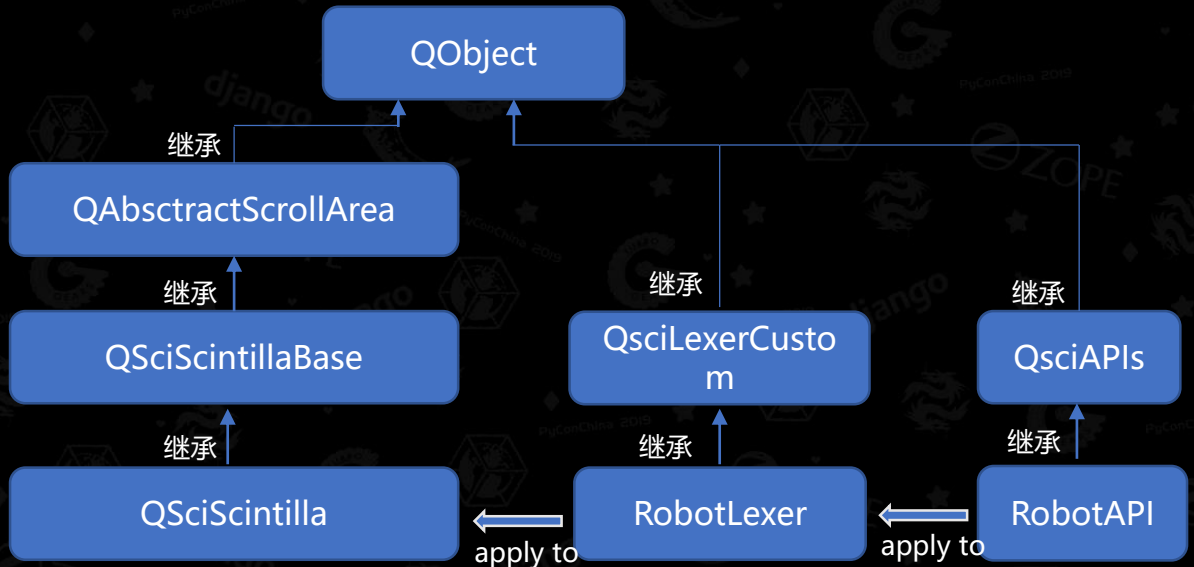
以**QScintilla**为例说明如何定制DSL语言的Editor, 实现高亮、补全、主题设置、代码折叠、异步渲染等特性

如何实现



Architecture

如何实现



如何实现

- 用pyparsing实现语义基本解析
- 定义自己的Token
- 对于AutoComplete, 由QsciAPIs实现
- 语法高亮, 借由QsciLexerCustom的如下方法
  - defaultColor
  - defaultFont
  - styleText(*self*, *start*, *end*)

如何实现

对于Editor的其他特性, 则借由对QsciScintilla的设置实现. 比如设置默认编码方式为Utf-8和基于Tab的补全可用如下方法

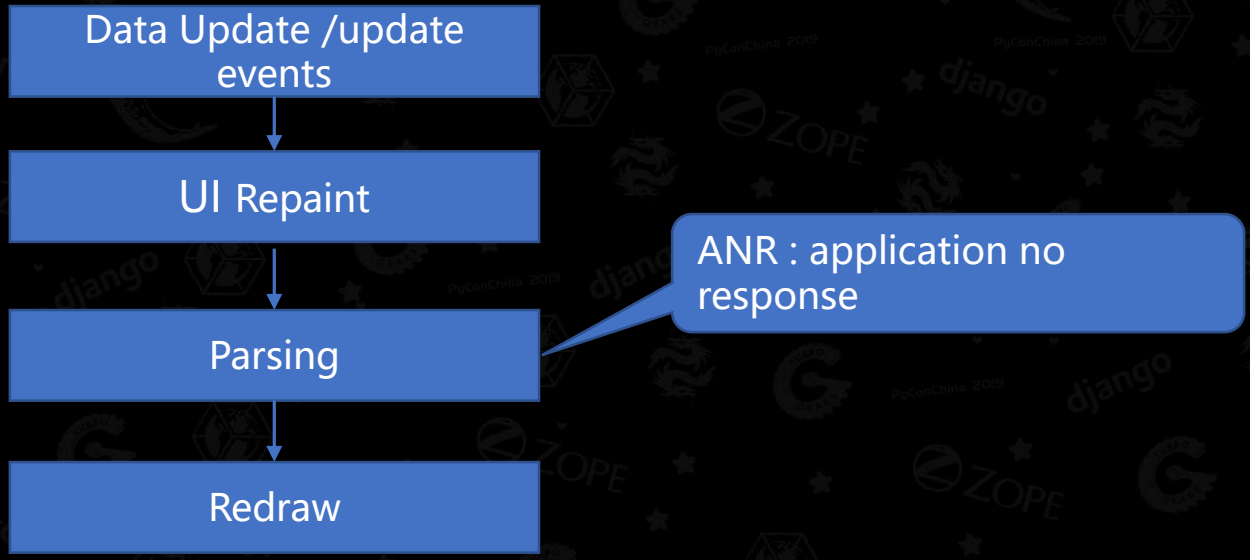
```
def init_by_default(self, tm: EditorTheme):
    self.setUtf8(True)
    # enable multi select as default
    self.SendScintilla(QsciScintilla.SCI_SETMULTIPLESELECTION, True)
    # edit multiple lines at the same time
    self.SendScintilla(QsciScintilla.SCI_SETADDITIONALSELECTIONTYPING, True)
    self.setMouseTracking(True)
    self.setIndicatorForegroundColor(QColor("white"))
```

如何实现

Scintilla内部维护一个指针来实现对语法的高亮和渲染, 只需要用户简单的实现setStyling接口来指定Token类型, 框架即可完成语法着色.

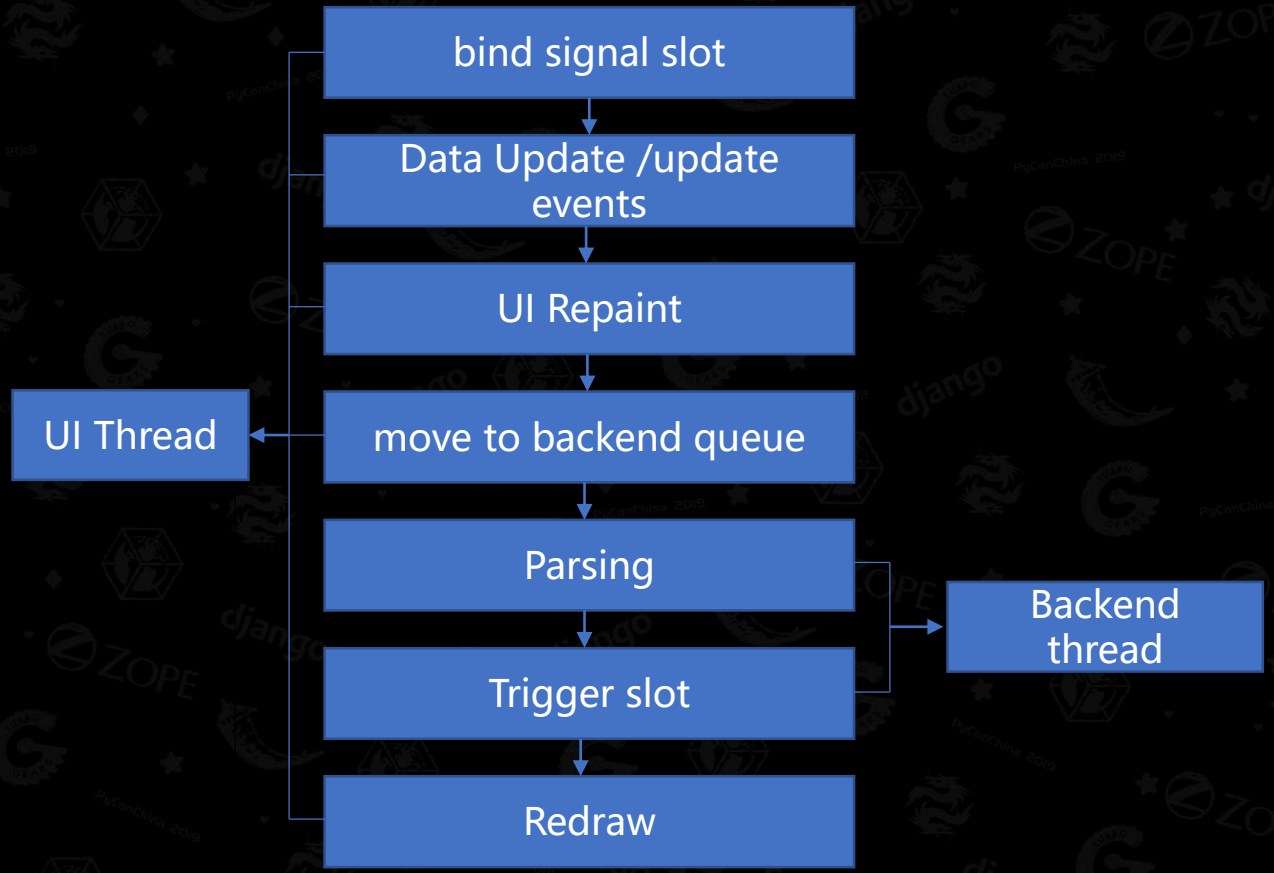
```
def real_style_text(self, txt: str, start: int, end: int):
    for (index, line) in enumerate(txt.splitlines()):
        if not line.strip():
            self.setStyling(len(line), RobotToken.T_DEFAULT)
            continue
        pass
```

如何实现


Data Update /update events
UI Repaint
Parsing
Redraw
ANR : application no response

都在主线程

如何实现

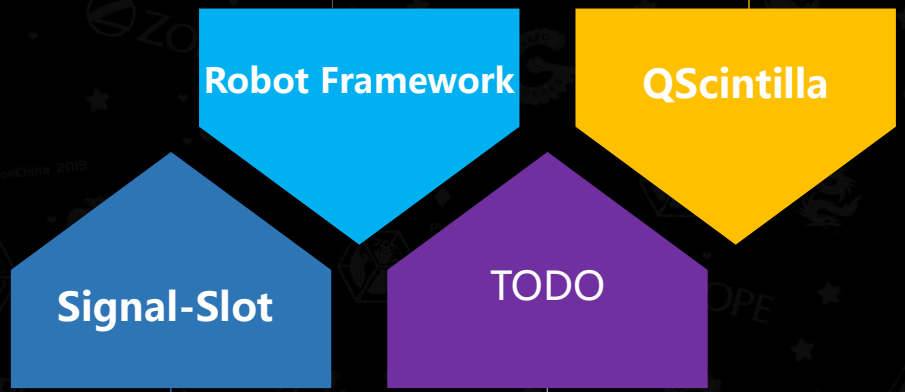


后台线程解析, UI线程渲染

4 总结

回顾总结

总结



5 Q&A

THANK YOU