

# Konacha

<https://github.com/jfirebaugh/konacha>



# Kas yra Konacha?

Įrankis skirtas testuoti *Javascript* kodą

# Kodēl naudot Konach'a?

Pritaikytas būtent  
Rails platformai



# Pasirengimas darbui

## Gemfile

```
# ..
```

```
group :test, :development do  
  gem 'konacha'  
end
```

```
# ..
```

# Pirmi „spekai“

```
1  #= require array
2
3  describe "Array#sum", ->
4    it "should return 0 when the Array is empty", ->
5      [].sum().should.equal(0)
6
7    it "should return the sum of numeric elements", ->
8      [1,2,3].sum().should.equal(6)
```

# Paleidimas (shell)

```
_vilu@fedora:~/home/shared/workspace/konacha-test
_vilu@fedora konacha-test$ bundle exec rake konacha:run

      FF

Failed: should return 0 when the Array is empty
  [].sum is not a function
in :

Failed: should return the sum of numeric elements
  [1, 2, 3].sum is not a function
in :

Finished in 6.35 seconds
2 examples, 2 failures
_vilu@fedora konacha-test$
```

# Paleidimas (naršyklėje)

passes: 0 failures: 2 duration: 0.11s

100%

## Array#sum

✘ should return 0 when the Array is empty (view source)

```
TypeError: Object has no method 'sum'  
  at Context.<anonymous> (http://localhost:3500/...  
  at Test.run (http://localhost:3500/assets/moch...  
  at Runner.runTest (http://localhost:3500/asse...  
  at http://localhost:3500/assets/mocha.js:2978  
  at next (http://localhost:3500/assets/mocha.js...  
  at http://localhost:3500/assets/mocha.js:2870  
  at next (http://localhost:3500/assets/mocha.js...  
  at http://localhost:3500/assets/mocha.js:2838  
  at http://localhost:3500/assets/mocha.js:3619
```



```
_vilu@fedora: /home/shared/workspace/konacha-test  
_vilu@fedora konacha-test$ bundle exec rake konacha:serve  
your tests are here:  
http://localhost:3500/  
>> Thin web server (v1.3.1 codename Triple Espresso)  
>> Maximum connections set to 1024  
>> Listening on 0.0.0.0:3500, CTRL+C to stop
```

# Konacha suteikia:

- Transakcijas testams
- Į rspec panašią sintaksę
- Testų išskyrstymą į kelių lygių aprašymus
- Galybę matcher'ių testams



# Matcher'iai

```
[].empty().should.be.true
[].empty().should.be.ok
[1, 2, 3].empty().should.not.be.ok
[1, 2, 3].empty().should.false
[].should.be.empty
[1, 2].sum().should.equal(3)
[1, 2].sum().should.be.below(10)
[1, 2].sum().should.be.above(1)
({foo: 'bar'}).should.eql({ foo: 'bar' })
('Bar').should.be.a('string')
('Bar').should.be.an.instanceof(String)
someObject.should.have.property('size')
someObject.should.have.property('size', 100)
[1, 2].should.have.length(3)
'foobar'.should.match(/^foo/)
'foobar'.should.have.string('foo')
[1, 2, 3].should.include(3)
({ foo: 1, bar: 2 }).should.have.keys(['foo', 'bar']);
({ foo: 1, bar: 2, baz: 3 }).tshould.contain.keys('foo', 'bar')
someFunction.should.throw(ReferenceError).and.not.throw(/good function/)
cat.should.respondTo('jump')
someObject.to.satisfy (object) -> object.x > 15
(1.5).should.be.closeTo(1, 0.5)
```

# Taip pat

Įrankis pasikartojantiems duomenims testuose kurti (įkvėptas factory\_girl):

<https://github.com/bkeepers/rose>

Alternatyvos, kiti įrankiai:

- Jasmine
- Qunit
- JSLitmus („performanco“ testavimui)
- Zombie.js (vartotojo interfeisui, vietoj rspec-capybara)

?