



**FREE eBook**

**LEARNING**

**model-view-controller**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#model-  
view-**

**controller**

# Table of Contents

About.....	1
Chapter 1: Getting started with model-view-controller.....	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Credits.....	4

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [model-view-controller](#)

It is an unofficial and free model-view-controller ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official model-view-controller.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with model-view-controller

## Remarks

Model–view–controller (MVC) is an [architectural pattern](#) in software design, [not a design pattern](#), which describes a way to structure our application and the responsibilities and interactions for each part in that structure. [\[1\]](#)

Historically, MVC was first described in 1979 by [Trygve Reenskaug](#) as a way to approach (desktop) GUI (graphical user interface) design. Reenskaug described his early Smalltalk and object oriented conceptual efforts as follows: [\[2\]](#)

"MVC was conceived as a general solution to the problem of users controlling a large and complex data set. The hardest part was to hit upon good names for the different architectural components. Model-View-Editor was the first set. After long discussions, particularly with Adele Goldberg, we ended with the terms Model-View-Controller."

The wild popularization of MVC for web applications is due to its inclusion in two frameworks that became immensely popular: Struts and Ruby on Rails. These two environments marked the way for the hundreds of frameworks created later. [\[3\]](#)

### Advantages:

1. It's immensely popular.
2. It provides strong routing when used in conjunction with the Front Controller pattern. (web-based MVC)
3. It achieves separation of concerns.
4. It makes collaboration easier as there's less to be concerned with.
5. It makes it easier to follow how complicated applications work, thereby reducing bugs when contributors contribute.

### Disadvantages:

1. Abstraction can overly-complicate relatively simple applications.
2. The learning curve can be difficult as there's many concepts involved.
3. Many frameworks describe themselves as MVC, yet differ in implementation, thus one developer will have different habits from the next.

## Examples

### Installation or Setup

Detailed instructions on getting model-view-controller set up or installed.

Read **Getting started with model-view-controller** online: <https://riptutorial.com/model-view-controller/topic/2298/getting-started-with-model-view-controller>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with model-view-controller	<a href="#">Community</a> , <a href="#">Ultimater</a>