

# What is an example of the Single Responsibility Principle? [closed]

Asked 12 years, 7 months ago   Modified 4 years, 10 months ago

Viewed 40k times



44



**Closed.** This question needs to be more [focused](#). It is not currently accepting answers.

💡 **Want to improve this question?** Update the question so it focuses on one problem only by [editing this post](#).

Closed 7 years ago.

[Improve this question](#)

Can someone give me an example of the Single Responsibility Principle? I am trying to understand what it means, in practice, for a class to have a single responsibility as I fear I probably break this rule daily.

oop

definition

solid-principles

single-responsibility-principle

design-principles

edited Sep 3, 2017 at 13:17



Steve Chambers

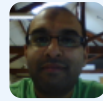
39.3k ● 29 ● 172 ● 220

Share

Improve this question

Follow

asked May 16, 2012 at 13:51



Sachin Kainth

46.7k ● 84 ● 208 ● 308

Take a look here: [stackoverflow.com/questions/659232/...](http://stackoverflow.com/questions/659232/...)

– Adolfo Perez May 16, 2012 at 13:55

[phpfreaks.com/tutorial/oo-php-part-2-boring-oo-principles](http://phpfreaks.com/tutorial/oo-php-part-2-boring-oo-principles)

– Jack May 16, 2012 at 14:03

Here's a good article about it. It help me a lot to figure out what Single Responsibility Principle mean

[http://www.codeproject.com/Articles/611593/SOLID-](http://www.codeproject.com/Articles/611593/SOLID-Principles-Single-Respons)

[Principles-Single-Respons](http://www.codeproject.com/Articles/611593/SOLID-Principles-Single-Respons) – Thermech Nov 23, 2013 at 15:59

2 Answers

Sorted by: Highest score (default)



54



The most effective way to break applications is to create [GOD](#) classes. Those are classes that keep track of a lot of information and have several responsibilities. One code change will most likely affect other parts of the class and therefore indirectly all other classes that use it. That in turn leads to an even bigger maintenance mess since no one dares to do any changes other than adding new functionality to it.

The following example is a TypeScript class that defines a `Person`, this class should not include email validation

because that is not related with a person behaviour:

```
class Person {
    public name : string;
    public surname : string;
    public email : string;
    constructor(name : string, surname : string, email : string) {
        this.surname = surname;
        this.name = name;
        if(this.validateEmail(email)) {
            this.email = email;
        }
        else {
            throw new Error("Invalid email!");
        }
    }
    validateEmail(email : string) {
        var re = /^([\w-]+(?:\.[\w-]+)*)@((?![\w-]+)\.?)
{2,6}([a-z]{2})?)$/i;
        return re.test(email);
    }
    greet() {
        alert("Hi!");
    }
}
```

We can improve the class above by removing the responsibility of email validation from the Person class and creating a new `Email` class that will have that responsibility:

```
class Email {
    public email : string;
    constructor(email : string){
        if(this.validateEmail(email)) {
            this.email = email;
        }
        else {
            throw new Error("Invalid email!");
        }
    }
}
```

```

    }
}
validateEmail(email : string) {
    var re = /^([\w-]+(?:\.[\w-]+)*)@((?![\w-]+\.)([a-z]{2,6}(?:\.[a-z]{2})?)$)/i;
    return re.test(email);
}
}

class Person {
    public name : string;
    public surname : string;
    public email : Email;
    constructor(name : string, surname : string, email
        this.email = email;
        this.name = name;
        this.surname = surname;
    }
    greet() {
        alert("Hi!");
    }
}

```

Making sure that a class has a single responsibility makes it per default also easier to see what it does and how you can extend/improve it.

Share Improve this answer

Follow

edited Feb 13, 2020 at 23:58



Jason Butler

646 ● 1 ● 14 ● 27

answered Apr 3, 2015 at 15:21



Remo H. Jansen

24.9k ● 12 ● 72 ● 99

---

8 So now we'll get tons of single methos classes in our project?  
 – N.D.B Jan 5, 2017 at 7:49

---

8 Yes, you create more complex features by composing very simple single-concern entities. – [Remo H. Jansen](#) Jul 19, 2017 at 15:45

---

I think it takes time to be good to understand SRP  
– [Mahmood Garibov](#) Oct 4, 2022 at 20:28 ✎

---



Check out the [Solid description](#).

7

Unless you ask for something more specific, it will be hard to help more.



Single responsibility is the concept of a Class doing one specific thing (responsibility) and not trying to do more than it should, which is also referred to as High Cohesion.



Classes don't often start out with Low Cohesion, but typically after several releases and different developers adding onto them, suddenly you'll notice that it became a monster or God class as some call it. So the class should be refactored.

It's hard to think of a good example, but one I can think of recently would be a class we have that manages different packet processing stages, a type of **Chain of Responsibility**. The initial intention of this class was to maintain a list of stages and to orchestrate calling `packetProcess()` on them. Well, it ended up that everybody added anything to do with the processing stages (since the manager class was an easy place to access the stages) to this manager class, especially stage configuration. The manager class no longer had a

Single Responsibility, but instead was also responsible for making calls to the stages for configuration changes: thus the Cohesion had been reduced.

We ended up having to refactor the manager class, ripping out all the stage configuration and putting it in a factory, thus leaving the manager to do what it was intended to do.

[Share](#) [Improve this answer](#)

[edited May 16, 2012 at 14:26](#)

[Follow](#)

answered May 16, 2012 at 13:55



[Brady](#)

10.3k ● 3 ● 23 ● 61

- 
- 5 I ran into this few weeks ago. I needed an Object Factory class which would create instances of different types of objects, Serialize them, Persist them to DB, etc. My first thought was creating a Factory class with a Serialize Method, but as a I read about the SRP, it made more sense having a Class exclusively dedicated to Serialize, one class to Persist objects in DB, etc. This makes your application much more maintainable and modular. – [Adolfo Perez](#) May 16, 2012 at 14:05
-

- 3 You still haven't provided a good explanation... you didn't even seem to understand the concept yourself... Remo's answer should be the selected one, not your vague one. Also "Unless you ask for something more specific, it will be hard to help more" is a total cop-out and not helpfull at all, he was specific enough, give an example of SRP in classes. The only thing that would save this question is transforming your example into a psuedo code mve. – [Krupip](#) Jun 1, 2017 at 20:12
- 



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.