

# JSON WITH PHP

[http://www.tutorialspoint.com/json/json\\_php\\_example.htm](http://www.tutorialspoint.com/json/json_php_example.htm)

Copyright © tutorialspoint.com

This chapter covers how to encode and decode JSON objects using PHP programming language. Let's start with preparing the environment to start our programming with PHP for JSON.

## Environment

As of PHP 5.2.0, the JSON extension is bundled and compiled into PHP by default.

## JSON Functions

Function	Libraries
json_encode	Returns the JSON representation of a value.
json_decode	Decodes a JSON string.
json_last_error	Returns the last error occurred.

## Encoding JSON in PHP *json\_encode*

PHP json\_encode function is used for encoding JSON in PHP. This function returns the JSON representation of a value on success or FALSE on failure.

## Syntax

```
string json_encode ( $value [, $options = 0 ] )
```

## Parameters

- **value** – The value being encoded. This function only works with UTF-8 encoded data.
- **options** – This optional value is a bitmask consisting of JSON\_HEX\_QUOT, JSON\_HEX\_TAG, JSON\_HEX\_AMP, JSON\_HEX\_APOS, JSON\_NUMERIC\_CHECK, JSON\_PRETTY\_PRINT, JSON\_UNESCAPED\_SLASHES, JSON\_FORCE\_OBJECT.

## Example

The following example shows how to convert an array into JSON with PHP –

```
<?php
    $arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
    echo json_encode($arr);
?>
```

While executing, this will produce the following result –

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```

The following example shows how the PHP objects can be converted into JSON –

```
<?php
class Emp {
    public $name = "";
    public $hobbies = "";
    public $birthdate = "";
}
```

```

$e = new Emp();
$e->name = "sachin";
$e->hobbies = "sports";
$e->birthdate = date('m/d/Y h:i:s a', "8/5/1974 12:20:03 p");
$e->birthdate = date('m/d/Y h:i:s a', strtotime("8/5/1974 12:20:03"));

echo json_encode($e);
?>

```

While executing, this will produce the following result –

```

{"name":"sachin","hobbies":"sports","birthdate":"08\05\1974 12:20:03 pm"}

```

## Decoding JSON in PHP *json\_decode*

PHP `json_decode` function is used for decoding JSON in PHP. This function returns the value decoded from json to appropriate PHP type.

### Syntax

```

mixed json_decode ($json [, $assoc = false [, $depth = 512 [, $options = 0 ]]])

```

### Parameters

- **json\_string** – It is an encoded string which must be UTF-8 encoded data.
- **assoc** – It is a boolean type parameter, when set to TRUE, returned objects will be converted into associative arrays.
- **depth** – It is an integer type parameter which specifies recursion depth
- **options** – It is an integer type bitmask of JSON decode, JSON\_BIGINT\_AS\_STRING is supported.

### Example

The following example shows how PHP can be used to decode JSON objects –

```

<?php
$json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

var_dump(json_decode($json));
var_dump(json_decode($json, true));
?>

```

While executing, it will produce the following result –

```

object(stdClass)#1 (5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
    ["e"] => int(5)
}

array(5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
    ["e"] => int(5)
}

```