



OpenSSL::PKey::PKeyError at /_browserid_assert

rsa#set_key= is incompatible with OpenSSL 3.0

- **file:** browserid.rb
- **location:** set_key
- **line:** 51

BACKTRACE

[\(condense\)](#)

JUMP TO: [GET POST COOKIES ENV](#)

- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-portier-2.0/lib/sinatra/browserid.rb in **set_key**
- ```
44. # For that, fetch the public key from the LA instance (TODO: Do that beforehand for
 trusted instances, and generally cache the key)
45. public_key_jwks_uri = Addressable::URI.parse(settings.browserid_url + '/keys.json')
46. public_key_jwks = ::JSON.parse(URI.parse(public_key_jwks_uri).read)
47. public_key = OpenSSL::PKey::RSA.new
48. if public_key.respond_to? :set_key
49. # Set n and d via the new set_key function, as direct access to n and e is blocked for
 some ruby and openssl versions.
50. # Note that we have no d, as this is a public key, which would be the third param
51. public_key.set_key((OpenSSL::BN.new Ur1SafeBase64.decode64(public_key_jwks["keys"]
 [0]["n"]), 2),
52. (OpenSSL::BN.new Ur1SafeBase64.decode64(public_key_jwks["keys"][0]["e"]), 2),
53. nil)
54. else
55. public_key.e = OpenSSL::BN.new Ur1SafeBase64.decode64(public_key_jwks["keys"][0]["e"]),
 2
56. public_key.n = OpenSSL::BN.new Ur1SafeBase64.decode64(public_key_jwks["keys"][0]["n"]),
 2
57. end
58.
```
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-portier-2.0/lib/sinatra/browserid.rb in

**block in registered**

- 44. # For that, fetch the public key from the LA instance (TODO: Do that beforehand for trusted instances, and generally cache the key)
- 45. public\_key\_jwks\_uri = Addressable::URI.parse(settings.browserid\_url + '/keys.json')
- 46. public\_key\_jwks = ::JSON.parse(URI.parse(public\_key\_jwks\_uri).read)
- 47. public\_key = OpenSSL::PKey::RSA.new
- 48. if public\_key.respond\_to? :set\_key
- 49. # Set n and d via the new set\_key function, as direct access to n and e is blocked for some ruby and openssl versions.
- 50. # Note that we have no d, as this is a public key, which would be the third param
- 51. public\_key.set\_key( (OpenSSL::BN.new UriSafeBase64.decode64(public\_key\_jwks["keys"][0]["n"]), 2),
- 52. (OpenSSL::BN.new UriSafeBase64.decode64(public\_key\_jwks["keys"][0]["e"]), 2),
- 53. nil)
- 54. else
- 55. public\_key.e = OpenSSL::BN.new UriSafeBase64.decode64(public\_key\_jwks["keys"][0]["e"]), 2
- 56. public\_key.n = OpenSSL::BN.new UriSafeBase64.decode64(public\_key\_jwks["keys"][0]["n"]), 2
- 57. end
- 58.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **call**
- 1678.
- 1679. pattern = compile(path, route\_mustermann\_opts)
- 1680. method\_name = "#{verb} #{path}"
- 1681. unbound\_method = generate\_method(method\_name, &block)
- 1682. conditions, @conditions = @conditions, []
- 1683. wrapper = block.arity != 0 ?
- 1684. proc { |a, p| unbound\_method.bind(a).call(\*p) } :
- 1685. proc { |a, p| unbound\_method.bind(a).call }
- 1686.
- 1687. [ pattern, conditions, wrapper ]
- 1688. end
- 1689.
- 1690. def compile(path, route\_mustermann\_opts = {})
- 1691. Mustermann.new(path, \*\*mustermann\_opts.merge(route\_mustermann\_opts))
- 1692. end
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **block in compile!**
- 1678.
- 1679. pattern = compile(path, route\_mustermann\_opts)
- 1680. method\_name = "#{verb} #{path}"
- 1681. unbound\_method = generate\_method(method\_name, &block)
- 1682. conditions, @conditions = @conditions, []
- 1683. wrapper = block.arity != 0 ?
- 1684. proc { |a, p| unbound\_method.bind(a).call(\*p) } :
- 1685. proc { |a, p| unbound\_method.bind(a).call }
- 1686.
- 1687. [ pattern, conditions, wrapper ]
- 1688. end
- 1689.

```
1690. def compile(path, route_mustermann_opts = {})
1691. Mustermann.new(path, **mustermann_opts.merge(route_mustermann_opts))
1692. end
```

- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **block (3 levels) in route!**
- 1016. def route!(base = settings, pass\_block = nil)
1017. if routes = base.routes[@request.request\_method]
1018. routes.each do |pattern, conditions, block|
1019. response.delete\_header('Content-Type') unless @pinned\_response
1020.
1021. returned\_pass\_block = process\_route(pattern, conditions) do |\*args|
1022. env['sinatra.route'] = "#{@request.request\_method} #{pattern}"
1023. route\_eval { block[\*args] }
1024. end
1025.
1026. # don't wipe out pass\_block in superclass
1027. pass\_block = returned\_pass\_block if returned\_pass\_block
1028. end
1029. end
1030.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **route\_eval**
- 1035.
1036. route\_eval(&pass\_block) if pass\_block
1037. route\_missing
1038. end
1039.
1040. # Run a route block and throw :halt with the result.
1041. def route\_eval
1042. throw :halt, yield
1043. end
1044.
1045. # If the current request matches pattern and conditions, fill params
1046. # with keys and call the given block.
1047. # Revert params afterwards.
1048. #
1049. # Returns pass block.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **block (2 levels) in route!**
- 1016. def route!(base = settings, pass\_block = nil)
1017. if routes = base.routes[@request.request\_method]
1018. routes.each do |pattern, conditions, block|
1019. response.delete\_header('Content-Type') unless @pinned\_response
1020.
1021. returned\_pass\_block = process\_route(pattern, conditions) do |\*args|
1022. env['sinatra.route'] = "#{@request.request\_method} #{pattern}"
1023. route\_eval { block[\*args] }
1024. end
1025.
1026. # don't wipe out pass\_block in superclass
1027. pass\_block = returned\_pass\_block if returned\_pass\_block
1028. end

```
1029. end
1030.
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in block in process_route
• 1064. @params[:captures] = force_encoding(captures) unless captures.nil? || captures.empty?
1065. else
1066. values += params.values.flatten
1067. end
1068.
1069. catch(:pass) do
1070. conditions.each { |c| throw :pass if c.bind(self).call == false }
1071. block ? block[self, values] : yield(self, values)
1072. end
1073. rescue
1074. @env['sinatra.error.params'] = @params
1075. raise
1076. ensure
1077. params ||= {}
1078. params.each { |k, _| @params.delete(k) } unless @env['sinatra.error.params']
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in catch
• 1062. captures = pattern.match(route).captures.map { |c| URI_INSTANCE.unescape(c) if c }
1063. values += captures
1064. @params[:captures] = force_encoding(captures) unless captures.nil? || captures.empty?
1065. else
1066. values += params.values.flatten
1067. end
1068.
1069. catch(:pass) do
1070. conditions.each { |c| throw :pass if c.bind(self).call == false }
1071. block ? block[self, values] : yield(self, values)
1072. end
1073. rescue
1074. @env['sinatra.error.params'] = @params
1075. raise
1076. ensure
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in process_route
• 1062. captures = pattern.match(route).captures.map { |c| URI_INSTANCE.unescape(c) if c }
1063. values += captures
1064. @params[:captures] = force_encoding(captures) unless captures.nil? || captures.empty?
1065. else
1066. values += params.values.flatten
1067. end
1068.
1069. catch(:pass) do
1070. conditions.each { |c| throw :pass if c.bind(self).call == false }
1071. block ? block[self, values] : yield(self, values)
1072. end
1073. rescue
1074. @env['sinatra.error.params'] = @params
1075. raise
1076. ensure
```

- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **block in route!**
- 1014.  
1015. # Run routes defined on the class and all superclasses.  
1016. def route!(base = settings, pass\_block = nil)  
1017. if routes = base.routes[@request.request\_method]  
1018. routes.each do |pattern, conditions, block|  
1019. response.delete\_header('Content-Type') unless @pinned\_response  
1020.  
1021. returned\_pass\_block = process\_route(pattern, conditions) do |\*args|  
1022. env['sinatra.route'] = "#{@request.request\_method} #{pattern}"  
1023. route\_eval { block[\*args] }  
1024. end  
1025.  
1026. # don't wipe out pass\_block in superclass  
1027. pass\_block = returned\_pass\_block if returned\_pass\_block  
1028. end
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **each**
- 1011. block.call(result) if block\_given?  
1012. end  
1013. end  
1014.  
1015. # Run routes defined on the class and all superclasses.  
1016. def route!(base = settings, pass\_block = nil)  
1017. if routes = base.routes[@request.request\_method]  
1018. routes.each do |pattern, conditions, block|  
1019. response.delete\_header('Content-Type') unless @pinned\_response  
1020.  
1021. returned\_pass\_block = process\_route(pattern, conditions) do |\*args|  
1022. env['sinatra.route'] = "#{@request.request\_method} #{pattern}"  
1023. route\_eval { block[\*args] }  
1024. end  
1025.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **route!**
- 1011. block.call(result) if block\_given?  
1012. end  
1013. end  
1014.  
1015. # Run routes defined on the class and all superclasses.  
1016. def route!(base = settings, pass\_block = nil)  
1017. if routes = base.routes[@request.request\_method]  
1018. routes.each do |pattern, conditions, block|  
1019. response.delete\_header('Content-Type') unless @pinned\_response  
1020.  
1021. returned\_pass\_block = process\_route(pattern, conditions) do |\*args|  
1022. env['sinatra.route'] = "#{@request.request\_method} #{pattern}"  
1023. route\_eval { block[\*args] }  
1024. end  
1025.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in **block in dispatch!**

```
• 1133. end
1134.
1135. invoke do
1136. static! if settings.static? && (request.get? || request.head?)
1137. filter! :before do
1138. @pinned_response = !response['Content-Type'].nil?
1139. end
1140. route!
1141. end
1142. rescue ::Exception => boom
1143. invoke { handle_exception!(boom) }
1144. ensure
1145. begin
1146. filter! :after unless env['sinatra.static_file']
1147. rescue ::Exception => boom
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in block in invoke
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
1107. send_file path, options.merge(:disposition => nil)
1108. end
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in catch
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
1107. send_file path, options.merge(:disposition => nil)
1108. end
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in invoke
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
```

```
1107. send_file path, options.merge(:disposition => nil)
1108. end
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in dispatch!
• 1128. # Avoid passing frozen string in force_encoding
1129. @params.merge!(@request.params).each do |key, val|
1130. next unless val.respond_to?(:force_encoding)
1131. val = val.dup if val.frozen?
1132. @params[key] = force_encoding(val)
1133. end
1134.
1135. invoke do
1136. static! if settings.static? && (request.get? || request.head?)
1137. filter! :before do
1138. @pinned_response = !response['Content-Type'].nil?
1139. end
1140. route!
1141. end
1142. rescue ::Exception => boom
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in block in call!
• 942. @env = env
943. @params = IndifferentHash.new
944. @request = Request.new(env)
945. @response = Response.new
946. @pinned_response = nil
947. template_cache.clear if settings.reload_templates
948.
949. invoke { dispatch! }
950. invoke { error_block!(response.status) } unless @env['sinatra.error']
951.
952. unless @response['Content-Type']
953. if Array === body && body[0].respond_to?(:content_type)
954. content_type body[0].content_type
955. elsif default = settings.default_content_type
956. content_type default
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in block in invoke
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
1107. send_file path, options.merge(:disposition => nil)
1108. end
```

```
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in catch
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
1107. send_file path, options.merge(:disposition => nil)
1108. end
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in invoke
• 1105. env['sinatra.static_file'] = path
1106. cache_control(*settings.static_cache_control) if settings.static_cache_control?
1107. send_file path, options.merge(:disposition => nil)
1108. end
1109.
1110. # Run the block with 'throw :halt' support and apply result to the response.
1111. def invoke
1112. res = catch(:halt) { yield }
1113.
1114. res = [res] if Integer === res or String === res
1115. if Array === res and Integer === res.first
1116. res = res.dup
1117. status(res.shift)
1118. body(res.pop)
1119. headers(*res)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call!
• 942. @env = env
943. @params = IndifferentHash.new
944. @request = Request.new(env)
945. @response = Response.new
946. @pinned_response = nil
947. template_cache.clear if settings.reload_templates
948.
```



```
949. invoke { dispatch! }
950. invoke { error_block!(response.status) } unless @env['sinatra.error']
951.
952. unless @response['Content-Type']
953. if Array === body && body[0].respond_to?(:content_type)
954. content_type body[0].content_type
955. elsif default = settings.default_content_type
956. content_type default
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call
• 931. @template_cache = Tilt::Cache.new
932. @pinned_response = nil # whether a before! filter pinned the content-type
933. yield self if block_given?
934. end
935.
936. # Rack call interface.
937. def call(env)
938. dup.call!(env)
939. end
940.
941. def call!(env) # :nodoc:
942. @env = env
943. @params = IndifferentHash.new
944. @request = Request.new(env)
945. @response = Response.new
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-protection-2.2.0/lib/rack/protection
/xss_header.rb in call
• 11. #
12. # Options:
13. # xss_mode:: How the browser should prevent the attack (default: :block)
14. class XSSHeader < Base
15. default_options :xss_mode => :block, :nosniff => true
16.
17. def call(env)
18. status, headers, body = @app.call(env)
19. headers['X-XSS-Protection'] ||= "1; mode=#{options[:xss_mode]}" if html? headers
20. headers['X-Content-Type-Options'] ||= 'nosniff' if options[:nosniff]
21. [status, headers, body]
22. end
23. end
24. end
25. end
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-protection-2.2.0/lib/rack/protection
/path_traversal.rb in call
• 9. #
10. # Unescapes '/' and '.', expands +path_info+.
11. # Thus <tt>GET /foo/%2e%2e%2fbar</tt> becomes <tt>GET /bar</tt>.
12. class PathTraversal < Base
13. def call(env)
14. path_was = env["PATH_INFO"]
15. env["PATH_INFO"] = cleanup path_was if path_was && !path_was.empty?
16. app.call env
```

```
17. ensure
18. env["PATH_INFO"] = path_was
19. end
20.
21. def cleanup(path)
22. encoding = path.encoding
23. dot = '.'.encode(encoding)
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-protection-2.2.0/lib/rack/protection
 /json_csrf.rb in call
• 19. class JsonCsrf < Base
 20. default_options :allow_if => nil
 21.
 22. alias react deny
 23.
 24. def call(env)
 25. request = Request.new(env)
 26. status, headers, body = app.call(env)
 27.
 28. if has_vector?(request, headers)
 29. warn env, "attack prevented by #{self.class}"
 30.
 31. react_and_close(env, body) or [status, headers, body]
 32. else
 33. [status, headers, body]
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-protection-2.2.0/lib/rack/protection/base.rb
 in call
• 43. end
 44.
 45. def call(env)
 46. unless accepts? env
 47. instrument env
 48. result = react env
 49. end
 50. result or app.call(env)
 51. end
 52.
 53. def react(env)
 54. result = send(options[:reaction], env)
 55. result if Array === result and result.size == 3
 56. end
 57.
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-protection-2.2.0/lib/rack/protection
 /frame_options.rb in call
• 24. frame_options = options[:frame_options]
 25. frame_options = options[:frame_options].to_s.upcase unless frame_options.respond_to?
 :to_str
 26. frame_options.to_str
 27. end
 28. end
 29.
 30. def call(env)
```

```
31. status, headers, body = @app.call(env)
32. headers['X-Frame-Options'] ||= frame_options if html? headers
33. [status, headers, body]
34. end
35. end
36. end
37. end
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/logger.rb in call
• 10. end
 11.
 12. def call(env)
 13. logger = ::Logger.new(env[RACK_ERRORS])
 14. logger.level = @level
 15.
 16. env[RACK_LOGGER] = logger
 17. @app.call(env)
 18. end
 19. end
 20. end
• /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/common_logger.rb in call
• 31. # will not be logged, so if exception handling middleware are used,
 32. # they should be loaded after this middleware. Additionally, because
 33. # the logging happens after the request body has been fully sent, any
 34. # exceptions raised during the sending of the response body will
 35. # cause the request not to be logged.
 36. def call(env)
 37. began_at = Utils.clock_time
 38. status, headers, body = @app.call(env)
 39. headers = Utils::HeaderHash[headers]
 40. body = BodyProxy.new(body) { log(env, status, headers, began_at) }
 41. [status, headers, body]
 42. end
 43.
 44. private
 45.
• /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call
• 248. env['sinatra.commonlogger'] ? @app.call(env) : super
 249. end
 250.
 251. superclass.class_eval do
 252. alias call_without_check call unless method_defined? :call_without_check
 253. def call(env)
 254. env['sinatra.commonlogger'] = true
 255. call_without_check(env)
 256. end
 257. end
 258. end
 259.
 260. class BadRequest < TypeError #:nodoc:
 261. def http_status; 400 end
 262. end
```

- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call
  - 241. end
  - 242. end
  - 243.
  - 244. # Behaves exactly like Rack::CommonLogger with the notable exception that it does nothing,
  - 245. # if another CommonLogger is already in the middleware chain.
  - 246. class CommonLogger < Rack::CommonLogger
  - 247. def call(env)
  - 248. env['sinatra.commonlogger'] ? @app.call(env) : super
  - 249. end
  - 250.
  - 251. superclass.class\_eval do
  - 252. alias call\_without\_check call unless method\_defined? :call\_without\_check
  - 253. def call(env)
  - 254. env['sinatra.commonlogger'] = true
  - 255. call\_without\_check(env)
- /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/head.rb in call
  - 5. # all other requests unchanged.
  - 6. class Head
  - 7. def initialize(app)
  - 8. @app = app
  - 9. end
  - 10.
  - 11. def call(env)
  - 12. status, headers, body = @app.call(env)
  - 13.
  - 14. if env[REQUEST\_METHOD] == HEAD
  - 15. [  - 16. status, headers, Rack::BodyProxy.new([]) do
  - 17. body.close if body.respond\_to? :close
  - 18. end
  - 19. ]
- /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/method\_override.rb in call
  - 17. method = method\_override(env)
  - 18. if HTTP\_METHODS.include?(method)
  - 19. env[RACK\_METHODOVERRIDE\_ORIGINAL\_METHOD] = env[REQUEST\_METHOD]
  - 20. env[REQUEST\_METHOD] = method
  - 21. end
  - 22. end
  - 23.
  - 24. @app.call(env)
  - 25. end
  - 26.
  - 27. def method\_override(env)
  - 28. req = Request.new(env)
  - 29. method = method\_override\_param(req) ||
  - 30. env[HTTP\_METHOD\_OVERRIDE\_HEADER]
  - 31. begin
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/show\_exceptions.rb in call

- 15. def @@eats\_errors.puts(\*) end  
16.  
17. def initialize(app)  
18. @app = app  
19. end  
20.  
21. def call(env)  
22. @app.call(env)  
23. rescue Exception => e  
24. errors, env["rack.errors"] = env["rack.errors"], @@eats\_errors  
25.  
26. if prefers\_plain\_text?(env)  
27. content\_type = "text/plain"  
28. body = dump\_exception(e)  
29. else
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call
- 211. # Some Rack handlers (Rainbows!) implement an extended body object protocol, however,  
212. # some middleware (namely Rack::Lint) will break it by not mirroring the methods in  
question.  
213. # This middleware will detect an extended body object and will make sure it reaches the  
214. # handler directly. We do this here, so our middleware and middleware set up by the app  
will  
215. # still be able to run.  
216. class ExtendedRack < Struct.new(:app)  
217. def call(env)  
218. result, callback = app.call(env), env['async.callback']  
219. return result unless callback and async?(\*result)  
220. after\_response { callback.call result }  
221. setup\_close(env, \*result)  
222. throw :async  
223. end  
224.  
225. private
- /home/ben/.local/share/gem/ruby/3.0.0/gems/sinatra-2.2.0/lib/sinatra/base.rb in call
- 1985. end  
1986.  
1987. def helpers  
1988. @instance  
1989. end  
1990.  
1991. def call(env)  
1992. @stack.call(env)  
1993. end  
1994.  
1995. def inspect  
1996. "#<#{@instance.class} app\_file=#{settings.app\_file.inspect}>"  
1997. end  
1998. end  
1999.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/session/abstract/id.rb in context

- 259. def call(env)  
260. context(env)  
261. end  
262.  
263. def context(env, app = @app)  
264. req = make\_request env  
265. prepare\_session(req)  
266. status, headers, body = app.call(req.env)  
267. res = Rack::Response::Raw.new status, headers  
268. commit\_session(req, res)  
269. [status, headers, body]  
270. end  
271.  
272. private  
273.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/rack-2.2.3/lib/rack/session/abstract/id.rb in **call**
- 253. @key = @default\_options.delete(:key)  
254. @cookie\_only = @default\_options.delete(:cookie\_only)  
255. @same\_site = @default\_options.delete(:same\_site)  
256. initialize\_sid  
257. end  
258.  
259. def call(env)  
260. context(env)  
261. end  
262.  
263. def context(env, app = @app)  
264. req = make\_request env  
265. prepare\_session(req)  
266. status, headers, body = app.call(req.env)  
267. res = Rack::Response::Raw.new status, headers
- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/configuration.rb in **call**
- 245. def initialize(config, app)  
246. @config = config  
247. @app = app  
248. end  
249.  
250. def call(env)  
251. env[Const::PUMA\_CONFIG] = @config  
252. @app.call(env)  
253. end  
254. end  
255.  
256. # Indicate if there is a properly configured app  
257. #  
258. def app\_configured?  
259. @options[:app] || File.exist?(rackup)
- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/request.rb in **block in handle\_request**
- 70. # array, we will invoke them when the request is done.  
71. #

```
72. after_reply = env[RACK_AFTER_REPLY] = []
73.
74. begin
75. begin
76. status, headers, res_body = @thread_pool.with_force_shutdown do
77. @app.call(env)
78. end
79.
80. return :async if client.hijacked
81.
82. status = status.to_i
83.
84. if status == -1
```

- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/thread\_pool.rb in **with\_force\_shutdown**
- 333. # provided block if the thread is forced to shutdown during execution.  
334. def with\_force\_shutdown  
335. t = Thread.current  
336. @shutdown\_mutex.synchronize do  
337. raise ForceShutdown if @force\_shutdown  
338. t[:with\_force\_shutdown] = true  
339. end  
340. yield  
341. ensure  
342. t[:with\_force\_shutdown] = false  
343. end  
344.  
345. # Tell all threads in the pool to exit and wait for them to finish.  
346. # Wait +timeout+ seconds then raise +ForceShutdown+ in remaining threads.  
347. # Next, wait an extra +grace+ seconds then force-kill remaining threads.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/request.rb in **handle\_request**
- 69. # A rack extension. If the app writes #call'ables to this  
70. # array, we will invoke them when the request is done.  
71. #  
72. after\_reply = env[RACK\_AFTER\_REPLY] = []  
73.  
74. begin  
75. begin  
76. status, headers, res\_body = @thread\_pool.with\_force\_shutdown do  
77. @app.call(env)  
78. end  
79.  
80. return :async if client.hijacked  
81.  
82. status = status.to\_i  
83.
- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/server.rb in **process\_client**
- 434.  
435. with\_force\_shutdown(client) do  
436. client.finish(@first\_data\_timeout)  
437. end

```

438.
439. while true
440. @requests_count += 1
441. case handle_request(client, buffer, requests + 1)
442. when false
443. break
444. when :async
445. close_socket = false
446. break
447. when true
448. buffer.reset

```

- /home/ben/.local/share/gem/ruby/3.0.0/gems/puma-5.6.4/lib/puma/thread\_pool.rb in **block in spawn\_thread**
- 140. end
 

```

141.
142. if @clean_thread_locals
143. ThreadPool.clean_thread_locals
144. end
145.
146. begin
147. @out_of_band_pending = true if block.call(work, *extra)
148. rescue Exception => e
149. STDERR.puts "Error reached top of thread-pool: #{e.message} (#{e.class})"
150. end
151. end
152. end
153.
154. @workers << th

```

## GET

No GET data.

## POST

### Variable

```

id_token "eyJhbGciOiJSUzI1NiIsImtpZCI6IiIhtUXJWZVJ5UHVBX05vWXdlUXdBTKhGby1oZWc5aC1tNnRz0dUn7KKSsYIrGqSvJCH-Bk93Pmjb57jkt6yhKoowxVsQTOhJZiePPH9BQSqyZD93gaNqMvOsW
state ""

```

## COOKIES

| Variable     | Value                                                             |
|--------------|-------------------------------------------------------------------|
| rack.session | "a4e1ae45400de1f95cf4877585bd7b3a0dc86ccdde9dc9238b15944c849cdb8" |

## Rack ENV

### Variable



```

CONTENT_LENGTH 777
CONTENT_TYPE application/x-www-form-urlencoded
GATEWAY_INTERFACE CGI/1.2
HTTP_ACCEPT text/html,application/xhtml+xml,application/xml;q=0.9,image/a
HTTP_ACCEPT_ENCODING gzip, deflate
HTTP_ACCEPT_LANGUAGE en-US,en;q=0.5
HTTP_CONNECTION keep-alive
HTTP_COOKIE rack.session=a4e1ae45400de1f95cf4877585bd7b3a0dc86ccddd
HTTP_DNT 1
HTTP_HOST 192.168.1.235:9292
HTTP_UPGRADE_INSECURE_REQUESTS 1
HTTP_USER_AGENT Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/
HTTP_VERSION HTTP/1.1
PATH_INFO /_browserid_assert
QUERY_STRING
REMOTE_ADDR 192.168.1.85
REQUEST_METHOD POST
REQUEST_PATH /_browserid_assert
REQUEST_URI /_browserid_assert
SCRIPT_NAME
SERVER_NAME 192.168.1.235
SERVER_PORT 9292
SERVER_PROTOCOL HTTP/1.1
SERVER_SOFTWARE puma 5.6.4 Birdie's Version
#<Puma::Configuration:0x000055969cc40ad0 @options=#<Pu
:logger=>#<IO:<STDOUT>>, :persistent_timeout=>20, :first_c
@plugins=[]>, @default_dsl=#<Puma::DSL:0x000055969cc3b
:first_data_timeout=>30, :raise_exception_on_sigterm=>true, :
puma.config
puma.request_body_wait 1
puma.socket #<TCPSocket:fd 14, AF_INET, 192.168.1.235, 9292>
rack.after_reply []
rack.errors #<Object:0x000055969d1ab808>
rack.hijack #<Puma::Client:0xd804 @ready=true>
rack.hijack? true
rack.input #<StringIO:0x00007fa2d47e81e0>
rack.logger #<Logger:0x00007fa2d05eb4b8 @level=1, @progname=nil, @
rack.multiprocess false
rack.multithread true
rack.request.cookie_hash {"rack.session"=>"a4e1ae45400de1f95cf4877585bd7b3a0dc86
rack.request.cookie_string rack.session=a4e1ae45400de1f95cf4877585bd7b3a0dc86ccddd
rack.request.form_hash {"id_token"=>"eyJhbGciOiJSUzI1NiIsImtpZCI6IiIhtUXJWZV.
z0dUn7KKSsYIrGqSvJCH-Bk93Pmjb57jkt6yhKoowxVsQTOI

```

```

rack.request.form_input #<StringIO:0x00007fa2d47e81e0>
rack.request.form_vars id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IlhtUXJWZVJ5UH
 z0dUn7KKSsYlRqSvJCH-Bk93Pmjb57jkt6yhKoowxVsQTOI
rack.request.query_hash {}
rack.request.query_string
rack.run_once false
rack.session #<Rack::Session::Abstract::SessionHash:0xd818 not yet loaded
 {:path=>"/", :domain=>nil, :expire_after=>259200000, :secure=
 /share/gem/ruby/3.0.0/gems/sqlite3-1.4.2/lib/sqlite3/database.rb
 <SQLite3::ResultSet:0x00007fa2d47e3b90 @db=#<SQLite3::I
 @authorizer=nil, @encoding=#<Encoding:UTF-8>, @busy_ha
 @type_translation=nil, @type_translator=#<Proc:0x000055969
 /gems/sqlite3-1.4.2/lib/sqlite3/database.rb:722 (lambda)>, @rea
 @delete=#<SQLite3::Statement:0x000055969d74f570 @conne
 @authorizer=nil, @encoding=#<Encoding:UTF-8>, @busy_ha
 @type_translation=nil, @type_translator=#<Proc:0x000055969
 /database.rb:722 (lambda)>, @readonly=false>, @stmt=#<SQL
 <SQLite3::Statement:0x000055969d74eaf8 @connection=#<St
 @authorizer=nil, @encoding=#<Encoding:UTF-8>, @busy_ha
 @type_translation=nil, @type_translator=#<Proc:0x000055969
 /lib/sqlite3/database.rb:722 (lambda)>, @readonly=false>, @re
 #<SQLite3::Statement:0x000055969d74f778 @connection=#<
 @encoding=#<Encoding:UTF-8>, @busy_handler=nil, @colla
 @type_translation=nil, @type_translator=#<Proc:0x000055969
 /database.rb:722 (lambda)>, @readonly=false>, @remainder="
 <SQLite3::ResultSet:0x00007fa2d47e3848 @db=#<SQLite3::I
 @encoding=#<Encoding:UTF-8>, @busy_handler=nil, @colla
 @type_translator=#<Proc:0x000055969dc8de88 /home/ben/.lo
 @readonly=false>, @remainder="", @columns=nil, @types=ni
 :create, :each_key]>}
rack.session.options []
rack.tempfiles []
rack.url_scheme http
rack.version [1, 6]
sinatra.accept [#<Sinatra::Request::AcceptEntry:0x00007fa2d06d2930 @entr
 @type="application/xml", @params={}, @q=0.9>, #<Sinatra::
sinatra.commonlogger true
sinatra.error #<OpenSSL::PKey::PKeyError: rsa#set_key= is incompatible v
 {"id_token"=>"eyJhbGciOiJSUzI1NiIsImtpZCI6IlhtUXJWZV
 z0dUn7KKSsYlRqSvJCH-Bk93Pmjb57jkt6yhKoowxVsQTOI
sinatra.error.params POST /_browserid_assert
sinatra.route POST /_browserid_assert

```

You're seeing this error because you have enabled the `show_exceptions` setting.