Self-Study

1. What is the class ModelStateDictionary ?

ModelStateDictionary in ASP.NET Core represents the state of an attempt to bind values from an HTTP request to an action method and includes validation information.

It tracks errors that come from two subsystems: model binding and model validation. Model binding errors occur when trying to bind incoming request data to action parameters or model properties, often due to type mismatches or conversion errors. Model validation errors happen when checking business rules or data annotations on the bound model after successful binding.

ModelStateDictionary stores the validation state and error messages for each input field or property. It helps determine if the received input is valid with properties like IsValid, and it supports adding errors manually with methods like AddModelError.

In controllers, the ModelState property is of type ModelStateDictionary, and it is typically checked before executing business logic in the action method to ensure the incoming data is valid.

Example of accessing and using ModelStateDictionary in ASP.NET Core:

```
public class BooksController : ControllerBase
{
    // POST /books
    [HttpPost]
    public IActionResult Post([FromBody] CreateBookInputModel model)
    {
        // Check if model binding and validation succeeded
        if (!ModelState.IsValid)
        {
            // Return bad request with validation errors
            return BadRequest(ModelState);
        }
        // Custom validation: check if a book with the same title already exists
        if (BookExists(model.Title))
        {
            // Add a custom model error with key "" (general error)
            ModelState.AddModelError("", "A book with the same title already exists.");
            return BadRequest(ModelState);
        }

        // Proceed with further processing if valid
        SaveBook(model);
        return Ok(model);
    }

    private bool BookExists(string title)
    {
        // Pretend check for existing book title in database
        return false;
    }

    private void SaveBook(CreateBookInputModel model)
    {
        // Save the book logic here
    }
}

public class CreateBookInputModel
{
    public string? Title { get; set; }
    public string? Description { get; set; }
}
```

Explanation:

- ModelState.IsValid checks if the model passed built-in validation and binding.

- If not valid, errors are returned to the client with a 400 Bad Request response.
- Custom errors can be added using ModelState.AddModelError to provide business rule validation feedback.
- The errors stored in ModelStateDictionary map input field keys to error messages for easy return and display.

In summary, ModelStateDictionary is a key class that tracks binding and validation states of user input in ASP.NET Core, providing error management and validation status during HTTP request processing.