Introduction to MVC Architecture

What is MVC?

- The **Model-View-Controller (MVC)** is an *architecture* that divides an application into three main components:
  - The model,
  - The view, and
  - The controller.

Model:

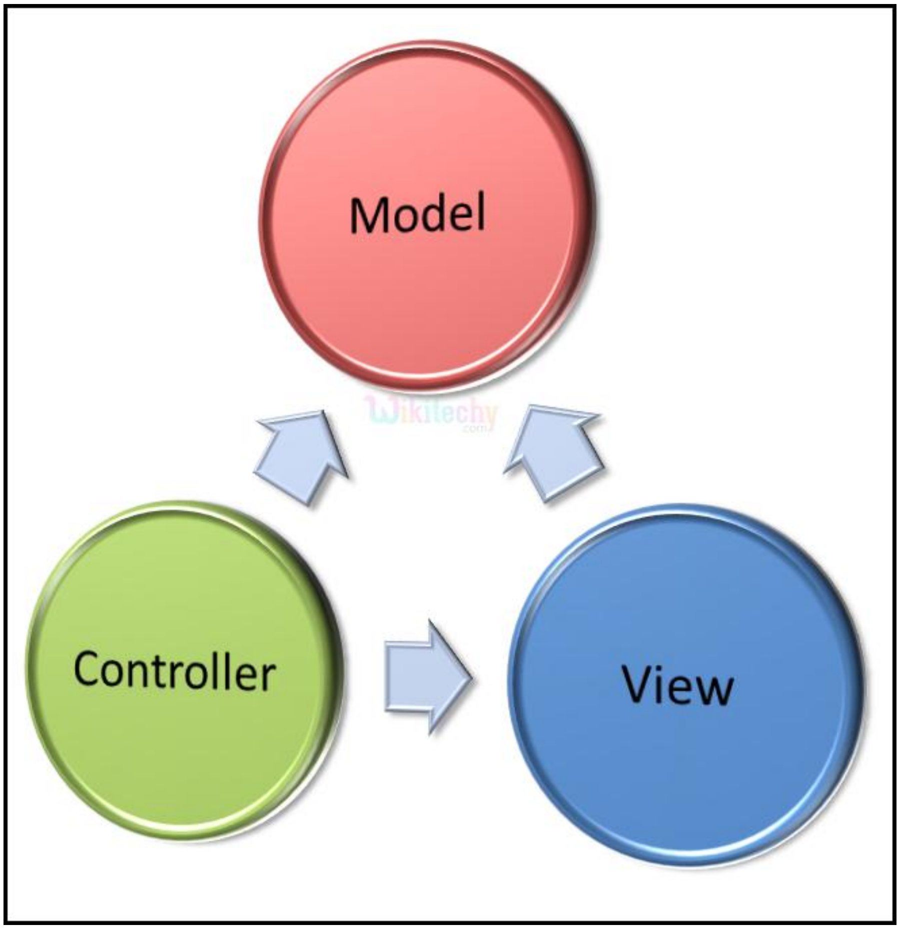
- It is the main business entity.
- The whole application depends and executes based on this entity.
- The data storage layer is compressed by this Model component.

View:

- It is the user interface component.
- It depicts the concepts of model entity via interactive UI.

Controller:

- It is a component to receive requests from a view.
- The request is then updates at the model level to insist on Model's state change.



Model-View-Controller

ASP.NET MVC Framework:

- The classes View, Controller and the Model are used to instrument MVC pattern in .NET.
- The ASP.NET MVC Framework has the following benefits:

1. Separation of tasks:

- Complex applications can easily be managed with component separation.

2. Test-Driven Development TDD:

- Behavior of applications can be tested in-depth with simulation objects.

3. Extensible and pluggable framework:

- ASP.NET MVC framework components are flexible enough to replace or customize.

4. Dependency Injection (DI):

- Objects can be injected into a class rather than waiting for a class to create its object.

5. Inversion of Control (IOC):

- Object access is made possible with configuration files.

6. URL routing mechanism:

- Understandable and searchable URLs can easily be added in applications for search engine optimizations.

7. Full control over application:

- The developer has more control on the application than the server hence reducing request and response times.

8. Support for ASP.NET features:

- ASP.NET features such as

- The architecture,
- Authentication and authorization,
- Membership and roles,
- Caching,
- Session etc.

are supported by MVC based applications.