

Web-программирование

Месон №8

(1 часть)

Каюсов С.В.

09.03.01.

УБМ-3, 3 курс

Тест №8

Местовый контроль „Основы создания библиотек Composer“

Часть 1. Основы работы с composer

1-15. Вопросы с выбором

1. б. Инструмент для управления зависимостями в PHP-проектах;

2. а. composer.json

3. а. composer init

4. а. Зависимости указываются вручную в файле composer.json, затем Composer автоматически загружает их с помощью команды composer install.

5. а. Загружает все зависимости, указанные в файле

composer.json, и устанавливает их в проект.

6. а. Файл, который содержит информацию о версиях всех установленных зависимостей;

7. б. composer update

8. с. Указать в разделе require

9. а. Использует специальные версии и диапазоны версий, указанные в composer.json

10. а. composer remove <package-name>

11. а. Метод, который автоматически загружает ~~файлы~~ файлы классов PHP по мере их необходимости.

12. а. Добавляет секцию autoload в composer.json;

13. б. Это конфигурационный

файл, в котором описываются зависимости и методологии проекта.

14. а. Используя файл composer.json для указания минимальной версии PHP;

15. б. composer init

Вопрос на сопоставление

26.

1 C

2 A

3 D

4 B

5 E

16. - 25. Открытые вопросы

16.

Ответ 16

17.

Ответ 17

18.

Ambern 18

19.

Ambern 19

20.

Ambern 20

21.

Ambern 21

22.

Ambern 22

23.

Ambern 23

24.

Ambern 24

25.

Ambern 25

Бед ~~и~~

Мечен в 8

Ковцов С.В.

3 куге

(1 часам)

16. Файл composer.json:

- Это основной файл конфигурации для проекта, использующего Composer. Он содержит информацию о проекте и его зависимостях.
- Ключевые разделы для библиотеки:
 - name: Имя библиотеки в формате vendor/package.
 - description: Краткое описание библиотеки.
 - require: Список зависимостей, необходимых для работы библиотеки.
 - autoload: Настройки автозагрузки классов.
 - version: Версия библиотеки.
 - authors: Информация об авторах библиотеки.
 - license: Лицензия библиотеки.

Lesson 16

17. Добавление дополнительного репозитория:

- В файле `composer.json` можно добавить раздел `repositories`, чтобы указать дополнительные репозитории для загрузки библиотек.

- Пример:

```
{  
  "repositories": [  
    {  
      "type": "vcs",  
      "url": "https://github.com/vendor/package"  
    }  
  ]  
}
```

Пример 17

18. Создание и настройка автозагрузки классов:

- В файле `composer.json` добавляется раздел `autoload`.

- Пример:

```
{  
  "autoload": {  
    "psr-4": {  
      "Vendor\\Namespace\\": "src/"  
    }  
  }  
}
```

- После этого выполняется команда `composer dump-autoload`, чтобы сгенерировать файл автозагрузки.

Пример 18

19. Ограничение версий зависимостей:

- В разделе require можно указать ограничения на версии зависимостей.
 - Пример для версии выше 1.2.3 и ниже 2.0.0:
 - {
 - "require": {
 - "vendor/package": ">1.2.3 <2.0.0"
 - }
- }

Amber 19

20. Командная строка Composer:

- Composer предоставляет командную строку для управления зависимостями.
- Основные команды:
 - `composer install`: Устанавливает зависимости, указанные в `composer.lock`.
 - `composer update`: Обновляет зависимости до последних версий, соответствующих указанным ограничениям.
 - `composer require`: Добавляет новую зависимость.

Ответ 20

21. Настройка специфических версий PHP:

- В файле `composer.json` можно указать требуемую версию PHP в разделе `require`.
- Пример:
- {
- "require": {
- "php": "^7.4"
- }
- }

Пример 21

22. Несовместимые зависимости:

- Если зависимости не совместимы между собой, Composer выдаст ошибку и не установит их.
- Необходимо разрешить конфликт вручную, изменив версии зависимостей.

Ответ 22

23. Обновление всех зависимостей:

- Команда `composer update` обновляет все зависимости до последних доступных версий, соответствующих указанным ограничениям.

Дмвсм 23

24. Решение проблемы с конфликтующими зависимостями:

- Composer использует алгоритм разрешения зависимостей, чтобы найти совместимые версии.

Если конфликт не может быть разрешен, Composer выдаст ошибку.

Ответ 24

25. Создание и подключение библиотеки:

- **Шаг 1:** Создайте структуру проекта библиотеки.
- **Шаг 2:** Создайте файл `composer.json` с необходимыми разделами (`name`, `description`, `require`, `autoload` и т.д.).
- **Шаг 3:** Выполните команду `composer install` для установки зависимостей.
- **Шаг 4:** В основном проекте добавьте библиотеку в раздел `require` файла `composer.json`.
- **Шаг 5:** Выполните команду `composer update` для установки библиотеки в основной проект.

Доберн 25