

Список загруженных / импортированных пакетов в Юлии

julia

reproducible-research

Как я могу получить список импортированных / использованных пакетов сеанса Julia?

`Pkg.status()` список всех установленных пакетов. Меня интересуют те, которые были импортированы / загружены через `using ...` или `import ...`

Кажется, что `whos()` содержит соответствующую информацию (имена и является ли это модуль или нет). Может ли вывод `whos()` быть записан в переменной?

Автор: Julian

Источник

Размещён: 14.07.2019 12:04

Ответы (6)

```
using Lazy
children(m::Module) =
  @>> names(m, true) map(x->m.(x)) filter(x->isa(x, Module) && x ≠ m)
```

Решение

`children(Main)` затем даст вам список загруженных модулей.

Изменить: я использовал `Lazy.jl` здесь для макроса `moloch` (`@>>`), но вы можете переписать его без достаточно легко:

```
children(m::Module) =
  filter(x->isa(x, Module) && x ≠ m, map(x->m.(x), names(m, true)))
```

В качестве альтернативы вы можете добавить `&& x ≠ Lazy` к `filter` чтобы избежать его включения.

Автор: one-more-minute

Размещён: 31.08.2014 02:10

Используйте `names` , например

```
julia> using JuMP

julia> using Gurobi

julia> names(Main)
13-element Array{Symbol,1}:
 :Calculus
 :ans
 :JuMP
 :DualNumbers
 :Graphs
 :DataStructures
 :ReverseDiffSparse
 :MathProgSolverInterface
 :Base
 :MathProgBase
 :Core
 :Gurobi
 :Main
```

Автор: IainDunning

Размещён: 29.08.2014 08:59

Предложенные ответы не работают с Julia 1.0 и, следовательно, вот версия Julia 1.0:

```
filter((x) -> typeof(eval(x)) <: Module && x ≠ :Main, names(Main, imported=true))
```

Автор: Przemyslaw Szufel

Размещён: 28.10.2018 10:24

Ответ выше больше не работает, как раньше в Julia 0.5. Это работает во многих случаях, например:

```
children(SIUnits) -> SIUnits.ShortUnits
```

Но большинство пакетов (которые я использую) на самом деле не определяют подмодули. Я нахожу это полезным для отладки, в версии для командной строки Джулии и в отличной IDE Juno на одну минуту:

```
loadedmodules() = filter(names(Main, false)) do n
    isa(eval(n), Module) && n ≠ Main
end
```

Автор: hustf

Размещён: 26.12.2016 07:25

Я использую,

```
using Pkg
function magic()
    println("Julia " * string(VERSION))
    for (key, version) ∈ sort(collect(Pkg.installed()))
        try
            isa(eval(Symbol(key)), Module) && println(key * " " * string(version))
        end
    end
end
```

Автор: José Bayoán Santiago Calderón

Размещён: 04.04.2018 08:09

Так что это не так приятно с одной строкой, но: он работает **v1.0**, и он позволяет простой рекурсивный поиск, чтобы показать все загруженные модули:

```
function children(m::Module)
    ns = names(m, imported=true, all=true)
    ms = []
    for n in ns
        if n != nameof(m)
            try
                x = Core.eval(m, n)
                x isa Module && push!(ms, x)
            catch end
        end
    end
    ms
end

function children_deep(m::Module)
    cs = children(m)
    for c in cs
        cs = union(cs, children(c))
    end
    cs
end
```

Затем:

```
julia> children_deep(Main)
43-element Array{Any,1}:
Base
Core
InteractiveUtils
Base.BaseDocs
Base.Broadcast
Base.Cartesian
Base.Checked
Core.Compiler.CoreDocs
:
Base.Sys
Base.Threads
Base.Unicode
Base.__toplevel__
Core.Compiler
Core.IR
Core.Intrinsics
Main
```

Автор: NHDaly

Размещён: 30.04.2019 02:23