

gleam-wisp / wisp Public

<> Code Issues 16 Pull requests 3 Actions Security and quality 2

# Commit 7a97874

 ipil committed last week · ✓ 5 / 5

Fix body limits

 main

1 parent [161118c](#) commit 7a97874

3 files changed +63 -7 lines changed

↑ Top 

Filter files...

CHANGELOG.md

src

wisp.gleam

test

wisp\_test.gleam

3 files changed +63 -7 lines changed

Search within code

CHANGELOG.md

@@ -5,6 +5,8 @@

```

5 5 - The Mist web server related module has been moved to the `wisp_mist` adapter
6 6   package. A `wisp_ewe` package has been created too!
7 7 - `send_file` failing will now result in an internal server error being sent.
8 8 + - Fixed a bug where the max body sizes would not be respected for chunked
9 9 +   request body streams.
```

```

8 10
9 11   ## v2.2.1 - 2026-03-08
10 12
```

src/wisp.gleam		...
↑		@@ -1172,6 +1172,7 @@ fn multipart_body(
1172	1172	http.MoreRequiredForBody(chunk, parse) -> {
1173	1173	let parse = fn_with_bad_request_error(parse, invalid_form)
1174	1174	let reader = BufferedReader(reader, <<>>)
1175	+	use quota <- result.try(decrement_quota(quota, size_read))
1175	1176	use data <- result.try(append(data, chunk))
1176	1177	multipart_body(reader, parse, boundary, chunk_size, quota, append,
		data)
1177	1178	}
↓		@@ -1242,6 +1243,8 @@ fn multipart_headers(
↑		
1242	1243	}
1243	1244	}
1244	1245	let reader = BufferedReader(reader, <<>>)
1246	+	let size_read = bit_array.byte_size(chunk)
1247	+	use quotas <- result.try(decrement_body_quota(quotas, size_read))
1245	1248	multipart_headers(reader, parse, chunk_size, quotas)
1246	1249	}
1247	1250	}
↓		@@ -2000,18 +2003,26 @@ pub fn get_cookie(
↑		
2000	2003	// Testing
2001	2004	//
2002	2005	
2003	-	// TODO: chunk the body
2004	2006	@internal
2005	2007	pub fn create_canned_connection(
2006	2008	body: BitArray,
2007	2009	secret_key_base: String,
2008	2010	) -> internal.Connection {
2009	-	internal.make_connection(
2010	-	fn(_size) {
2011	-	Ok(internal.Chunk(body, fn(_size) { Ok(internal.ReadingFinished) })))
2012	-	},
2013	-	secret_key_base,
2014	-	)
2011	+	internal.make_connection(canned_reader(body), secret_key_base)
2012	+	}

```

2013 +
2014 + fn canned_reader(data: BitArray) -> internal.Reader {
2015 +   fn(chunk_size) {
2016 +     case bit_array.byte_size(data) {
2017 +       0 -> Ok(internal.ReadingFinished)
2018 +       size -> {
2019 +         let take = int.min(chunk_size, size)
2020 +         let assert Ok(chunk) = bit_array.slice(data, 0, take)
2021 +         let assert Ok(rest) = bit_array.slice(data, take, size - take)
2022 +         Ok(internal.Chunk(chunk, canned_reader(rest)))
2023 +       }
2024 +     }
2025 +   }
2015 2026 }
2016 2027
2017 2028 /// Protects against Cross-Site Request Forgery (CSRF) attacks by checking
the

```



test/wisp\_test.gleam



@@ -885,6 +885,49 @@ Content-Disposition: form-data; name=\"one\"\\r

```

885 885 )
886 886 }
887 887
888 + // https://github.com/gleam-wisp/wisp/issues/108
889 + pub fn multipart_form_body_too_big_across_chunks_test() {
890 +   let data = "--b\\r
891 + Content-Disposition: form-data; name=\"x\"\\r
892 + \\r
893 + \" <> string.repeat(\"A\", 200) <> \"\\r
894 + --b--\\r
895 + \"
896 +   assert simulate.request(http.Post, "/")
897 +     |> simulate.string_body(data)
898 +     |> request.set_header("content-type", "multipart/form-data; boundary=b")
899 +     |> wisp.set_max_body_size(120)
900 +     |> wisp.set_read_chunk_size(100)
901 +     |> form_handler(fn(_) { panic as "should be unreachable" })
902 +     == Response(
903 +       413,

```

```
904 +     [{"content-type", "text/plain"}],
905 +     wisp.Text("Content too large"),
906 +   )
907 + }
908 +
909 + // https://github.com/gleam-wisp/wisp/issues/108
910 + pub fn multipart_form_headers_too_big_across_chunks_test() {
911 +   let data =
912 +     "--b\r
913 + Content-Disposition: form-data; name=\"x\"\r
914 + \r
915 + 1\r
916 + --b--\r
917 + "
918 +   assert simulate.request(http.Post, "/")
919 +     |> simulate.string_body(data)
920 +     |> request.set_header("content-type", "multipart/form-data; boundary=b")
921 +     |> wisp.set_max_body_size(10)
922 +     |> wisp.set_read_chunk_size(4)
923 +     |> form_handler(fn(_) { panic as "should be unreachable" })
924 +     == Response(
925 +       413,
926 +       [{"content-type", "text/plain"}],
927 +       wisp.Text("Content too large"),
928 +     )
929 + }
930 +
888 931   pub fn multipart_form_no_boundary_test() {
889 932     let data =
890 933       "--theboundary\r
```



## Comments 0



Please [sign in](#) to comment.