

[Foundation](#) / JSONEncoder

Class



JSONEncoder

An object that encodes instances of a data type as JSON objects.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 8.0+ | macOS 10.10+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class JSONEncoder
```

Mentioned in

-  Uploading data to a website
-  Encoding and Decoding Custom Types

Overview

The example below shows how to encode an instance of a simple `GroceryProduct` type from a JSON object. The type adopts [Codable](#) so that it's encodable as JSON using a [JSONEncoder](#) instance.

```
struct GroceryProduct: Codable {
    var name: String
    var points: Int
    var description: String?
}

let pear = GroceryProduct(name: "Pear", points: 250, description: "A ripe pear.")

let encoder = JSONEncoder()
encoder.outputFormatting = .prettyPrinted
```

```
let data = try encoder.encode(pear)
print(String(data: data, encoding: .utf8!))

/* Prints:
{
  "name" : "Pear",
  "points" : 250,
  "description" : "A ripe pear."
}
*/
```

Topics

First Steps

`init()`

Creates a new, reusable JSON encoder with the default formatting settings and encoding strategies.

`func encode<T>(T) throws -> Data`

Returns a JSON-encoded representation of the value you supply.

Customizing Encoding

`var outputFormatting: JSONEncoder.OutputFormatting`

A value that determines the readability, size, and element order of the encoded JSON object.

`struct OutputFormatting`

The output formatting options that determine the readability, size, and element order of an encoded JSON object.

`var keyEncodingStrategy: JSONEncoder.KeyEncodingStrategy`

A value that determines how to encode a type's coding keys as JSON keys.

`enum KeyEncodingStrategy`

The values that determine how to encode a type's coding keys as JSON keys.

`var userInfo: [CodingUserInfoKey : any Sendable]`

A dictionary you use to customize the encoding process by providing contextual information.

Encoding Dates

`var dateEncodingStrategy: JSONEncoder.DateEncodingStrategy`

The strategy used when encoding dates as part of a JSON object.

`enum DateEncodingStrategy`

The formatting strategies available for formatting dates when encoding a date as JSON.

Encoding Raw Data

`var dataEncodingStrategy: JSONEncoder.DataEncodingStrategy`

The strategy that an encoder uses to encode raw data.

`enum DataEncodingStrategy`

The strategies for encoding raw data.

Encoding Exceptional Numbers

`var nonConformingFloatEncodingStrategy: JSONEncoder.NonConformingFloatEncodingStrategy`

The strategy used by an encoder when it encounters exceptional floating-point values.

`enum NonConformingFloatEncodingStrategy`

The strategies for encoding nonconforming floating-point numbers, also known as IEEE 754 exceptional values.

Instance Methods

`func encode<T, C>(T, configuration: C.Type) throws -> Data`

`func encode<T>(T, configuration: T.EncodingConfiguration) throws -> Data`

Relationships

Conforms To

Copyable
NetworkEncoder
Sendable
SendableMetatype
TopLevelEncoder

See Also

JSON

`class` JSONDecoder

An object that decodes instances of a data type from JSON objects.

`class` JSONSerialization

An object that converts between JSON and the equivalent Foundation objects.