### 

### **1. Codificar dados PHP para JSON**

Usamos a função json\_encode() para converter um array ou objeto PHP em uma string JSON.

#### **Exemplo 1: Codificando um array simples**

php

Copiar código

<?php

$data = [

"nome" => "João",

"idade" => 30,

"email" => "joao@example.com"

];

$json = json\_encode($data);

echo $json;

// Saída: {"nome":"João","idade":30,"email":"joao@example.com"}

?>

#### **Exemplo 2: Codificando um objeto PHP**

php

Copiar código

<?php

class Pessoa {

public $nome;

public $idade;

public function \_\_construct($nome, $idade) {

$this->nome = $nome;

$this->idade = $idade;

}

}

$pessoa = new Pessoa("Maria", 25);

$json = json\_encode($pessoa);

echo $json;

// Saída: {"nome":"Maria","idade":25}

?>

### **2. Decodificar JSON para um array ou objeto PHP**

Usamos a função json\_decode() para converter uma string JSON em um array ou objeto PHP.

#### **Exemplo 1: Decodificando uma string JSON em um array associativo**

php

Copiar código

<?php

$json = '{"nome":"João","idade":30,"email":"joao@example.com"}';

$data = json\_decode($json, true); // O segundo parâmetro 'true' retorna um array associativo

echo $data['nome']; // Saída: João

echo $data['idade']; // Saída: 30

?>

#### **Exemplo 2: Decodificando JSON em um objeto PHP**

php

Copiar código

<?php

$json = '{"nome":"João","idade":30,"email":"joao@example.com"}';

$data = json\_decode($json); // Retorna um objeto PHP

echo $data->nome; // Saída: João

echo $data->idade; // Saída: 30

?>

### **3. Tratamento de erros na codificação e decodificação de JSON**

É importante verificar erros após codificar ou decodificar dados JSON.

#### **Exemplo de verificação de erro após json\_encode()**

php

Copiar código

<?php

$data = [

"nome" => "João",

"idade" => 30,

"email" => "joao@example.com"

];

$json = json\_encode($data);

if (json\_last\_error() !== JSON\_ERROR\_NONE) {

echo 'Erro na codificação JSON: ' . json\_last\_error\_msg();

} else {

echo $json;

}

?>

#### **Exemplo de verificação de erro após json\_decode()**

php

Copiar código

<?php

$json = '{"nome":"João","idade":30,"email":}'; // JSON inválido

$data = json\_decode($json);

if (json\_last\_error() !== JSON\_ERROR\_NONE) {

echo 'Erro na decodificação JSON: ' . json\_last\_error\_msg();

} else {

echo $data->nome;

}

?>

### **4. Codificar JSON com opções adicionais**

Você pode passar opções para json\_encode() para melhorar a legibilidade ou controlar como o JSON é gerado.

#### **Exemplo com opções de formatação**

php

Copiar código

<?php

$data = [

"nome" => "João",

"idade" => 30,

"email" => "joao@example.com"

];

$json = json\_encode($data, JSON\_PRETTY\_PRINT); // Formatação legível (indentação)

echo $json;

?>

Saída:

json

Copiar código

{

"nome": "João",

"idade": 30,

"email": "joao@example.com"

}

### **5. Codificar e decodificar grandes objetos ou arrays JSON com UTF-8**

O PHP tem suporte nativo para UTF-8, mas ao trabalhar com JSON, é bom garantir que o encoding esteja correto.

#### **Exemplo com UTF-8:**

php

Copiar código

<?php

$data = [

"mensagem" => "Olá, mundo! 🌎"

];

$json = json\_encode($data, JSON\_UNESCAPED\_UNICODE); // Evita a conversão de caracteres Unicode

echo $json;

// Saída: {"mensagem":"Olá, mundo! 🌎"}

?>

### **6. Armazenar e ler JSON de um arquivo**

Você pode salvar dados JSON em um arquivo e também carregá-los de volta.

#### **Exemplo de escrita em um arquivo JSON:**

php

Copiar código

<?php

$data = [

"nome" => "João",

"idade" => 30,

"email" => "joao@example.com"

];

$json = json\_encode($data, JSON\_PRETTY\_PRINT);

file\_put\_contents('dados.json', $json);

?>

#### **Exemplo de leitura de um arquivo JSON:**

php

Copiar código

<?php

$json = file\_get\_contents('dados.json');

$data = json\_decode($json, true);

echo $data['nome']; // Saída: João

echo $data['idade']; // Saída: 30

?>

Esses são os exemplos básicos para manipulação de JSON em PHP. Eles cobrem a codificação e decodificação, bem como o armazenamento e a leitura de arquivos JSON.

quando temos um exemplo de arquivo com json com o seguinte conteúdo.

{"info":{"count":20},

"data":[

{"id\_escolaridade":20,"descricao":"Outros (Sem Informa\u00e7\u00e3o)","ativo":false},

{"id\_escolaridade":99,"descricao":"N\u00e3o Informada","ativo":true},

{"id\_escolaridade":1,"descricao":"Analfabeto","ativo":true},

{"id\_escolaridade":2,"descricao":"Alfabetizado sem cursos regulares","ativo":true},

{"id\_escolaridade":3,"descricao":"Primeiro grau incompleto ate a 4a serie","ativo":false},

{"id\_escolaridade":4,"descricao":"4a.S\u00e9rie do primeiro grau completa","ativo":true},

{"id\_escolaridade":6,"descricao":"Ensino fundamental","ativo":true},

{"id\_escolaridade":7,"descricao":"Segundo grau incompleto","ativo":false},

{"id\_escolaridade":8,"descricao":"Ensino medio","ativo":true},

{"id\_escolaridade":9,"descricao":"Superior incompleto","ativo":false},

{"id\_escolaridade":10,"descricao":"Ensino superior","ativo":true},

{"id\_escolaridade":5,"descricao":"Ensino fundamental incompleto","ativo":false},

{"id\_escolaridade":11,"descricao":"Aperfeicoamento","ativo":true},

{"id\_escolaridade":12,"descricao":"Especializacao","ativo":true},

{"id\_escolaridade":13,"descricao":"Mestrado","ativo":true},

{"id\_escolaridade":14,"descricao":"Doutorado","ativo":true},

{"id\_escolaridade":15,"descricao":"Pos graduacao","ativo":false},

{"id\_escolaridade":16,"descricao":"PHD","ativo":false},

{"id\_escolaridade":18,"descricao":"Pos doutorado","ativo":true},

{"id\_escolaridade":19,"descricao":"Livre docencia","ativo":true}]}

Para ler e processar o conteúdo JSON do arquivo descrito a cima em PHP, você pode seguir os seguintes passos:

### **Passos:**

1. **Ler o conteúdo do arquivo JSON**.
2. **Decodificar o JSON** em um array ou objeto PHP usando json\_decode().
3. **Acessar os dados** a partir da estrutura decodificada.

### **Exemplo completo:**

Vamos supor que o arquivo com os dados JSON se chame dados.json. Aqui está um exemplo de código PHP para ler e acessar essas informações:

#### **Passo 1: Ler o conteúdo do arquivo JSON**

php

Copiar código

<?php

// Ler o conteúdo do arquivo JSON

$json = file\_get\_contents('dados.json');

// Decodificar o JSON para um array associativo

$data = json\_decode($json, true);

// Verificar se houve algum erro na decodificação

if (json\_last\_error() !== JSON\_ERROR\_NONE) {

echo 'Erro na decodificação JSON: ' . json\_last\_error\_msg();

exit;

}

// Agora podemos acessar os dados no array $data

?>

#### **Passo 2: Acessar os dados**

Após a decodificação, você pode acessar as informações do JSON de acordo com a estrutura. No caso do JSON que você forneceu, ele tem uma estrutura de chave info e uma chave data, onde data contém um array com informações de escolaridade.

php

Copiar código

<?php

// Exemplo de como acessar os dados decodificados

// A chave "info" contém o "count"

$count = $data['info']['count'];

echo "Contagem: " . $count . "<br>";

// A chave "data" contém o array de informações sobre escolaridade

foreach ($data['data'] as $item) {

$id\_escolaridade = $item['id\_escolaridade'];

$descricao = $item['descricao'];

$ativo = $item['ativo'] ? 'Ativo' : 'Inativo'; // Condicional para mostrar "Ativo" ou "Inativo"

echo "ID Escolaridade: $id\_escolaridade - Descrição: $descricao - Status: $ativo<br>";

}

?>

### **Explicação:**

1. **file\_get\_contents('dados.json')**: Lê o conteúdo do arquivo JSON.
2. **json\_decode($json, true)**: Decodifica o conteúdo JSON em um array associativo. O parâmetro true indica que queremos um array associativo em vez de um objeto.
3. **Acessando os dados**:
   * data['info']['count']: Acessa o valor de count.
   * data['data']: Acessa o array de itens dentro de data. Usamos um foreach para percorrer os itens da lista e acessar as chaves id\_escolaridade, descricao e ativo.

#### **Saída esperada (para o seu JSON de exemplo):**

yaml

Copiar código

Contagem: 20

ID Escolaridade: 20 - Descrição: Outros (Sem Informação) - Status: Inativo

ID Escolaridade: 99 - Descrição: Não Informada - Status: Ativo

ID Escolaridade: 1 - Descrição: Analfabeto - Status: Ativo

ID Escolaridade: 2 - Descrição: Alfabetizado sem cursos regulares - Status: Ativo

ID Escolaridade: 3 - Descrição: Primeiro grau incompleto ate a 4a serie - Status: Inativo

ID Escolaridade: 4 - Descrição: 4a.Série do primeiro grau completa - Status: Ativo

ID Escolaridade: 6 - Descrição: Ensino fundamental - Status: Ativo

ID Escolaridade: 7 - Descrição: Segundo grau incompleto - Status: Inativo

ID Escolaridade: 8 - Descrição: Ensino medio - Status: Ativo

ID Escolaridade: 9 - Descrição: Superior incompleto - Status: Inativo

ID Escolaridade: 10 - Descrição: Ensino superior - Status: Ativo

ID Escolaridade: 5 - Descrição: Ensino fundamental incompleto - Status: Inativo

ID Escolaridade: 11 - Descrição: Aperfeiçoamento - Status: Ativo

ID Escolaridade: 12 - Descrição: Especializacao - Status: Ativo

ID Escolaridade: 13 - Descrição: Mestrado - Status: Ativo

ID Escolaridade: 14 - Descrição: Doutorado - Status: Ativo

ID Escolaridade: 15 - Descrição: Pos graduacao - Status: Inativo

ID Escolaridade: 16 - Descrição: PHD - Status: Inativo

ID Escolaridade: 18 - Descrição: Pos doutorado - Status: Ativo

ID Escolaridade: 19 - Descrição: Livre docencia - Status: Ativo

### **Observações:**

* **Codificação de caracteres**: No seu JSON, existem caracteres especiais como \u00e7 (que é o "ç"). O PHP decodifica corretamente esses caracteres em UTF-8, portanto, não é necessário fazer nada além de usar json\_decode() e exibir o resultado com echo.
* **Validação de erros**: O código usa json\_last\_error() para verificar se houve algum erro durante a decodificação, o que é uma boa prática quando você trabalha com dados JSON em PHP.