

## ✓ Merge pull request from GHSA-qvxg-wjxc-r4gg

[Browse files](#)

\* Add test for crash

\* The sensible option

\* Crude fix

\* Throw an error if we hit a nesting limit

\* Catch error in test

main  
4.67.4 ... 4.61.1

OxTim committed on Jun 61 parent d2d1910 commit 6c63226a4ab82ce53730eb1afb9ca63866fcf033

Showing 4 changed files with 20 additions and 4 deletions.

Split Unified

8 Sources/Vapor/URLEncodedForm/URLEncodedFormData.swift

```
54 54 internal struct URLEncodedFormData: ExpressibleByArrayLiteral, ExpressibleByStringLiteral
55 55     var values: [URLQueryFragment]
56 56     var children: [String: URLEncodedFormData]
57 57 +     let maxRecursionDepth = 100
58 58
59 59     var hasOnlyValues: Bool {
60 60         return children.count == 0
61 61         self.children = Dictionary(uniqueKeysWithValues: dictionaryLiteral)
62 62     }
63 63
64 64 - mutating func set(value: URLQueryFragment, forPath path: [String]) {
65 65 + mutating func set(value: URLQueryFragment, forPath path: [String], recursionDepth: Int) {
66 66 +     guard recursionDepth <= maxRecursionDepth else {
67 67 +         throw URLEncodedFormError.reachedNestingLimit
68 68 +     }
69 69 +     guard let firstElement = path.first else {
70 70         self.values.append(value)
71 71         return
72 72     } else {
73 73         child = []
```

```

103 107      }
104      -      child.set(value: value, forPath: Array(path[1...]))
      108      +      try child.set(value: value, forPath: Array(path[1...]), recursionDepth: recursion
105 109      self.children[firstElement] = child
106 110      }
107 111      }

```

3 Sources/Vapor/URLEncodedForm/URLEncodedFormError.swift

```

...    @@ -1,6 +1,7 @@
1      1      /// Errors thrown while encoding/decoding `application/x-www-form-urlencoded` data.
2      2      enum URLEncodedFormError: Error {
3      3          case malformedKey(key: Substring)
      4      4      +      case reachedNestingLimit
4      5      }
5      6
6      7      extension URLEncodedFormError: AbortError {
12     13          switch self {
13     14              case .malformedKey(let path):
14     15                  return "Malformed form-urlencoded key encountered: \(path)"
      16     16      +      case .reachedNestingLimit:
      17     17      +      return "The data supplied is too nested"
15     18          }
16     19      }
17     20  }

```

4 Sources/Vapor/URLEncodedForm/URLEncodedFormParser.swift

```

14     14      switch kv.count {
15     15      case 1:
16     16          let value = String(kv[0])
17     17      -      result.set(value: .urlEncoded(value), forPath: [])
      17     17      +      try result.set(value: .urlEncoded(value), forPath: [], recursionDepth: 0)
18     18      case 2:
19     19          let key = kv[0]
20     20          let value = String(kv[1])
21     21      -      result.set(value: .urlEncoded(value), forPath: try parseKey(key: Substring))
      21     21      +      try result.set(value: .urlEncoded(value), forPath: try parseKey(key: Substring))
22     22      default:
23     23          //Empty `&&`
24     24          continue

```

9 Tests/VaporTests/URLEncodedFormTests.swift

574	574	])
575	575	XCTAssertEqual(data, "test=%26%3B!\$'(),/:%3D%3F@~")
576	576	}
	577	+
	578	+ func testHeavilyNestedArray() throws {
	579	+ var body = "x"
	580	+ body += String(repeating: "[]", count: 80000)
	581	+ body += "=y"
	582	+ struct Foo: Content {}
	583	+ XCTAssertThrowsError(try URLEncodedFormDecoder().decode(Foo.self, from: body))
	584	+ XCTAssert(true, "We should not have crashed")
	585	+ }
577	586	}
578	587	
579	588	private struct User: Codable, Equatable {

0 comments on commit 6c63226

Please [sign in](#) to comment.