



xdebug и улучшенный вывод var_dump()

Инженер-программист в БГУ

lifedu.by

Вечернее, заочное 18 мес., 24 мес.



Давайте посмотрим, один из самых широко используемых методов отладки это вызов функции **var_dump()**. Нет ничего плохого в использовании **var_dump()**. Я делаю это все время. Препятствием является то, что для того чтобы произвести отладку, используя эту функцию необходимо модифицировать код программы. xdebug предоставляет интересную альтернативу использования **var_dump()** для целей отладки кода, мы рассмотрим это в следующих статьях. Но сейчас вы будете рады слышать, что даже xdebug улучшает работу вашего любимого отладчика **var_dump()**. Когда загружено расширение **xdebug**, вывод функции **var_dump()** автоматически становится намного лучше для улучшения читаемости, скриншот показан ниже:

```
array
  1 => string 'Apple' (length=5)
  2 => string 'Pear' (length=4)
  3 => string 'Banana' (length=6)

object(Test) [1]
  public 'name' => string 'Test' (length=4)
  protected 'connected' => boolean false
  protected 'foo' =>
    object(Foo) [2]
      protected 'foo' => int 2357231
      protected 'bar' => float 1234723.234
```

Вы можете настроить как именно xdebug должен формировать вывод функции **var_dump()** с помощью различных настроек в **php.ini**. Во-первых, вы можете изменить длину выводимой строки. Значение по умолчанию 512; более длинные строки обрезаются автоматически. В зависимости от того выводить ли полностью строку или нет, зависит от ситуации и данных, с которыми вы работаете. Если вы работаете с большими строками, вывод функции **var_dump()** может быть слишком длинным и трудно читаемым, поэтому идея укорачивания строки выглядит достаточно приятной. В противоположность, если вы работаете со специфическими значениями, вы, скорее всего, захотите увидеть значение полностью. Для изменения длины строки выводимой xdebug добавьте

```
xdebug.var_display_max_data=
```

в **php.ini** и перезапустите ваш веб-сервер. В качестве альтернативы, вы можете изменить эту настройку в своем скрипте используя функцию **ini_set**, вы можете добавить

```
ini_set('xdebug.var_display_max_data', );
```

в начале вашего скрипта. Проверьте, чтобы вызов **ini_set** был до первого вызова **var_dump()**. Конфигурация xdebug во время выполнения избавит вас от перезапуска веб-сервера каждый раз когда вы меняете **php.ini** и позволит более гибко его настраивать. Вы также можете контролировать количество элементов массива и свойств объектов, которых будет выводить xdebug. Этого можно достигнуть модифицируя **xdebug.var_display_max_children** (значение по умолчанию 128). Этого значения будет вполне достаточно для отображения свойств вашего объекта, но если вы работаете с массивами, может быть необходимо увеличить значение. Если вы работаете с вложенными объектами или массивами, вы можете модифицировать **xdebug.var_display_max_depth**. Эта настройка имеет значение по умолчанию 3, значит, что отобразиться три измерения в

Информация

Это неофициальный ресурс. Если вы нашли на сайте ошибку или у вас есть интересная статья о Xdebug, пожалуйста, сообщите в разделе контакты.

массиве и три уровня вложенности в объекте.

Вы также можете вывести значения суперглобальных переменных, используя функцию **xdebug_dump_superglobals()**. Так как суперглобальные массивы, особенно **\$_SERVER**, - это массивы, содержащие большое количество значений, вы должны явно указать xdebug какой ключ массива вы хотите видеть. Чтобы сделать это установите **xdebug.dump.**, где это один из следующих имен **GET, POST, SERVER, COOKIE, FILES, REQUEST** или **SESSION**. Используйте ключи массивов, которые вы хотите вывести с помощью xdebug в качестве аргументов, если вы хотите выводить несколько значений, то перечислите их через запятую. Вы можете использовать ***** как шаблон для отображения ключей, который может особенно полезен для вывода **\$_GET** и **\$_POST**. Используйте

```
ini_set('xdebug.dump.SERVER', 'HTTP_HOST, SERVER_NAME')
```

в вашем PHP скрипте или настройки **xdebug.dump.SERVER=HTTP_HOST, SERVER_NAME** в **php.ini** для отображения значений **\$_SERVER['HTTP_HOST']** и **\$_SERVER['SERVER_NAME']**. Для отображения всех значений **GET**, которые были переданы в скрипт, используйте

```
xdebug.dump.GET=*
```

Dump **\$_SERVER**

| | | |
|---------------------------------------|---|-------------------------------|
| <code>\$_SERVER['HTTP_HOST']</code> | = | string 'localhost' (length=9) |
| <code>\$_SERVER['SERVER_NAME']</code> | = | string 'localhost' (length=9) |

Dump **\$_GET**


| | | |
|--------------------------------|---|---------------------------------|
| <code>\$_GET['test']</code> | = | string 'some value' (length=10) |
| <code>\$_GET['command']</code> | = | string 'execute' (length=7) |
| <code>\$_GET['gnu']</code> | = | string '' (length=0) |

По умолчанию xdebug не выводит неопределенные (**undefined**) переменные. Для того чтобы все же отображать неопределенные переменные, установите параметр **xdebug.dump_undefined** в **On**. Я советую установить этот параметр в оп. [оригинал статьи](#)

Видео находится у меня



Яндекс

 Xdebug - незаменимый инструмент PHP программиста. Этот отладчик позволяет получить подробную информацию о работе PHP скрипта, а так же профайлинг и трассировка скриптов все это значительно облегчают работу разработчика. Xdebug создан by Derick Rethans.

tech

Оставьте свой отзыв