

[Discover](#)[Design](#)[Develop](#)[Distribute](#)[Support](#)[Account](#)[Documentation](#) [Foundation](#)[Archives and Serialization](#)[JSONDecoder](#)[Language: Swift](#)[API Changes: None](#)

JSONDecoder

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

Declaration

```
class JSONDecoder
```

Overview

The example below shows how to decode an instance of a simple `GroceryProduct` type from a JSON object. The type adopts `Codable` so that it's decodable using a `JSONDecoder` instance.

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

let json = """
{
    "name": "Durian",
    "points": 600,
    "description": "A fruit with a distinctive scent."
}
""".data(using: .utf8)!

let decoder = JSONDecoder()
let product = try decoder.decode(GroceryProduct.self, from: json)

print(product.name) // Prints "Durian"
```

[SDKs](#)[iOS 7.0+](#)[macOS 10.9+](#)[tvOS 9.0+](#)[watchOS 2.0+](#)[Xcode 9.0+](#)[Framework](#)[Foundation](#)[On This Page](#)[Declaration](#) ⌵[Overview](#) ⌵[Topics](#) ⌵[See Also](#) ⌵

Topics

First Steps

`init()`

Creates a new, reusable JSON decoder with the default formatting settings and decoding strategies.

`func decode<T>(T.Type, from: Data) -> T`

Returns a value of the type you specify, decoded from a JSON object.

Customizing Decoding

`var keyDecodingStrategy: JSONDecoder.KeyDecodingStrategy`

A value that determines how a type's coding keys are decoded from JSON keys.

`enum JSONDecoder.KeyDecodingStrategy`

The values that determine how a type's coding keys are decoded from JSON keys.

`var userInfo: [CodingUserInfoKey : Any]`

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

Decoding Dates

`var dateDecodingStrategy: JSONDecoder.DateDecodingStrategy`

The strategy used when decoding dates from part of a JSON object.

`enum JSONDecoder.DateDecodingStrategy`

The strategies available for formatting dates when decoding them from JSON.

Decoding Raw Data

`var dataDecodingStrategy: JSONDecoder.DataDecodingStrategy`

The strategy that a decoder uses to decode raw data.

`enum JSONDecoder.DataDecodingStrategy`

The strategies for decoding raw data.

Decoding Exceptional Numbers

```
var nonConformingFloatDecodingStrategy: JSONDecoder.NonConformingFloatDecodingStrategy
```

The strategy used by a decoder when it encounters exceptional floating-point values.

```
enum JSONDecoder.NonConformingFloatDecodingStrategy
```

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

See Also

JSON

 Using JSON with Custom Types

Demonstrates approaches for encoding and decoding different kinds of JSON in Swift.

```
class JSONEncoder
```

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

```
class JSONSerialization
```

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