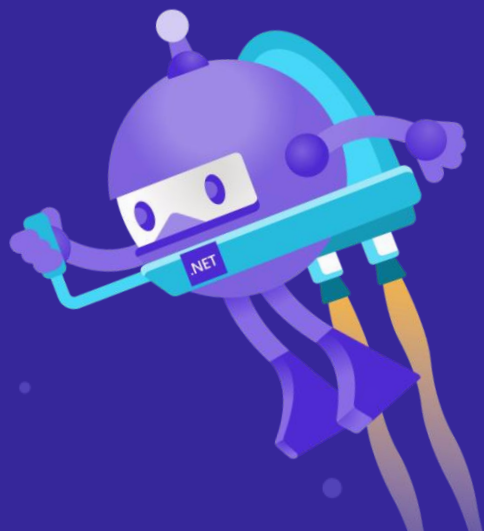
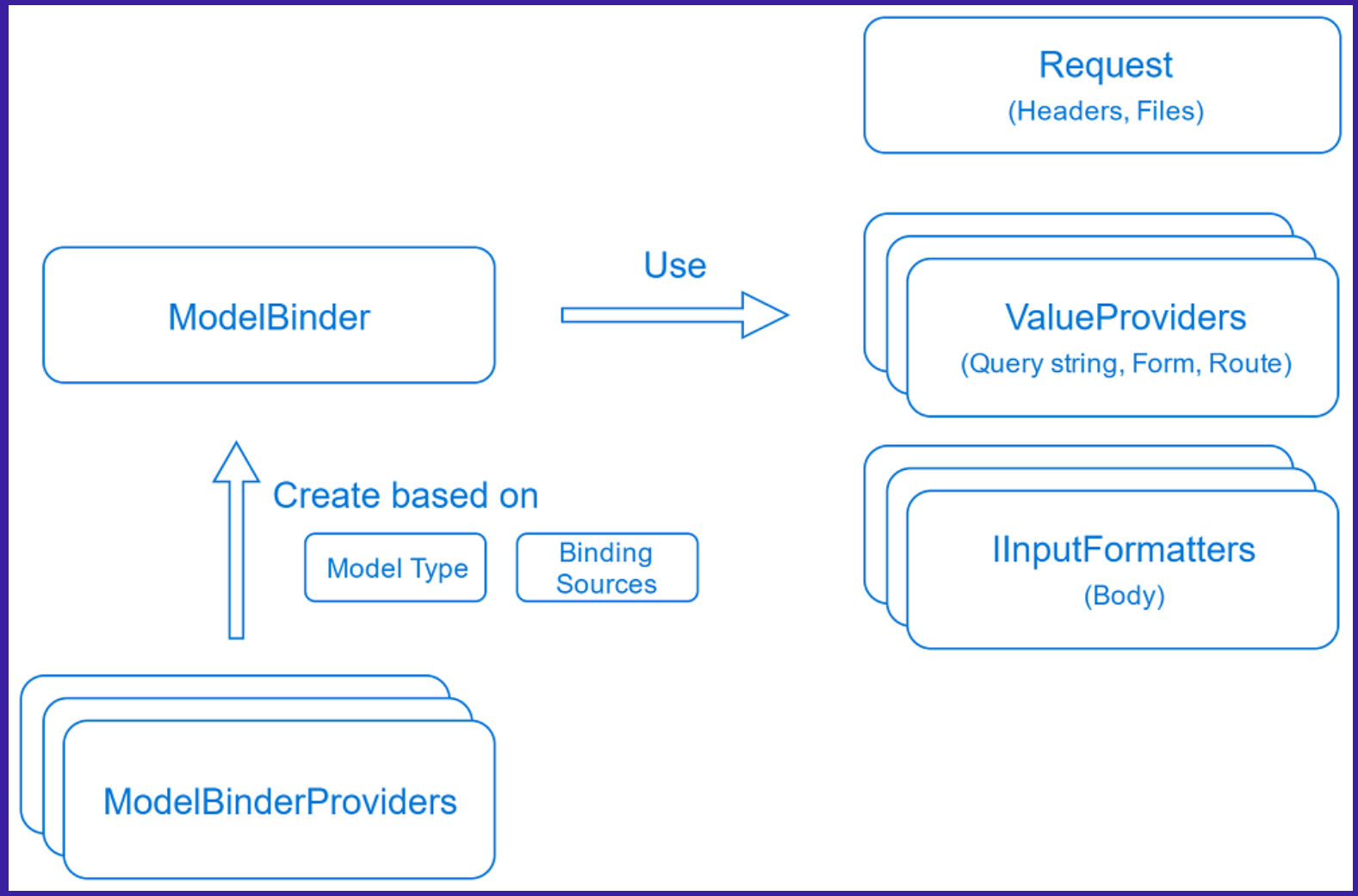
Day 10 : Model Binding, Input Validation, CORS



Model Binding

- Allows controller actions to work directly with model types
- Handles mapping between incoming request data and application models
- Can extend built-in model binding functionality by implementing custom model binders

Model Binding



Model Binding – Behind the Scenes

- Retrieves data from various sources such as route data, form fields, and query strings.
- Provides the data to controllers and Razor pages in method parameters and public properties.
- Converts string data to .NET types.
- Updates properties of complex types.

Model Binding – Sources

- Form fields
- The request body (For controllers that have the [ApiController] attribute.)
- Route data
- Query string parameters
- Uploaded files

Model Binding – Input Formatters

- Data in the request body can be in JSON, XML, or some other format.
- Model binding uses an input formatter to handle a particular content type
- By default, ASP.NET Core includes JSON based input formatters for handling JSON data
- Selects input formatters based on the Consumes attribute. If no attribute is present, it uses the Content-Type header

Custom Model Binding

- Create a new class by inheriting IModelBinder
- Define the logic inside BindModelAsync method

```
public class CustomModelBinder : IModelBinder
    {
        public Task BindModelAsync(ModelBindingContext bindingContext)
        {
            throw new NotImplementedException();
        }
    }
```

Custom Model Binding - Usage

- Register the custom binder using the ***ModelBinder*** attribute
- Can be either used on Model or on action methods

```
[ModelBinder(BinderType = typeof(CustomModelBinder))]
    public class User
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Address { get; set; }
    }
//in action method
[HttpGet]
[Route("test")]
public IActionResult Index([ModelBinder(BinderType = typeof(CustomModelBinder))]User
u)
{
```

Model Validation

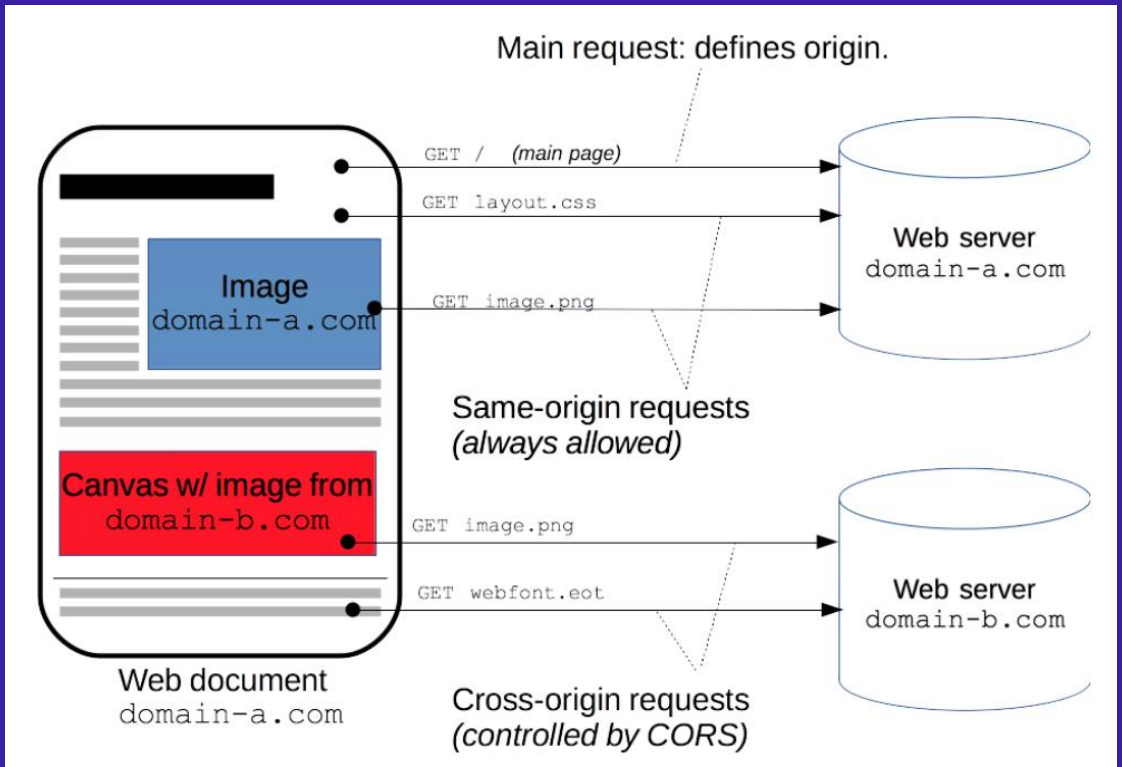
- Is the process of ensuring the input data is appropriate to bind to the Model
- ModelState represents errors coming from two subsystems
  - Model Binding
  - Model Validation
- Both these errors occur before the action method is executed

Validation Attributes

- Lets you to specify validation rules for model properties
- Built-in attributes
  - CreditCard
  - Compare
  - EmailAddress
  - Phone
  - Range
  - RegularExpression
  - Required
  - StringLength
  - Url
  - Remote
- Allows to implement Custom attribute

CORS (Cross –Origin Resource Sharing)

- Is a mechanism which allows controlled access to resources located outside of your domain



CORS (Cross –Origin Resource Sharing)

- Is a W3C standard that allows server to relax the same origin policy
- Allows a server to explicitly allow some cross-origin requests while rejecting others
- CORS is not a security feature

SAME ORIGIN Policy

- Two URLs have the same origin if they have identical schemes, hosts, and ports

These two have same origins:
- https://test.com/page1.html
- https://test.com/page2.html

The ones below belongs to different domains
- https://test.net: ***Different domain***
- https://www.test.com/foo.html: ***Different subdomain***
- http://test.com/foo.html: ***Different scheme***
- https://test.com:9000/foo.html: ***Different port***

Working With Databases

- ADO.NET is a set of classes that expose data access services
- Provides a rich set of components for creating distributed, data-sharing applications
- Can use ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain
- Includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results
- ADO.NET classes are found in System.Data.dll

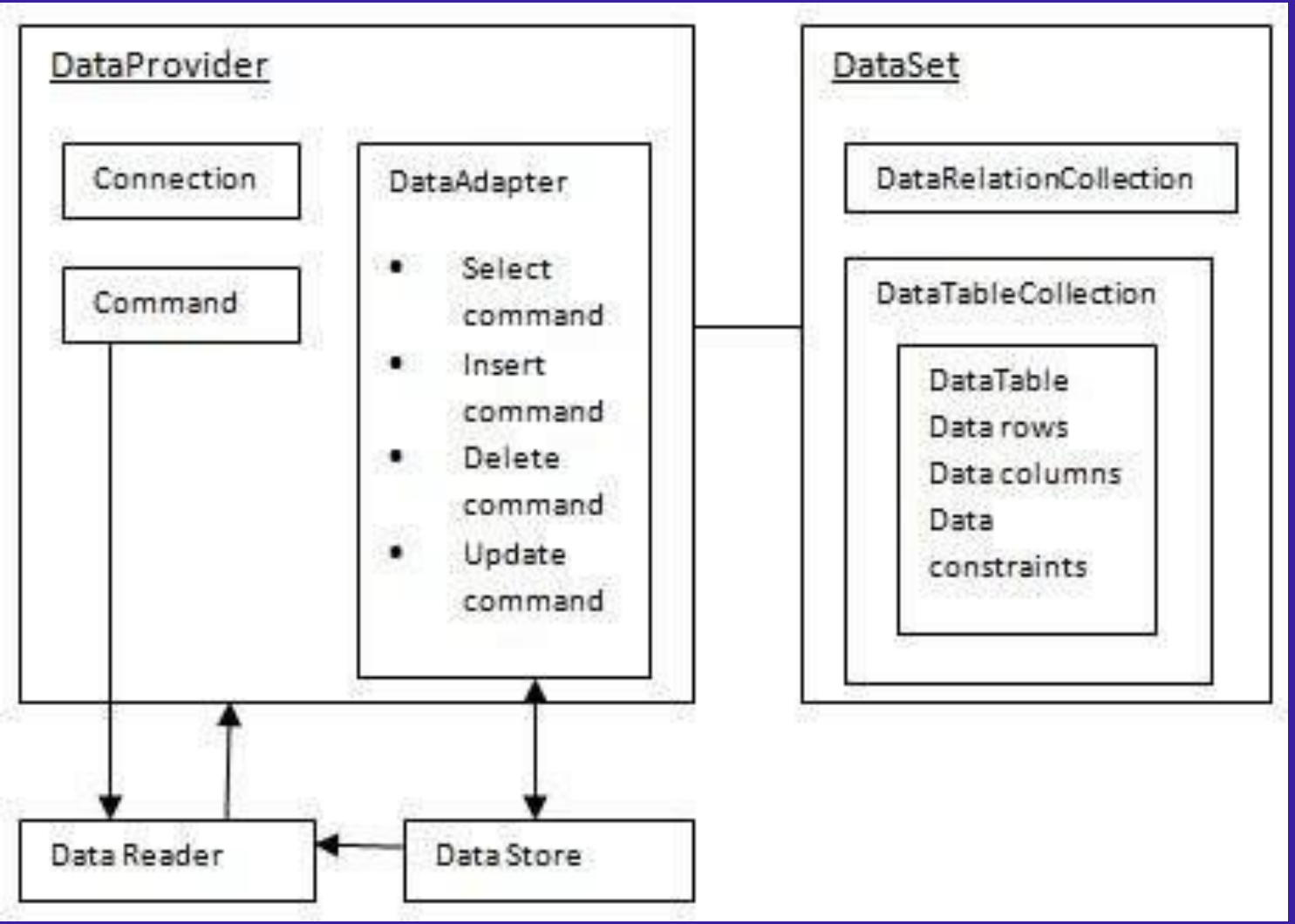
ADO.NET

- Mainly comprised of two components
  - .NET Framework Data Providers

    are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data
  - DataSet

    Is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application

ADO.NET


DataProvider
Connection
Command
DataAdapter
Select command
Insert command
Delete command
Update command
DataSet
DataRelationCollection
DataTableCollection
DataTable
Data rows
Data columns
Data constraints
Data Reader
Data Store

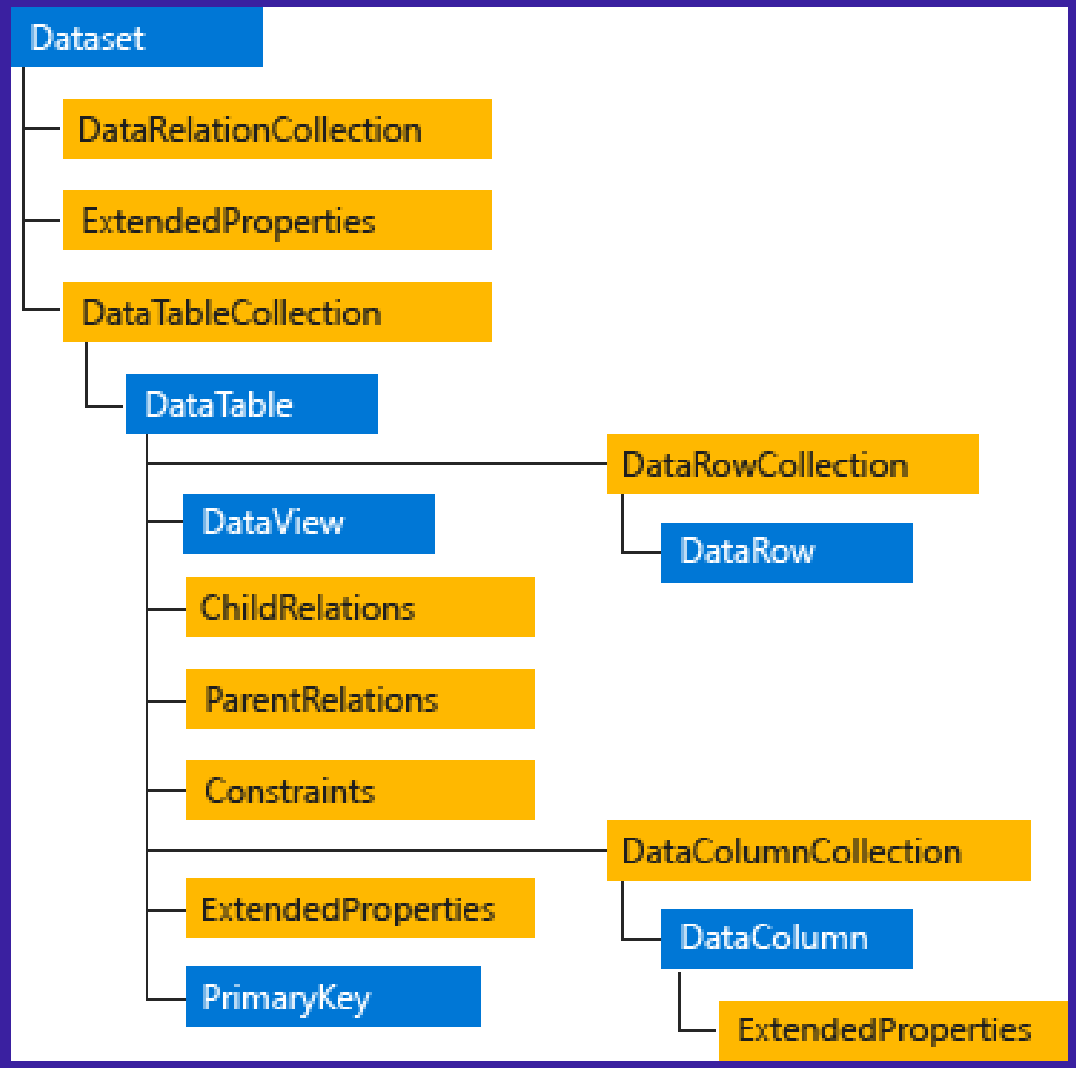
.NET Framework Data Providers

.NET Framework data provider	Description
.NET Framework Data Provider for SQL Server	Provides data access for Microsoft SQL Server. Uses the System.Data.SqlClient namespace.
.NET Framework Data Provider for OLE DB	For data sources exposed by using OLE DB. Uses the System.Data.OleDb namespace.
.NET Framework Data Provider for ODBC	For data sources exposed by using ODBC. Uses the System.Data.Odbc namespace.
.NET Framework Data Provider for Oracle	For Oracle data sources. The .NET Framework Data Provider for Oracle supports Oracle client software version 8.1.7 and later, and uses the System.Data.OracleClient namespace.
EntityClient Provider	Provides data access for Entity Data Model (EDM) applications. Uses the System.Data.EntityClient namespace.
.NET Framework Data Provider for SQL Server Compact 4.0.	Provides data access for Microsoft SQL Server Compact 4.0. Uses the System.Data.SqlServerCe namespace.

.NET Framework Data Providers – Core Objects

Object	Description
Connection	Establishes a connection to a specific data source. The base class for all Connection objects is the DbConnection class.
Command	Executes a command against a data source. Exposes Parameters and can execute in the scope of a Transaction from a Connection. The base class for all Command objects is the DbCommand class.
DataReader	Reads a forward-only, read-only stream of data from a data source. The base class for all DataReader objects is the DbDataReader class.
DataAdapter	Populates a DataSet and resolves updates with the data source. The base class for all DataAdapter objects is the DbDataAdapter class.

DataSet


Dataset
DataRelationCollection
ExtendedProperties
DataTableCollection
DataTable
DataRowCollection
DataRow
DataView
ChildRelations
ParentRelations
Constraints
DataColumnCollection
ExtendedProperties
DataColumn
PrimaryKey
ExtendedProperties

Thanks for joining!