

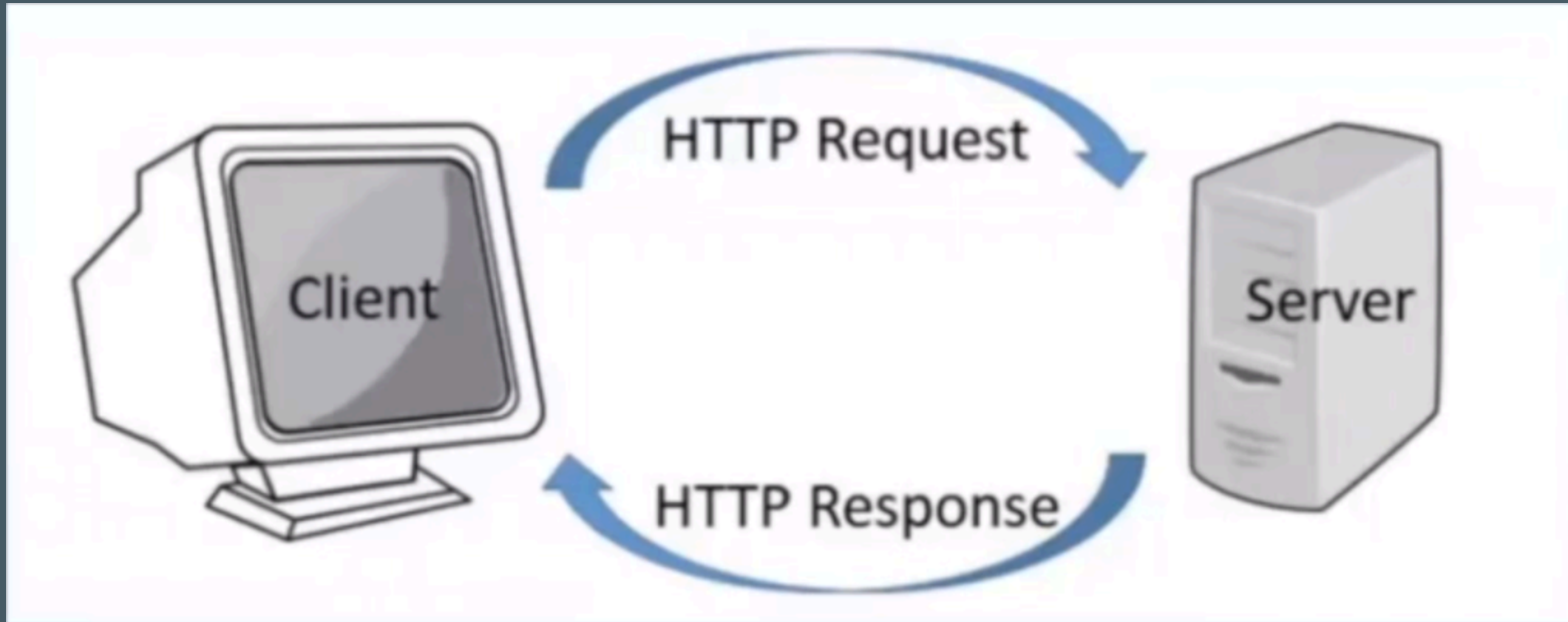
Rack

Хекслет: Rails-программист

Что обсудим

- HTTP
- Rack
- Middlewares

HTTP



HTTP Request

- Verb (GET, POST, PUT, etc)
- Path(<https://ru.hexlet.io>)
- Headers
- Parameters (/courses/42?step=2)

HTTP Response

- Status (1xx, 2xx, 3xx, 4xx, 5xx)
- Headers
- Body

Пример

```
curl --head https://ru.hexlet.io
```

```
curl --head https://ru.hexlet.io
HTTP/2 200
server: nginx/1.21.1
date: Wed, 01 Dec 2021 12:50:51 GMT
content-type: text/html; charset=utf-8
vary: Accept-Encoding
vary: Accept
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-download-options: noopen
x-permitted-cross-domain-policies: none
referrer-policy: strict-origin-when-cross-origin
# ...
```

Пример

```
curl -X GET https://ru.hexlet.io
```

```
<!DOCTYPE html>
<html lang='ru' prefix='og: https://ogp.me/ns#'>
<head>
<title>Hexlet – больше чем школа программирования. Онлайн курсы, сообщество программистов</title>
<meta name="csrf-param" content="authenticity_token" />
<meta name="csrf-token" content="" />
...
```

Rack

- Стандарт интерфейса веб-сервера
- Каркас для Middlewares
- Gem

Rack: Стандарт интерфейса веб-сервера

- `Rack::Handler::Puma`
- `Rack::Handler::Thin`
- `Rack::Handler::WEBrick`

Rack: Стандарт интерфейса веб-сервера

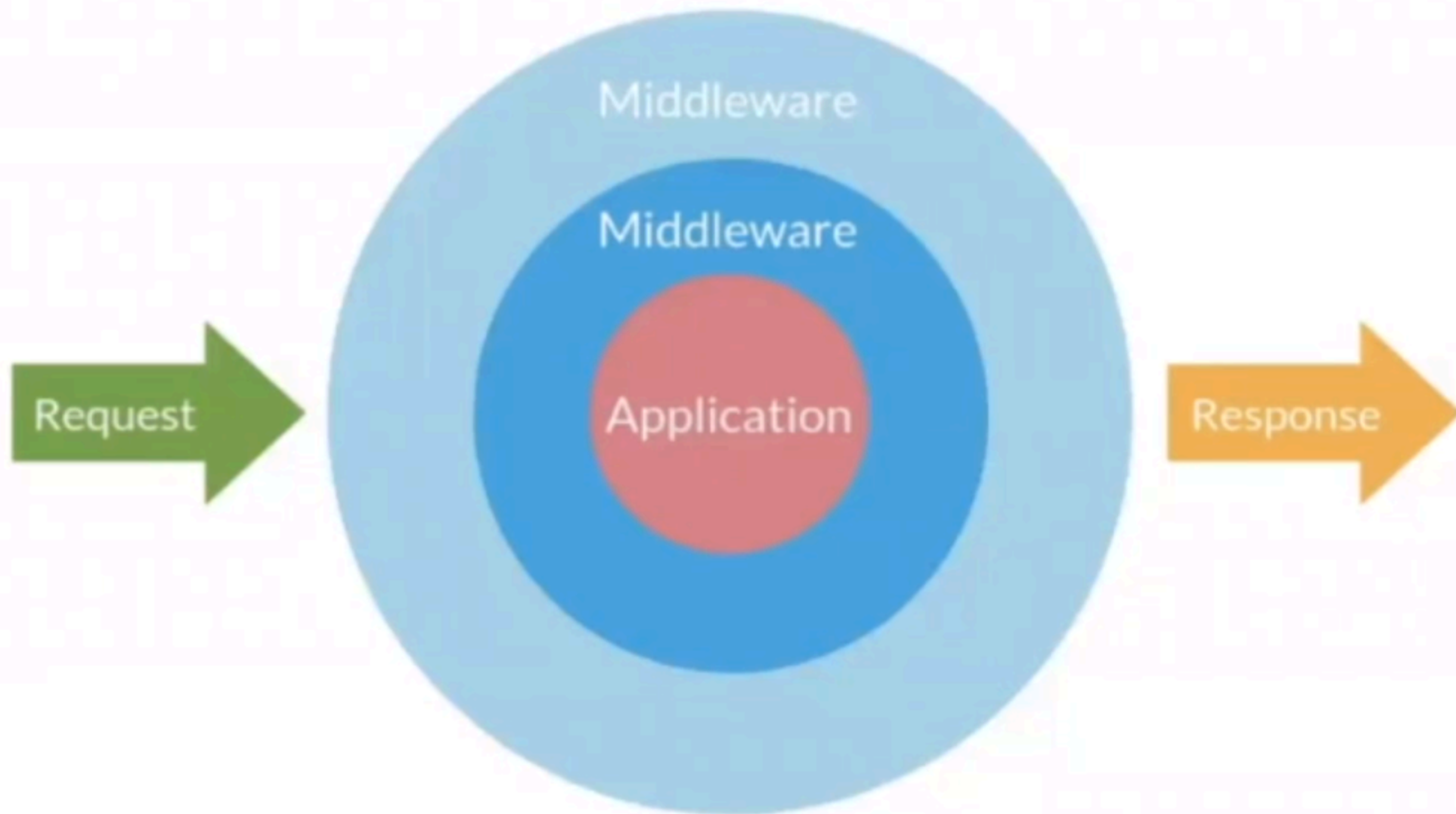
- `irb | ruby my_app.rb`

```
require 'rack'  
Rack::Handler::Thin.run lambda { |env| [200, {'Content-Type' => 'text/plain'}, ['Hello!']] }
```

- `config.ru`

```
rackup  
curl -X GET localhost:9292  
# => Hello!
```

Каск. каркас для Middleware



Middleware

```
class MyMiddleware
  def initialize(app)
    @app = app
  end

  def call(env)
    status, headers, body = @app.call(env)
  end
end

class App
  def call(env)
    [200, {}, ["success"]]
  end
end

use MyMiddleware
run App.new
```

Middleware

Назначение

- Авторизация и аутентификация
- Мониторинг
- Логирование
- Сериализация
- Роутинг
- Бизнес-логика

Rack: Gem

Rack Middlewares

- Rack::Files
- Rack::Events
- Rack::Head
- Rack::Lock
- Rack::Reloader
- Rack::Runtime
- Rack::ShowException

Rack: Gem

Rack Helpers

- Rack::Request
- Rack::Response
- Rack::MockRequest, Rack::MockResponse
- Rack::MediaType
- Rack::Mime
- Rack::RewindableInput