

Forum:[Hack Forums](#)

Thread: [Is MVC still the most efficient way to code PHP?](#)

HappinessIsKey - October 13, 2015, 4:40 pm

Hey I was wondering if creating a Model View Framework was still the most efficient way to use PHP. If so could someone post a little guidance down below on how to actually create an MVC application. If MVC is not the most efficient way what is? Thank you :)

dersuss - October 13, 2015, 4:43 pm

There is no best way to do anything.

Some people say object oriented is the best way to do things, other people argue that since they say it slows things down because of the abstraction and the use of many objects which causes alot of memory usage.

Look at Linus torvalds how he reacts to C++.

My opinion is do whatever is faster :P, but that's just me.

6577657534735 - October 13, 2015, 4:59 pm

Depends on what you mean by efficient. Realistically you could make a really optimized application consisting of just functions and globals that outperforms an application designed using MVC. But it will be shit and unmaintainable. It's a tradeoff between performance and good design practices. Do what you want

HappinessIsKey - October 13, 2015, 5:01 pm

(Oct 13, 2015 04:59 PM)Abraluna Wrote: [\[CITE\]](#) Depends on what you mean by efficient.

Realistically you could make a really optimized application consisting of just functions and globals that outperforms an application designed using MVC. But it will be shit and unmaintainable. It's a tradeoff between performance and good design practices. Do what you want

When I say 'efficient' I mean how easy it is to go back and change a section of the code without it effecting everything in the entire application.

dersuss - October 14, 2015, 4:18 am

(Oct 13, 2015 04:59 PM)Abraluna Wrote: [\[CITE\]](#) Depends on what you mean by efficient.

Realistically you could make a really optimized application consisting of just functions and globals that outperforms an application designed using MVC. But it will be shit and unmaintainable. It's a tradeoff between performance and good design practices. Do what you want

MVC could be used in procedural programming not just OOP practices.

Also you could make a really good code in procedural that will be readable and maintainable. That's just a matter of practice and experience.

SIM-1947090 - October 14, 2015, 9:16 am

I would definitely go for MVC pattern..

And, those who say "OOP makes applications slow" is just...Seriously? ?

SIM-1947090 - October 14, 2015, 1:25 pm

(Oct 14, 2015 12:07 PM)Yani Wrote: [\[CITE\]](#) MVC has it's uses. It's only good for people who always use frameworks with a MVC model (as a base for starting their websites) or for projects with multiple coders.

Don't believe coders who say it's the "good way". It's only a possibility that you either love or hate.

OOP on the other hand is something where you define "stuff" as objects, and is definitely worth using.

Excuse me, but i have to be rude here, i find myself quite offended.

- 1.) "It's only good for people who always use frameworks with a MVC model" - this is wrong big time. What made you write that? You do realize separating application logic from "user interface" is a giant advantage in writing your code/applications?
- 2.) "or for projects with multiple coders." - this is like 1% of what MVC is good for (there is still like 99% to accomplish)
- 3.) "It's only a possibility that you either love or hate." - seriously? something you either love or hate?
- 4.) "OOP on the other hand is something where you define "stuff" as objects" - just dont say you think this piece of code makes it OOP - "\$ob = new SomeClass();"??

Once again, i apologize for being rude, but this just made me angry ?

SIM-1947090 - October 17, 2015, 12:44 pm

(Oct 14, 2015 03:02 PM)Yani Wrote: [CITE](#) No offence taken. I can't always bring over this point of view in English as it isn't my main language.

1. OOP can do that. MVC is something different
2. Nope, that's what MVC and frameworks are actually really useful for
3. Yeah, because it's something that defines a new "logic". As I said, it's a possible way of doing it, but not the right or wrong way. Just a way. (and let's look at it as "like" or "dislike" instead of love or hate)
4. I'm talking about the big picture. You could define a lot as an "object". Ex: a user, an alert, breadcrumbs, posts, tables, etc..

MVC could be a good thing to learn, but it wouldn't affect how your website reacts towards the end-user depending on your skills. MVC & Frameworks are so much overkill for 99% of the website I see from people of HF.

In my eyes, MVC only depends on how you feel comfortable perceiving the logic behind your code. Sorry for late response (as i said in other post, i was busy...)

- 3.) Well, thats something i can accept actually. It's better rephrased ;)
- 4.) In that case, i accept your statement

And aah, if you are talking about websites, then yes, you are correct. MVC and frameworks are very likely going to be overkill for a website.

I probably didnt understand this topic correctly, i had web applications in my mind (real applications, not some random web site).