Model, Model binding

Lesson Objectives

- *Model*
- *Model binding*

Section 1

MODEL

Model

- In ASP.NET MVC, model is a C# class to represent the data as well as to manage the data.
- Is accessible by both controller and view.

Model

- Can be used to pass data from controller action methods to a view.
- Can also be used by a view to display data in a page (HTML output).

Model from Entity

- In MVC project, we often use Entity Framework for simplifying the data access layer.
- Basically, entity class can be consider as a model of the MVC project.

Model from Entity

- Make clearly:
  - ✓ When the class used for manipulate data with database, it is entity
  - ✓ When the class used for modelling data, it is model

ViewModel

- In ASP.NET MVC application:
  - ✓ a single model object may not contain all the necessary data required for a view.
  - ✓ a complex model object may not present all properties to the client

=> we need to use the concept ViewModel

ViewModel

- ViewModel is used by view for present purpose
- ViewModel is used by controller for calculate data/bridge data to the entity
- ViewModel is not used by DbContext directly

Demo time

- Create Employee model with properties
  - ✓ EmployeeId
  - ✓ Name
  - ✓ Gender
  - ✓ Department
  - ✓ Salary
  - ✓ AddressId

Demo time

- Create Address model with properties
  - ✓ AddressId
  - ✓ Country
  - ✓ State
  - ✓ City
  - ✓ Pin

Demo time

- Create EmployeeDetails view model to combine information from 2 models
  - ✓ Name
  - ✓ Gender
  - ✓ Department
  - ✓ Country
  - ✓ State
  - ✓ City

Section 2

MODEL BINDING

Model Binding

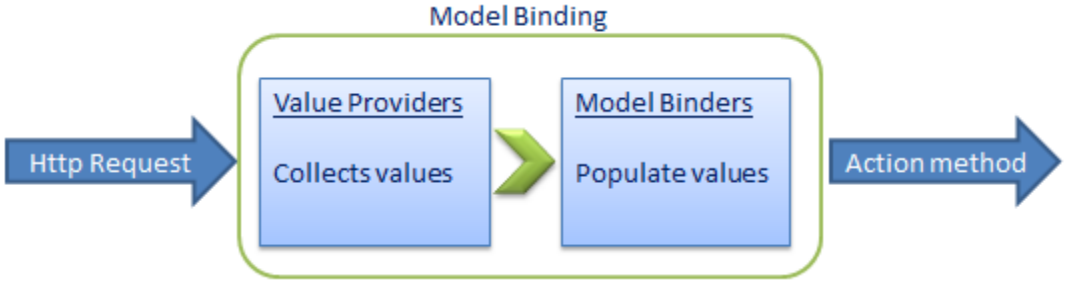
- How does the data come from the request to action?

Model Binding

- The process that MVC framework converts the http request values (from query string or form collection) to action method parameters.
- These parameters can be of primitive type or complex type.

Model Binding

- Step 1: collect values from the incoming http request
- Step 2: populate primitive type or complex type with these values

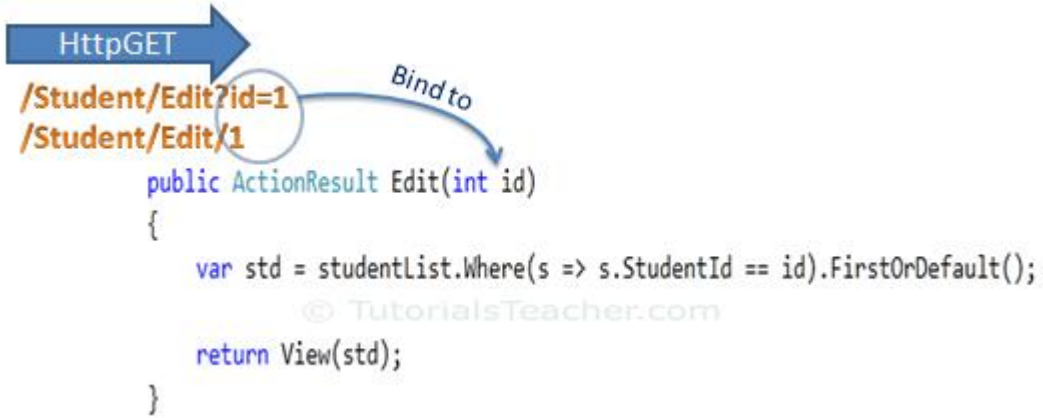


Model Binding

- By default, the value provider collection evaluates values from the various sources in the following order:
  1. Previously bound action parameters, when the action is a child action
  2. Form fields (Request.Form)
  3. The property values in the JSON Request body (Request.InputStream), but only when the request is an AJAX request
  4. Route data (RouteData.Values)
  5. Querystring parameters (Request.QueryString)
  6. Posted files (Request.Files)

Binding to Primitive type

- HttpGET request embeds data into a query string.
- MVC framework automatically converts a query string to the action method parameters.



Binding to Primitive type

- Query string values will be converted into parameters based on matching name.
- We can have multiple parameters in the action method with different data types.
- This binding is case insensitive. So "id" parameter can be "ID" or "Id".

Example

- URL: *http://localhost/Student/Edit?id=1&name=John*
- Would be expected to map to **id** and **name** parameters of the action

```
public ActionResult Edit(int id, string name)
{
    // do something here

    return View();
}
```

Binding to Complex type

- Model binding in MVC framework automatically converts form field data of HttpPOST request to the properties of a complex type parameter of an action method.
- Mapping is based on name of form field data, NOT by id

Binding to Complex type

- <input name="**StudentName**"> => std.**StudentName**
- <select name="**Age**"> => std.**Age**

FormCollection

- We can include FormCollection type parameter in the action method instead of complex type,
- To retrieve the value from view form field, use name as a key of dictionary

Bind Attribute

- All properties is bound automatically.
- Use [Bind] attribute to specify the exact properties a model binder should include or exclude in binding.
  - ✓ To include property(ies), use Include parameter
  - ✓ To exclude property(ies), use Exclude parameter

Bind Attribute

- The Bind attribute will improve the performance
  - ✓ only bind properties which needed
  - ✓ reduce data volume
  - ✓ save time to convert data
- The Bind attribute will improve security
  - ✓ prevent to update/modify unnecessary property

Lesson Summary

- Model is a C# class to represent the data as well as to manage the data.
- Model binding is process that MVC framework converts the http request values (from query string or form collection) to action method parameters.
- These parameters can be of primitive type or complex type.

Thank you