

Relatório para Converter código PHP

Proprietário



Victor Hugo Zeschau

Desenvolvi duas funções modificadas para aprimorar o script de registro de pessoas, com a missão de alterar o método de armazenamento de dados. A orientação inicial era transicionar da gravação em formato JSON para o formato XML.

Levando isso em consideração, efetuei ajustes nas funções responsáveis por qualquer modificação no arquivo JSON. Na função de salvamento, incluí uma linha de código para invocar a função `salvar_xml`. Essa nova função recebe como parâmetros a variável que contém os dados originalmente em JSON e uma constante que especifica o nome do arquivo XML e o local onde ele será armazenado.

```
function salvar()
{
    $json = NULL;
    $pessoa = tela2array();

    $json = ler_json(ARQUIVO_JSON);

    if ($json == NULL) {
        $json = array();
    }

    array_push($json, $pessoa);

    salvar_json(json_encode($json), ARQUIVO_JSON); // Salvar em JSON
    salvar_xml($json, ARQUIVO_XML); // Salvar em XML
    header("location:" . DESTINO);
}
```

A Definição da constante no topo do código

```
0
1 /* definição de constantes */
2 define("DESTINO", "index.php");
3 define("ARQUIVO_JSON", "pessoa.json");
4 define("ARQUIVO_XML", "pessoa.xml");
5
```

Desenvolvi a função `salvar_xml` com o propósito de converter dados em formato de array (JSON) para XML e, posteriormente, armazená-los em um arquivo. Essa rotina inicia criando um elemento XML raiz chamado `<peessoas>` usando a classe `SimpleXMLElement`.

Em seguida, chama a função `array_to_xml`, que é responsável por transformar recursivamente o array fornecido em uma estrutura XML aninhada sob o elemento raiz.

Finalmente, a função `asXML` é utilizada para gravar essa estrutura XML no caminho do arquivo especificado. Esse processo simplifica a migração dos dados de um formato para outro, garantindo que as informações sejam armazenadas de maneira estruturada e acessível.

```
function salvar_xml($data, $arquivo) {  
    $xml = new SimpleXMLElement('<?xml version="1.0"?><peessoas></peessoas>');  
    array_to_xml($data, $xml);  
    $xml->asXML($arquivo);  
}
```

Desenvolvi uma função denominada `array_to_xml`, cujo principal objetivo é converter arrays e objetos do PHP para o formato XML. Essa necessidade surgiu da importância de ter uma maneira eficaz e segura de armazenar dados, minimizando o risco de erros durante o processo de gravação.

Adicionalmente, implementei uma solução para tratar caracteres especiais, utilizando a função `htmlspecialchars`. Esse cuidado garante a integridade e a correta formatação do XML, prevenindo erros que poderiam comprometer a leitura dos dados. Essa abordagem foi inspirada por uma solução encontrada em um fórum do GitHub, que se mostrou essencial para aprimorar o processo de criação do arquivo XML, eliminando as mensagens de erro anteriormente enfrentadas. Este ajuste reflete o compromisso com a qualidade e a precisão dos dados armazenados, aspectos cruciais para o sucesso do projeto universitário.

```

function array_to_xml($data, &$xml_data) {
    foreach ($data as $key => $value) {
        if (is_object($value)) {
            $value = (array)$value;
        }
        if (is_array($value)) {
            if (is_numeric($key)) {
                $key = 'item' . $key;
            }
            $subnode = $xml_data->addChild($key);
            array_to_xml($value, $subnode);
        } else {
            $xml_data->addChild("$key", htmlspecialchars((string)$value));
        }
    }
}

```

Para complementar, adaptei as funções responsáveis pela exclusão e alteração de registros na listagem de pessoas, permitindo que utilizassem a função `salvar_xml`. Dessa forma, qualquer modificação realizada nos dados JSON reflete-se automaticamente no arquivo XML correspondente. Esse procedimento garante que a aplicação mantenha a manipulação dos dados em JSON, ao mesmo tempo em que realiza o armazenamento paralelo em XML, assegurando a consistência e a integridade dos dados em ambos os formatos.

```

function excluir()
{
    $id = isset($_GET['id']) ? $_GET['id'] : "";
    $json = ler_json(ARQUIVO_JSON);
    if ($json == null)
        $json = array();

    $novo = array();
    for ($x = 0; $x < count($json); $x++) {
        var_dump($json[$x]);
        if ($json[$x]->id != $id)
            array_push($novo, $json[$x]);
    }
    salvar_json(json_encode($novo), ARQUIVO_JSON);
    salvar_xml($json, ARQUIVO_XML);
    header("location:" . DESTINO);
}

```

```
function alterar()
{
    $novo = tela2array();

    $json = ler_json(ARQUIVO_JSON);

    for ($x = 0; $x < count($json); $x++) {
        if ($json[$x]→id == $novo['id']) {
            array2json($novo, $json[$x]);
        }
    }

    salvar_json(json_encode($json), ARQUIVO_JSON);
    salvar_xml($json, ARQUIVO_XML);
    header("location:" . DESTINO);
}
```

