

# Model-View-Controller Architecture Description

## What is MVC

- ▶ When a request like :  
`http://localhost:6790/home/get/1` comes in, it is received by the controller, who handles it.
- ▶ The controller communicates with the model, who has the classes that describe the data
- ▶ In addition to the data, the model can also have the logic to retrieve the data from the underlying data source
- ▶ In addition to the model, the controller selects a view and passes the model object to the view
- ▶ The view contains the logic to generate the html to present the model data i.e. the Customer model data provided through the view by the controller
- ▶ This html is then sent through the network to the user making the request

## Model-View-Controller Architecture Description

**MVC Model:** Let's understand the model, view and the controller in terms of the customer model. We want to retrieve the customer details and display them in an html table. In the completed application, the model is the Customer class and the CustomerRepository as in the next slide

**MVC View:** A view in MVC should only contain the logic to display the model data provided to it by the controller. The model object has the customer data,

but the model doesn't know anything about how to present this data. So this model object is provided to the view by the controller. There should be no complex logic in the view.

**MVC Controller:** When a request from the browser arrives at our application, it is the controller that handles the incoming http request and response to the user action.

### ***Summary***

MVC is design pattern for implementing User Interface layer of an application

- Model is a set of classes that represent data + logic to manage that data
- View contains the display logic to present the Model data provided to it by the controller
- Controller handles the http request, work with the model, and selects a view to render that model