

作って理解するComposer <クイックコース>
を本当に作れるのか理解する会

今日の目的

フィードバックをください！！！！

- (connpass作った時には色々と書きましたが)
主に、「やってみて、作ってみた内容や難易度はどうだった？」が欲しいです
 - 難しかった所
 - おもしろい！！感動した！！っていう所
 - もっと知りたい・ちょっと物足りないなーってなった所
- たぶん時間に余裕がある想定なのですが、
コンテンツを増やすとしたら何処ら辺が良さそうですかね？？も
最後に聞きます💧

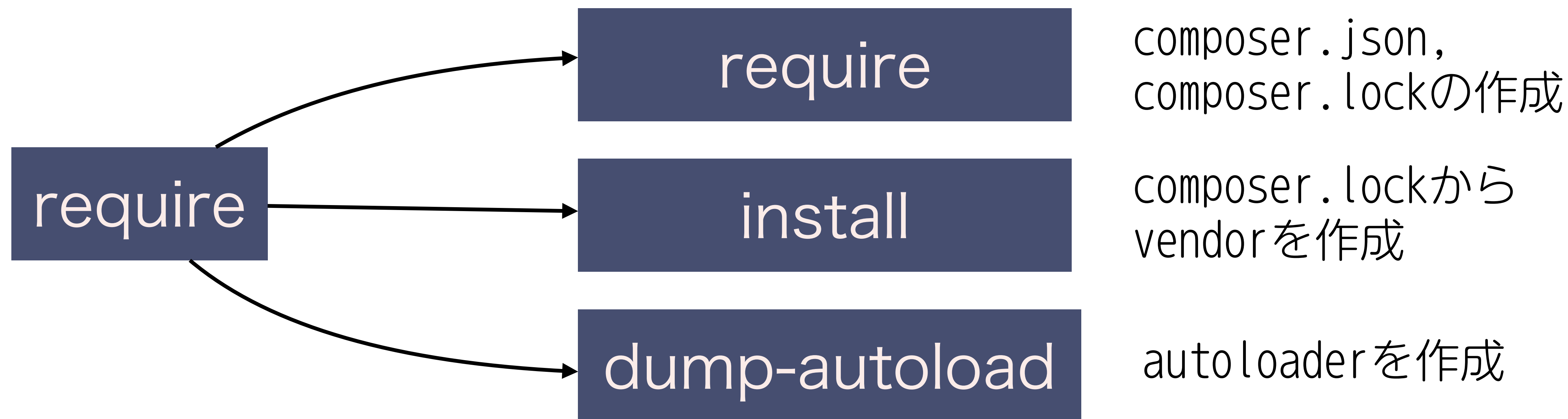
扱うこと

- 最終的に `composer require` を動かします！

```
d9a7714c211c:/tmp/tmp2# php /app/require/require.php psr/log aura/cli psr/container psr/clock
d9a7714c211c:/tmp/tmp2# cat composer.json
{
    "require": {
        "aura/cli": "2.2.0",
        "psr/clock": "1.0.0",
        "psr/container": "2.0.2",
        "psr/log": "3.0.2"
    }
}
```

扱うこと

- 大まかに3つの要素に分割して、個別にやっていきましょう💪
 - ※ 実際の所、本物も内部ではcommand pattern(かな?)で分割されています



まずはお手元に環境を・・・

- GitHubにレポジトリがあるので、cloneしてください
 - URL: **TBD**
- Dockerを利用します
 - bash, vimが入ったalpineです
 - Xdebugも入れてあります
- `docker compose run -it app bash` で作業してもらえると楽かも
 - ※ ご自身の責任の範囲で、お手元のPHPで構いません！

```
FROM composer:2.8.3
RUN apk add --no-cache linux-headers autoconf build-base
RUN pecl install xdebug-3.4.0
RUN docker-php-ext-enable xdebug
RUN apk add bash vim
```

まずはお手元に環境を・・

- GitHub APIも使うので、tokenを用意してください
 - <https://github.com/settings/tokens/new>
 - TBD: scopeどれだっけ
- できたらrootの ``.env`` に入れてください
 - docker compose経由で環境ファイルを渡しますが、アプリケーションがdotenvを用意しているわけではないので
自前環境の場合は環境変数 ``GITHUB_OAUTH_TOKEN`` をセット

① install

ゴール

- 予め(本物の！)composerで用意した、composer.lockを利用します
 - src: `app/install/composer.lock`
- コレを読み取って、
指定されたパッケージ・バージョンをローカルに配置するところまで
- ※ 本物とは違い、composer.jsonにある諸々の要件のチェックは端折ります

composer.lockのどこを見ればいいの？

- 今回使うのは ``$.packages[] . {name|dist.url}``

```
content-hash : 6e44e2c20b802abf2efe054d527a915d ,  
"packages": [  
    {  
        "name": "psr/container",  
        "version": "2.0.2",  
        "source": {  
            "type": "git",  
            "url": "https://github.com/php-fig/container.git",  
            "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963"  
        },  
        "dist": {  
            "type": "zip",  
            "url": "https://api.github.com/repos/php-fig/container/zipball/c71ecc56dfe541dbd90c5360474fbc405f8d5963",  
            "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963",  
            "shasum": ""  
        },  
    }  
]
```

composer.lockのどこを見ればいいの？

- 今回使うのは`\$.packages []. {name|dist.url}`

```
content-hash : 6e44e2c20b002a6f2e1e054d527a915d ,
"packages": [
  {
    "name": "psr/container",
    "version": "2.0.2",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/container.git",
      "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/container/zipball/c71ecc56dfe541dbd90c5360474fbc405f8d5963",
      "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963",
      "shasum": ""
    }
  },
  {
    "name": "psr/http-client",
    "version": "1.0.1",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/http-client.git",
      "reference": "2dfb8f6a5f3ad5aa7aa31c668c864009a6e73d16"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/http-client/zipball/2dfb8f6a5f3ad5aa7aa31c668c864009a6e73d16",
      "reference": "2dfb8f6a5f3ad5aa7aa31c668c864009a6e73d16",
      "shasum": ""
    }
  },
  {
    "name": "psr/http-factory",
    "version": "1.0.1",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/http-factory.git",
      "reference": "1b27464003ad8563b5551d8f68f5ff71d90771d"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/http-factory/zipball/1b27464003ad8563b5551d8f68f5ff71d90771d",
      "reference": "1b27464003ad8563b5551d8f68f5ff71d90771d",
      "shasum": ""
    }
  },
  {
    "name": "psr/http-message",
    "version": "1.0.1",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/http-message.git",
      "reference": "414906b7c57f4e6c06208c9a9c914d3961b5be7e"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/http-message/zipball/414906b7c57f4e6c06208c9a9c914d3961b5be7e",
      "reference": "414906b7c57f4e6c06208c9a9c914d3961b5be7e",
      "shasum": ""
    }
  },
  {
    "name": "psr/log",
    "version": "1.1.3",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/log.git",
      "reference": "3f6294e3f0e42ac148541e7013eb0fe9cce23949"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/log/zipball/3f6294e3f0e42ac148541e7013eb0fe9cce23949",
      "reference": "3f6294e3f0e42ac148541e7013eb0fe9cce23949",
      "shasum": ""
    }
  },
  {
    "name": "symfony/console",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/console.git",
      "reference": "90e270317e017130f7b3376c8cf6fd5f5ac62c72"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/console/zipball/90e270317e017130f7b3376c8cf6fd5f5ac62c72",
      "reference": "90e270317e017130f7b3376c8cf6fd5f5ac62c72",
      "shasum": ""
    }
  },
  {
    "name": "symfony/css-selector",
    "version": "v5.3.4",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/css-selector.git",
      "reference": "884e664ad889394b641326e1b38574732149cfa"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/css-selector/zipball/884e664ad889394b641326e1b38574732149cfa",
      "reference": "884e664ad889394b641326e1b38574732149cfa",
      "shasum": ""
    }
  },
  {
    "name": "symfony/debug",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/debug.git",
      "reference": "3f084e76f52453e92702722405c1a1ac2f3e76a0"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/debug/zipball/3f084e76f52453e92702722405c1a1ac2f3e76a0",
      "reference": "3f084e76f52453e92702722405c1a1ac2f3e76a0",
      "shasum": ""
    }
  },
  {
    "name": "symfony/dependency-injection",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/dependency-injection.git",
      "reference": "50e2546407304e885078b79918e8240cd00c9e6c"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/dependency-injection/zipball/50e2546407304e885078b79918e8240cd00c9e6c",
      "reference": "50e2546407304e885078b79918e8240cd00c9e6c",
      "shasum": ""
    }
  },
  {
    "name": "symfony/error-handler",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/error-handler.git",
      "reference": "3bc067ba6279e9f2c52e7bd65462e3e428191658"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/error-handler/zipball/3bc067ba6279e9f2c52e7bd65462e3e428191658",
      "reference": "3bc067ba6279e9f2c52e7bd65462e3e428191658",
      "shasum": ""
    }
  },
  {
    "name": "symfony/event-dispatcher",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/event-dispatcher.git",
      "reference": "770b0752f7c6edd4636bb19812ed38f802c31cbe"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/event-dispatcher/zipball/770b0752f7c6edd4636bb19812ed38f802c31cbe",
      "reference": "770b0752f7c6edd4636bb19812ed38f802c31cbe",
      "shasum": ""
    }
  },
  {
    "name": "symfony/event-dispatcher-contracts",
    "version": "v2.5.0",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/event-dispatcher-contracts.git",
      "reference": "e632b147f824c1d4e72005f2ea029e5984d0e083"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/event-dispatcher-contracts/zipball/e632b147f824c1d4e72005f2ea029e5984d0e083",
      "reference": "e632b147f824c1d4e72005f2ea029e5984d0e083",
      "shasum": ""
    }
  },
  {
    "name": "symfony/filesystem",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/filesystem.git",
      "reference": "82e7b63c4c26a64571514b34d4d18e3649b91da"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/filesystem/zipball/82e7b63c4c26a64571514b34d4d18e3649b91da",
      "reference": "82e7b63c4c26a64571514b34d4d18e3649b91da",
      "shasum": ""
    }
  },
  {
    "name": "symfony/http-client-contracts",
    "version": "v2.5.0",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/http-client-contracts.git",
      "reference": "121437cf0171d172266cc6eb0878740042c34cd6"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/http-client-contracts/zipball/121437cf0171d172266cc6eb0878740042c34cd6",
      "reference": "121437cf0171d172266cc6eb0878740042c34cd6",
      "shasum": ""
    }
  },
  {
    "name": "symfony/http-foundation",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/http-foundation.git",
      "reference": "e58d7f77ba7809652d24221a169bf871905dd935"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/http-foundation/zipball/e58d7f77ba7809652d24221a169bf871905dd935",
      "reference": "e58d7f77ba7809652d24221a169bf871905dd935",
      "shasum": ""
    }
  },
  {
    "name": "symfony/http-kernel",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/http-kernel.git",
      "reference": "7e0b43f039ec6362b8b2c606dd0673ea67dcb4d2"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/http-kernel/zipball/7e0b43f039ec6362b8b2c606dd0673ea67dcb4d2",
      "reference": "7e0b43f039ec6362b8b2c606dd0673ea67dcb4d2",
      "shasum": ""
    }
  },
  {
    "name": "symfony/mailer",
    "version": "v5.3.7",
    "source": {
      "type": "git",
      "url": "https://github.com/symfony/mailer.git",
      "reference": "30c243a12f0cf2627b7300c2c2b88b968218c54"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/symfony/mailer/zipball/30c243a12f0cf2627b7300c2c2b88b968218c54",
      "reference": "30c243a12f0cf2627b7300c2c2b88b968218c54",
      "shasum": ""
    }
  },
  {
    "name": "symfony/mime",
    "version
```

composer.lockのどこを見ればいいの？

- 今回使うのは ``$.packages[] . {name|dist.url}``

```

content-hash : 6e44e2c20b602a6f2e1e054d527a915d ,
"packages": [
  {
    "name": "psr/container",
    "version": "2.0.2",
    "source": {
      "type": "git",
      "url": "https://github.com/php-fig/container.git",
      "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963"
    },
    "dist": {
      "type": "zip",
      "url": "https://api.github.com/repos/php-fig/container/zipball/c71ecc56dfe541dbd90c5360474fbc405f8d5963",
      "reference": "c71ecc56dfe541dbd90c5360474fbc405f8d5963",
      "shasum": ""
    },
  },

```

distファイル (GitHub API経由)

大まかな流れ

1. .lockファイルを読み込んで
2. その中のpackagesごとにループ処理
3. distファイルを落としてきて
4. zipファイルを展開して
5. パッケージ名に従って、最終的なディレクトリに配置

lockファイルの読み込み

```
// $lockJsonPath = YOUR_LOCK_FILE_PATH  
$lockJson = file_get_contents($lockJsonPath);  
$lock = json_decode($lockJson);
```

その中のpackagesごとにループ処理

```
foreach ($lock->packages as $package) {  
    var_dump($package->name);  
}
```


distファイルを落としてきて

```
foreach ($lock->packages as $package) {  
    $ch = curl_init($package->dist->url);  
    curl_setopt_array($ch, [  
        \CURLOPT_RETURNTRANSFER => true,  
        \CURLOPT_FOLLOWLOCATION => TRUE,  
        \CURLOPT_HTTPHEADER => [  
            "Authorization: Bearer " . getenv('GITHUB_OAUTH_TOKEN'),  
            "User-Agent: MyPHPApp/1.0",  
        ],  
    ]);  
    $dist = curl_exec($ch);  
}
```

zipの展開

```
foreach ($lock->packages as $package) {  
    // さっきの続き  
    // create zip stream  
    $fp = tmpfile();  
    fwrite($fp, $dist);  
    $zip = new ZipArchive();  
    $zip->open(stream_get_meta_data($fp)['uri']);  
  
    // extract (tmp)  
    for ($i = 0; $i < $zip->numFiles; $i++) {  
        $zip->extractTo($tmpDir, $zip->getNameIndex($i));  
    }  
}
```

NOTE:

ZIPファイルの中身(構造)について
簡単な解説をした方がよい

パッケージ名に従って、最終的なディレクトリに配置

```
foreach ($lock->packages as $package) {  
    // さっきの続き  
  
    // install  
    $target = "{$workingDir}/vendor/{$package->name}";  
    mkdir($target, 0777, true);  
    rename "{$tmpDir}/{$zip->getNameIndex(0)}", $target);  
  
    // tear down  
    $zip->close(); // Close the zip archive  
    unset($fp);  
}
```

② dump-autoload

ゴール

- PSR-4, `files`(イーガーロード)に対応させます
- `vendor`以下にある`composer.json`を読み取って行います

大まかな流れ

1. vendor以下の各パッケージのcomposer.jsonを読み取って
2. files, psr-4のクラスマップを取得して
3. 取得した結果を一旦ファイルに吐き出して
4. オートローダーの雛形をコピーする

オートローダー

```
require __DIR__ . '/autoload_files.php';
function loadClass(string $class) {
    $psr4Map = require __DIR__ . '/autoload_classmap_psr4.php';
    $elements = explode('\\', $class);
    while ($elements) {
        $search = implode('\\', $elements) . '\\';
        if (array_key_exists($search, $psr4Map)) {
            $packageRootPath = $psr4Map[$search];
            $sub = str_replace($search, '', $class);
            $subPath = str_replace('\\', '/', $sub);
            $filePath = $packageRootPath . $subPath . '.php';
            require realpath($filePath);
        }
        array_pop($elements);
    }
}

spl_autoload_register('loadClass');
```

vendor以下の各パッケージのcomposer.jsonを読み取って

```
// $vendorDir = YOUR_VENDOR_DIRECTORY_PATH
$packageFiles = glob("{ $vendorDir }/*/*/composer.json");

$psr4ClassMaps = $autoloadFiles = [];
foreach ($packageFiles as $packageFile) {
    //
}
```

vendor以下の各パッケージのcomposer.jsonを読み取って

```
// foreach ($packageFiles as $packageFile) {  
$package = json_decode(file_get_contents($packageFile));  
$autoload = $package->autoload ?? null;  
if (!$autoload) {  
    continue;  
}  
$packageDir = dirname($packageFile);  
$packagePsr4ClassMaps = $autoload->{'psr-4'} ?? [];  
foreach ($packagePsr4ClassMaps as $namespace => $dirName) {  
    $psr4ClassMaps[$namespace] = "{$packageDir}/{$dirName}";  
}  
  
$packageAutoloadFiles = $autoload->files ?? [];  
foreach ($packageAutoloadFiles as $packageAutoloadFile) {  
    $autoloadFiles[] = "{$packageDir}/{$packageAutoloadFile}";  
}
```

取得した結果を一旦ファイルに吐き出して

```
file_put_contents(
    "{$vendorDir}/autoload_classmap_psr4.php",
    '<?php return ' . var_export($psr4ClassMaps, true) . ';'
);

file_put_contents(
    "{$vendorDir}/autoload_files.php",
    '<?php return ' . var_export($autoloadFiles, true) . ';'
);

copy(__DIR__ . '/autoload.php', "{$vendorDir}/autoload.php");
```