XSD2PHP

Emanuele Davide

5BII

22/02/2018

[**Composer**](#_30j0zll) **1**

[**Download dello strumento e configurazione**](#_1fob9te) **1**

[**Comando**](#_3znysh7) **2**

## Composer

Per eseguire correttamente la creazione di classi PHP a partire da un XSD, c’è bisogno di un gestore di pacchetti a livello applicativo per il linguaggio PHP, chiamato Composer. Questo strumento fornisce un formato standard per la gestione delle dipendenze del software PHP e delle librerie richieste.

Composer funziona leggendo l'elenco dei pacchetti da installare per ogni singolo progetto da un file chiamato composer.json. Gestire le dipendenze è fondamentale quando, per funzionare, un pacchetto dipende da un altro.

## Download dello strumento e configurazione

Da GitHub, al seguente link <https://github.com/goetas-webservices/xsd2php> , è possibile scaricare una cartella compressa, con tutti i file necessari e il comando da introdurre in shell per la creazione di classi PHP.

Fra questi, vi è un file chiamato “composer.json”, il quale va configurato secondo le istruzioni presenti sul sito, aggiungendo le seguenti dipendenze direttamente nel file :

*"require": {  
 ..  
 "goetas-webservices/xsd2php-runtime":"^0.2.2",  
 ..  
 },  
 "require-dev": {  
 ..  
 "goetas-webservices/xsd2php":"^0.3",*

*..  
 },*

In dettaglio, il frammento di “codice” da aggiungere permette l’installazione del file eseguibile

“xsd2php”.

Troveremo inoltre un altro file denominato “config.yml”, il quale va configurato a seconda delle preferenze dell’utente. In questo file si decide la cartella di destinazione delle classi PHP, e i namespace da usare.

## Comando

Il comando per generare le classi direttamente da shell, dopo aver fatto le opportune configurazioni è il seguente : vendor/bin/xsd2php convert config.yml <percorsoxsd/nomexsd\*.xsd>.

A questo punto, il file PHP con le classi generate dovrebbe trovarsi nella cartella di destinazione definita in precedenza.

## Procedimento

1. Installazione di XAMPP e avvio di Apache ;
2. Installazione di Composer ;
3. Creazione di una cartella in Documenti/, di nome composer contenente il file composer.json editato in questo modo :

*{*

*"require": {*

*"goetas-webservices/xsd2php-runtime":"^0.2.2"*

*},*

*"require-dev": {*

*"goetas-webservices/xsd2php":"^0.3"*

*}*

*}*

4. Esecuzione, a linea di comando, da dentro la cartella composer/ : “*composer install*” ;

5. Creazione automatica di una cartella vendor/ ;

6. Allo stesso livello di vendor/, come “fratello”, creazione di una cartella orienteering/ ;

7. Inserimento del file XSD da usare e del file config.yml all’interno di orienteering/ ;

8. Editazione del file config.yml nel seguente modo :

*xsd2php:*

*namespaces:*

*' ': 'TestNs\MyApp'*

*destinations\_php:*

*'TestNs\MyApp': soap\src # on Windows*

*destinations\_jms:*

*'TestNs\MyApp': soap\metadata # on Windows*

9. Avvio del seguente comando da shell :

*..\vendor\bin\xsd2php convert .\config.yml .\CastelliXSD2.txt*

Le classi saranno generate automaticamente nella cartella orienteering/.