PHP 10권 – JSON

**08. JSON 파일**

====================

컴포저의 패키지들은 JSON형태의 데이터 구조 파일로 되어 있습니다.

또한 컴포저 및 페키지 설치를 하게 되면 두개의 JSON 파일이 생성이 됩니다. composer.json 과 composer.lock 파일입니다.

**08.1 기본 설정**

컴포저를 초기화 하게 되면 composer.json 파일이 생성이 됩니다. 컴포저는 json 형태의 파일내용으로 페키지를 관리하게 됩니다. 페지키는 여러 가지의 내용을 담고 있습니다.

**08.1.1 Name**

스페이스를 포함하지 않습니다.

"name": "infoh/composer",

**08.1.2 Description**

설명문구를 추가할 수 있습니다.

"description":"jiny composer website study",

**08.1.3 authors**

저자부분은 배열객체 형태의 값으로 가지고 있습니다.

"authors": [

{

"name": "hojinlee",

"email": "infohojin@gmail.com"

},

{

"name": "hojinlee",

"email": "infohojin@gmail.com"

}

]

만일 여러명의 개발자로 저자명을 등록을 하고 싶다면은 배열 형태로 추가 하시면 됩니다.

**08.1.4 homepage**

패키지 배포와 관련된 홈페이지의 주소를 같이 작성할 수 있습니다.

"homepage": "https://hojin.io",

**08.1.5 require**

실제적인 페키지들의 목록이 나열되어 있습니다. 또한, 패키지가 다른 패키지와 의존관계가 있는 경우 같이 적어 주시면 됩니다.

다음은 illuminate/support 패키지의 require부분입니다.

"require": {

"php": "^7.1.3",

"ext-mbstring": "\*",

"doctrine/inflector": "~1.1",

"illuminate/contracts": "5.6.\*",

"nesbot/carbon": "^1.24.1"

}

패키지의 PHP 적용 버전을 지정할 수 있습니다. 또한 의존 패키지들 목록과 버전을 같이 지정을 합니다.

**08.1.6 require-dev**

require-dev는 Require와 유사하지만 차이점은 개발과 실제서버의 페키지를 분리하는 것입니다. require-dev는 개발모드일 때 같이 설치되는 페키지들을 나열합니다.

**08.2 오토로드 설정**

composer.json 파일에는 패키지들의 의존성과 오토로드 처리를 위한 설정들을 관리합니다.

**08.2.1 Autoload**

오토로드 필드를 추가할 수 있습니다. 오토로드 필드는 몇 개의 키를 같이 작성하여 사용을 합니다. 약간의 키값이 있습니다.

예를 들어

"autoload":{

"psr-4":{

},

"files":[],

"classmap":[],

},

"autoload-dev":{

}

**08.2.2 Files**

오토로드 필드의 또다른 항목으로 files가 있습니다. 이 항목은 배열로 구성이 되어 있습니다.

오토로드에서 PSR-4와 차이점은 무었일까요? PSR-4와 레거시 프로젝트간의 분리가 필요한 경우 있습니다.

다음과 같이 일반적인 PHP 파일을 하나 생성을 합니다.

App\function.php

<?php

function bindString($str1, $str2) {

return $str1 . $str2;

}

?>

생성한 파일의 정보를 composer.json 의 오토로드 필드 항목에 추가합니다.

"files": [

"app/function.php"

]

Dumpautoload를 실행하여 컴포저를 갱신합니다.

func.php

<?php

require "vendor/autoload.php";

$strings = bindString("hello ", "world");

echo $strings;

**결과**

hello world

**08.2.3 ClassMap**

오토로드의 3번째 기능으로 classmap 기능이 있습니다. 클래스맵은 PSR4와 매우 유사합니다.

서브폴더의 처리방식에 있어서 차이점이 있습니다.

하지만 클래스맵은 재귀적으로 클래스 디렉토리를

"classmap": [

"app/classes"

]

와 같이 추가하고 dumpautoload를 실행합니다.

<?php

require "vendor/autoload.php";

$class1 = new Class1();

$class2 = new Class2();

$class3 = new Class3();

클래스맵은 /vendor/composer/autoload\_classmap.php 파일을 생성하여 관리하게 됩니다.

**08.4 lock파일**

컴포저는 composer.lock 파일을 하나 더 가지게 됩니다.