Revise（动态修改）使用说明

1 概述

Revise.jl允许您修改代码并使用更改，而无需重新启动Julia。

MWORKS.Syslab默认安装了Revise库。

2 使用方法

2.1 常规使用

不妨以`Example.jl`为例，介绍如何使用Revise来辅助Julia包的开发。

首先，在联网情况下，我们可以在Julia REPL包模式下，输入以下命令来安装`Example.jl`。

```
# 在联网模式下
(v1.7) pkg> dev Example
[...output related to installation...]
```

当然，在离线情况下，我们也可以通过Syslab的`开发库`面板来新建一个包。

```
# Example.jl源代码
module Example
export hello, domath

"""
    hello(who::String)

Return "Hello, `who`".
"""
hello(who::String) = "Hello, $who"

"""
    domath(x::Number)

Return `x + 5`.
"""
domath(x::Number) = x + 5

end
```

其次，加载 Revise：

```
julia> using Revise # importantly, this must come before `using Example`

julia> using Example

julia> hello("world")
"Hello, world"
```

目前，`Example`模块中还不存在一个名为`f`的函数：

```
julia> Example.f()
ERROR: UndefVarError: f not defined
```

此时，可以在编辑器中打开`Example.jl`源代码，添加一个函数`f() = π`并保存文件。

回到Julia REPL（不要重新启动 Julia）并尝试以下操作：

```
julia> Example.f() #无需重启Julia REPL
π = 3.1415926535897...
```

2.2 修改结构体

不妨以`Example.jl`为例，继续介绍如何基于Revise动态修改结构体。

首先，在`Example.jl`示例基础上，增加一段代码：

```
export FooStruct, processFoo

# 版本1
struct FooStruct1
    bar::Int
end
FooStruct = FooStruct1

# 函数不用变
function processFoo(foo::FooStruct)
    @info foo.bar
end
```

在REPL中执行以下语句：

```
julia> using Revise #若已加载则忽略此句
julia> using Exmaple
julia> processFoo(FooStruct(1))
[ Info: 1
```

其次，若想修改结构体，如新增一个成员：

```
# 版本2
struct FooStruct2  # change version here
    bar::Int
    str::String # new add
end
FooStruct = FooStruct2   # change version here
```

此时，不需要重启REPL，只需执行`revise(Module)`，即可生效，如下所示：

```
julia> revise(Example)
┌ Warning: Replacing docs for `Example.hello :: Tuple{String}` in module
`Example`
└ @ Base.Docs docs\Docs.jl:240
┌ Warning: Replacing docs for `Example.domath :: Tuple{Number}` in module
`Example`
└ @ Base.Docs docs\Docs.jl:240
true

julia> processFoo(FooStruct(1,"x"))
[ Info: 1
```

最后，当我们结构体设计稳定后，就可以使用正式名称，并重启REPL。

```
# 结构体稳定后，改为正式名称
struct FooStruct  # change version here
    bar::Int
    str::String
end
# FooStruct = FooStruct2    # 注释临时代码
```

3 注意事项

3.1 局限性

有些类型的更改是 Revise（或者通常是 Julia 本身）无法合并到正在运行的 Julia 会话中：

- 类型定义或常量的修改
- 共享相同名称的变量和函数之间的冲突
- 去除模块的导出项 `export`

4 参考文档

- 帮助文档：
  - https://timholy.github.io/Revise.jl/stable/
  - https://timholy.github.io/Revise.jl/stable/limitations/
- 工作流：https://juliacn.gitlab.io/JuliaZH.jl/manual/workflow-tips.html