

Please visit website: <http://cxyroad.com>

\* 插件使用效果:

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/397302aa120f42df81b7a799d8c3f104~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2918&h=1616&s=348890&e=png&b=282c34)

Hello World!

-----

我们先来看个最简单的例子，通过`->`、`-->`和`:`就可以在参与者之间传递消息，不用明确声明参与者。

...

@startuml

老张 -> 老王 : 老王，你好啊

老王 --> 老张 : 老张，你好啊

老张 -> 老王 : 最近有空一起喝茶

老张 <-- 老王 : OK

@enduml

...

![hello](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4b36ba21dd964403b0b0c0d1525dc510~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=450&h=557&s=33348&e=png&b=fefefe)

### 三、PlantUML时序图语法

=====

接下来，我们看看PlantUML时序图的一些具体语法。

声明参与者

-----

我们使用关键字participant 来声明参与者，就可以对该参与者进行更多的控制。声明的顺序就是默认的\*\*显示顺序\*\*。我们也可以用这些关键字来声明参与者，给参与者设置不同的形状。

- \* actor (角色)
- \* boundary (边界)
- \* control (控制)
- \* entity (实体)
- \* database (数据库)
- \* collections (集合)
- \* queue (队列)

我们还可以通过 `as` 关键字重命名参与者。

```
...  
@startuml  
participant Participant as Foo  
actor Actor as Foo1  
boundary Boundary as Foo2  
control Control as Foo3  
entity Entity as Foo4  
database Database as Foo5  
collections Collections as Foo6  
queue Queue as Foo7  
@enduml  
...
```

```
![参与者](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/e3701bbcaf314cb7b42017fb1dc807a1~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=834&h=269&s=30702&e=png&b=fefefe)
```

默认的颜色比较单调，也可以通过 `#` 来设置参与者的[颜色](http://cxyroad.com/ "https://plantuml.com/zh/color"):

```
...  
@startuml  
actor Bob #blue
```

' The only difference between actor  
'and participant is the drawing  
participant Alice #SkyBlue  
participant "I have a really\nlong name" as L #00ff00

Alice->Bob: Authentication Request  
Bob->Alice: Authentication Response  
Bob->L: Log transaction  
@enduml

...

![上颜色的参与者](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/34694f7bd4434358b61060a6ccb159e8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=682&h=587&s=41198&e=png&b=ffffff)

## 消息传递

-----

在不同参与者之间，通过箭头+:来表示消息传递。

\* 同步消息:

...

A -> B: 同步消息文本

...

![同步消息](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/d58f939afad1410b941dd828666692f8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=282&h=265&s=9633&e=png&b=fefefe)

\* 异步消息: 由发送者A指向接收者B，表示A发送后不需要等待B立即处理。

...

A ->> B: 异步消息文本

...

![异步消息](https://p3-juejin.byteimg.com/tos-cn-i-

k3u1fbpfcg/2fa3bff4f59c43048e82855210f0df01~tplv-k3u1fbpfcg-jj-mark:302 容，已经足够我们常见的时序图需求了。

#### 四、完整实例

=====

我们接下来看一个稍微完整一点例子，在这个例子中，我们的需求，是要在原本的登录的基础上，引入Google登录。

...

@startuml

skinparam ParticipantBackgroundColor #DeepSkyBlue

actor 用户 as c #DeepSkyBlue

participant "客户端" as client

participant "服务网关" as ga

participant "用户服务" as user

database "数据库" as DB #DeepSkyBlue

participant "Google服务" as google #LightCoral

activate c #DeepSkyBlue

activate client #DeepSkyBlue

c->client:用户登录

group#LightCoral #LightCoral Google登录客户端流程

client -> google : 请求Google OAuth登录

activate google #DeepSkyBlue

google-->client:登录url

client->google:跳转登录页

google -> google : 用户登录

google --> client : Google登录Token

deactivate google

end

|||

client -> ga : 登录请求

note right#LightCoral:新增登录方式，三方登录请求实体

activate ga #DeepSkyBlue

ga ->user:请求转发

activate user #DeepSkyBlue

```

alt#DeepSkyBlue 常规登录
  user -> DB : 查询用户信息
  activate DB #DeepSkyBlue
  DB -> user : 用户信息
  deactivate DB
  user->user:登录密码校验

|||
else Google登录
  group#LightCoral #LightCoral Google登录服务端流程
    user->google:验证token
    activate google #DeepSkyBlue
    google-->user:用户信息
    deactivate google
    user->user:存储或更新用户信息
  end group
end

user-->ga:登录结果
deactivate user
ga -> client : 响应
deactivate ga
alt#DeepSkyBlue 成功
  client -> c : 登录成功
else 失败
  client -> c : 登录失败
end
deactivate client
|||
@enduml

```

...

![登录时序图](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/545f5b0a25d947fcb5214200cb9426cb~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1021&h=1811&s=172112&e=png&b=fff)fff)

## 五、总结

====

PlantUML使用起来整体上还是非常舒服的，对我而言，这几个特点让我爱不释手：

- \* 提供了类似Markdown的所见即所得的使用体验，免去调整图形之扰
- \* PlantUML本质是文本，可以进行版本控制，多人协同
- \* 语雀支持文本画图的功能，可以嵌入PlantUML文本，支持在语雀文档里直接修改

PlantUML还有很强大的扩展性和灵活性，我们可以通过 [官方文档](<http://cxyroad.com/> "https://plantuml.com/zh/") 继续探索更多高级语法和技巧，来绘制更加复杂和丰富的图表。

----

**\*\*参考: \*\***

1. [开源工具，使用简单的文字描述画UML图](<http://cxyroad.com/> "https://plantuml.com/zh/")
2. [Choosing colors](<http://cxyroad.com/> "https://plantuml.com/zh/color")
3. [顺序图的语法和功能](<http://cxyroad.com/> "https://plantuml.com/zh/sequence-diagram")
4. 如何使用 PlantUML 画系统设计图? - 掘金
5. [顺序图的语法和功能](<http://cxyroad.com/> "https://plantuml.com/zh/sequence-diagram")
6. GPT老师

----

原文链接: <https://juejin.cn/post/7369470562143846410>