

Ruby - Bug #20686

URI::HTTPS can build URI with blank, invalid host

08/19/2024 07:40 PM - ronricardo (Roniece Ricardo)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	3.4.0+	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN

Description

In Ruby 3.4.0+, calling `URI::HTTPS.build(host: "")` does not raise `URI::InvalidComponentError` as expected. Instead, it returns `#<URI::HTTPS https://>`

I think this was introduced in [this PR](#).

Steps to Reproduce

1. Environment:

- Ruby Version: 3.4.0+

2. Steps:

- Open an IRB session.
- Run:

```
URI::HTTPS.build(host: "")
```

3. Expected Behavior:

- `URI::InvalidComponentError` should be raised due to the invalid empty host component.

4. Actual Behavior:

- Returns `#<URI::HTTPS https://>` without raising an error.

Ruby 3.1.4:

```
irb(main):008:0> RUBY_VERSION
=> "3.1.4"
irb(main):009:0> URI::HTTPS.build(host: "")
/home/vscode/.rbenv/versions/3.1.4/lib/ruby/3.1.0/uri/generic.rb:601:in
`check_host': bad component (expected host component): (URI::InvalidComponentError)
```

Ruby 3.4.0:

```
irb(...):015> RUBY_VERSION
=> "3.4.0"
irb(...):016> URI::HTTPS.build(host: "")
=> #<URI::HTTPS https://>
```

History

#1 - 08/19/2024 08:04 PM - jeremyevans0 (Jeremy Evans)

ronricardo (Roniece Ricardo) wrote:

In Ruby 3.4.0+, calling `URI::HTTPS.build(host: "")` does not raise `URI::InvalidComponentError` as expected. Instead, it returns `#<URI::HTTPS https://>`

I think this was introduced in [this PR](#).

That PR only affects #to_s, not .build, and is unrelated. This was caused by the RFC 2396 -> RFC 3986 parser change:

```
URI::HTTPS.new(nil, nil, "", nil, nil, nil, nil, nil, nil, URI::RFC3986_PARSER, true)
# => #<URI::HTTPS //>

URI::HTTPS.new(nil, nil, "", nil, nil, nil, nil, nil, nil, URI::RFC2396_PARSER, true)
# /home/jeremy/tmp/uri/lib/uri/generic.rb:601:in `check_host': bad component (expected host component): (URI::
InvalidComponentError)
```

It appears RFC 3986 allows empty hosts (<https://datatracker.ietf.org/doc/html/rfc3986#section-3.2.2>: reg-name = *(unreserved / pct-encoded / sub-delims)), so I think this is not a bug, but an expected behavior change.

#2 - 08/19/2024 10:45 PM - jhawthorn (John Hawthorn)

jeremyevans0 (Jeremy Evans) wrote in [#note-1](#):

It appears RFC 3986 allows empty hosts (<https://datatracker.ietf.org/doc/html/rfc3986#section-3.2.2>: reg-name = *(unreserved / pct-encoded / sub-delims)), so I think this is not a bug, but an expected behavior change.

It's allowed by the ABNF, but the next paragraph states that it isn't valid for HTTP/HTTPS URIs

If the URI scheme defines a default for host, then that default applies when the host subcomponent is undefined or when the registered name is empty (zero length). For example, the "file" URI scheme is defined so that no authority, an empty host, and "localhost" all mean the end-user's machine, **whereas the "http" scheme considers a missing authority or empty host invalid.**

#3 - 08/19/2024 10:59 PM - jhawthorn (John Hawthorn)

Interestingly RFC2396_PARSER seems to allow nil for a host but not empty string, so the newer behaviour is at least more consistent. It does seem like we are missing some expected validation here though.

```
> URI::HTTPS.new(nil, nil, "", nil, nil, nil, nil, nil, nil, URI::RFC2396_Parser.new, true)
/Users/jhawthorn/.rubies/ruby-3.3.2/lib/ruby/3.3.0/uri/generic.rb:601:in `check_host': bad component (expected
host component): (URI::InvalidComponentError)
    from /Users/jhawthorn/.rubies/ruby-3.3.2/lib/ruby/3.3.0/uri/generic.rb:640:in `host='
    ...
>> URI::HTTPS.new(nil, nil, nil, nil, nil, nil, nil, nil, nil, URI::RFC2396_Parser.new, true)
=> #<URI::HTTPS >
```

#4 - 08/19/2024 11:46 PM - jeremyevans0 (Jeremy Evans)

jhawthorn (John Hawthorn) wrote in [#note-2](#):

jeremyevans0 (Jeremy Evans) wrote in [#note-1](#):

It appears RFC 3986 allows empty hosts (<https://datatracker.ietf.org/doc/html/rfc3986#section-3.2.2>: reg-name = *(unreserved / pct-encoded / sub-delims)), so I think this is not a bug, but an expected behavior change.

It's allowed by the ABNF, but the next paragraph states that it isn't valid for HTTP/HTTPS URIs

If the URI scheme defines a default for host, then that default applies when the host subcomponent is undefined or when the registered name is empty (zero length). For example, the "file" URI scheme is defined so that no authority, an empty host, and "localhost" all mean the end-user's machine, **whereas the "http" scheme considers a missing authority or empty host invalid.**

Thank you for pointing that out. I obviously should have read a little further. I submitted a pull request to reject empty host for URI::HTTP{,S}: <https://github.com/ruby/uri/pull/116>