

1 Layer-focused survey

Good and Bad Practices in Web Application Development

This research is focused on good and bad practices in web application development. Our aim is to improve the quality of software development. Help us out to find what the best and bad practices are.

You will take around 10 15 minutes to fill up this survey. It may look like a long time to spend in a single survey, but remember: your 10 minutes may be really helpful for you in the future.

We divided the survey into many pages in this survey. So, please click on "Continue". We do appreciate your time.

A little bit about yourself

Q: Years of experience with software development, in general.

Q: Years of experience with web application development

Q: How many web applications have you worked that went to production?

Q: Language that you have used more to develop your web applications?

Q: How do you see yourself in terms of good practices (and code quality) of web application development? 1-10

Q: How often do you study/read about good practices in web application development? 1-5

Q: What is the framework you have used more?

The "M" - Model

In MVC, "models" are responsible to all business logic, as well as any other logic, such as data access, webservice, and so on.

Q: Is there any characteristic that would make a infrastructure code in domain classes **problematic**? Is there any characteristic that would make it **not problematic**?

Q: Is there any characteristic that would make a long method **problematic**? Is there any characteristic that would make it **not problematic**?

Q: Do you have any good practices to deal with domain classes?

Q: Do you have anything you consider a bad practice when dealing with domain classes?

Data Access Objects (DAOs)

A web app usually consumes a database. It is common to isolate that code in DAOs.

Q: Is there any business logic that you **would** consider a problem in a DAO? Is there any business logic what you **would not** consider a problem in a DAO?

Q: Do you have any good practices to deal with DAO classes?

Q: Do you have anything you consider a bad practice when dealing with DAO classes?

The "C" - Controller

Controllers are responsible to connect the web world (with all that HTTP requests and responses) and the domain model (with all the business logic).

Q: Is there any business logic that **would be** a problem in a controller? Is there any business logic that **would not be** a problem?

Q: Do you have any good practices to deal with Controller classes?

Q: Do you have anything you consider a bad practice when dealing with Controller classes?

The "V" - View

Views are responsible to present the data and interact with the user. Developers usually use a mixture of HTML, CSS, LESS, SASS, Javascript, etc.

Q: Is there any characteristic that *would* make a scriptlet problematic? Is there any characteristic that *would not* make it problematic?

Q: Do you have any good practices to deal with View files?

Q: Do you have anything you consider a bad practice when dealing with View files?

Last thoughts

Q: Are there any other *GOOD* practices we did not asked you or you did not said yet?

Q: Are there any other *BAD* practices we did not asked you or you did not said yet?

2 Role-focused survey

My name is Mauricio, I am a PhD student researching code quality. I found your Github profile, and I believe your experience in Spring MVC can be really helpful to us.

We are studying code smells in Spring MVC. By smell, we mean symptoms of poor design and implementation choices. Smells contribute to technical debt, and possibly affect the maintainability of a software system.

We want you to tell us about your experience. For each architectural role in Spring, we want you to describe smells you have faced and why you think they are smells. The survey has around 10 questions and requires less than 15 minutes to complete.

If you have any questions, just contact us at mauricioaniche@gmail.com.

Background

Q: What is your current position?

Q: How many years of experience in software development do you have?

Q: How many years of experience in Spring MVC do you have?

Q: How many Spring MVC applications have you put in production?

Smells in Spring MVC

We want you to tell us about your experience. For each architectural role in Spring, we want you to describe smells (all of them!) you have faced and why you think they are smells. By smell, we mean symptoms of poor design and implementation choices. Smells contribute to technical debt, and possibly affect the maintainability of a software system. If you perceive the same smell in more than one architectural role, please copy and paste or refer to it.

Q: What are the smells you face in Controllers?

Q: What are the smells you face in Entities?

Q: What are the smells you face in Services?

Q: What are the smells you face in Components?

Q: What are the smells you face in Repositories?

Final thoughts

Q: Do you have any other thoughts on smells in Spring MVC which you haven't mention yet?

Q: Can we contact you for an interview if we need to?

3 Interviews with developers

Q: What is your name?

Q: What is your current position?

Q: How many years of experience in software development do you have?

Q: How many years of experience in Web development do you have?

Q: What are the smells you face in Controllers?

Q: What are the smells you face in Entities?

Q: What are the smells you face in Services?

Q: What are the smells you face in Components?

Q: What are the smells you face in Repositories?