


# MVC

Theory

Practice

 100% completed, 0 problems solved

▼

## Theory

⌚ 4 minutes reading

Unskip this topic

Start practicing

When developing web or mobile applications, programmers use different patterns to make their code simpler and easier to work with. MVC is one such pattern. It describes how to separate the user interface from the business logic and data access logic. That way the user interface can be modified independently, so there is low coupling between the different parts of the application.

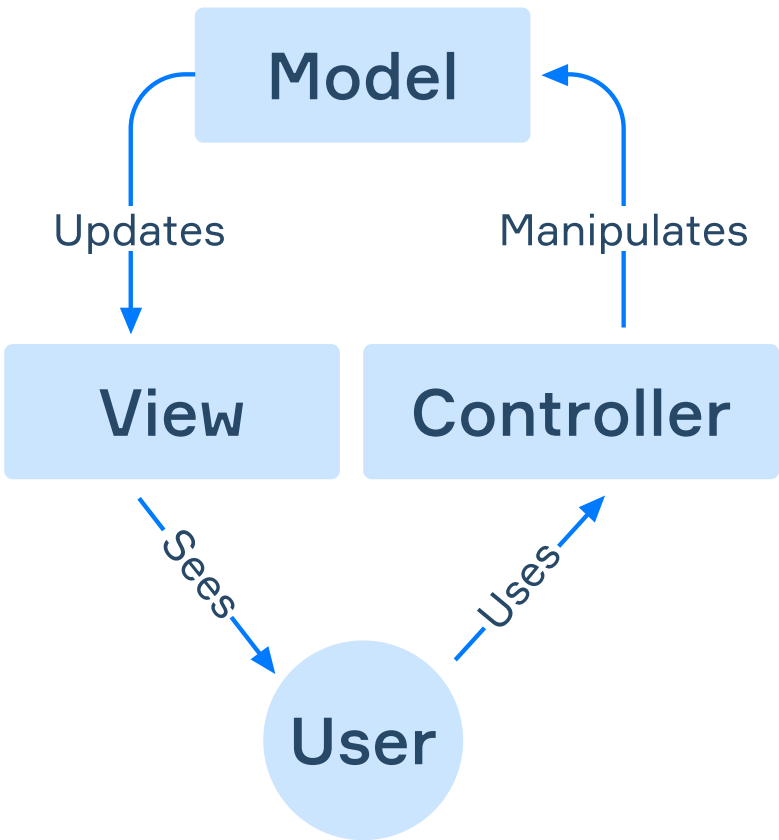
After reading this topic, you will learn about the components of the MVC pattern, as well as its advantages, and also consider when it is not a good idea to use MVC.

### §1. What are MVC and MVC components

**MVC (Model-View-Controller)** is an architectural pattern separating an application into three logical components:

- **Model** is responsible for all data and its related logic;
- **View** is responsible for presenting data to users or handling user interaction;
- **Controller** informs the Model of the need for changes.


In order to understand how to work with these components, consider the example below.




When the user clicks on interface elements, they interact with the *Controller*. The *Controller* accepts user input and interacts with the *Model*. The *Model* represents the state of an application. It can be data in a database, a file, an in-memory data, or something else. After modification, the *Model* updates the *View*, and its user sees this.

This describes the most basic version of the MVC architecture. There is also a common variant where there is a connection between *View* and *Controller*. The *Controller* in such cases provides the interconnection between the *Model* and *View* components.

#### 1 required topic






[Introduction to software architecture](#)

▼

#### 4 dependent topics



[Introduction to Spring Web MVC](#)

▼

[Django MVC](#)

▼

[Introduction to Flask](#)

▼

[Model-View-Controller \(MVC\) in Swing](#)

▼

#### Table of contents:

↑

[MVC](#)

[§1. What are MVC and MVC components](#)

[§2. Advantages and disadvantages of MVC](#)

[§3. Conclusion](#)

[Discussion](#)

## §2. Advantages and disadvantages of MVC

MVC has become a sought-after pattern and has been widely used because of its benefits:

- By separating into components, the flexibility, maintainability, and scalability of the application are increased.
- You can test components separately from each other.
- The components can be reused.
- Models can have multiple views.
- MVC allows you to configure different levels of security for different components.

And these are the disadvantages of the MVC pattern:

- One should keep in mind that it is not suitable for small applications. It makes simple applications more complex.
- MVC is also not suitable for high-performance applications. Sometimes it's more efficient to go through several layers of architecture.

MVC is not the only architectural pattern. This is worth bearing in mind if, for some reason, it is not suitable for the development of your application. There are several other patterns for similar needs: *MVP(Model-View-Presenter)*, *MVVM(Model-View-ViewModel)*. The MVC, MVP, and MVVM patterns are often called the **MV\* family**.

## §3. Conclusion

MVC is an architectural pattern that separates the user interface from business logic and business logic from data access logic. It has its pros and cons, and many implementations and interpretations. It's interesting to know that this pattern was originally used for desktop graphical user interfaces. Currently, MVC is used in the development of web applications and mobile applications. Therefore, its knowledge will be handy for you in the future.

 Report a typo

186 users liked this piece of theory. 2 didn't like it. What about you?



Start practicing

Unskip this topic

[Comments \(2\)](#)

[Useful links \(1\)](#)

[Show discussion](#)

 JetBrains Academy

Tracks

Pricing

For organizations

About

Contribute

Careers

 Become beta tester

Be the first to see what's new

[Terms](#) [Support](#)



Made with  by Hyperskill and JetBrains