In this article, we’ll discuss how to use **jQuery Ajax for**[ASP.NET](http://asp.net/)**Core MVC CRUD Operations** using Bootstrap Modal. With jQuery Ajax, we can make HTTP request to controller action methods without reloading the entire page, like a single page application.

To demonstrate CRUD operations – insert, update, delete and retrieve, the project will be dealing with details of a normal bank transaction. GitHub repository for this demo project : <https://bit.ly/33KTJAu>.

Sub-topics discussed :

* Form design for insert and update operation.
* Display forms in modal popup dialog.
* Form post using jQuery Ajax.
* Implement MVC CRUD operations with jQuery Ajax.
* Loading spinner in .NET Core MVC.
* Prevent direct access to MVC action method.

**Create**[ASP.NET](http://asp.net/)**Core MVC Project**

In Visual Studio 2019, Go to *File > New > Project (Ctrl + Shift + N)*.

From new project window, Select [Asp.Net](http://asp.net/) Core Web Application\_.\_

Once you provide the project name and location. Select Web Application(Model-View-Controller) and uncheck *HTTPS Configuration*. Above steps will create a brand new [ASP.NET](http://asp.net/) Core MVC project.

**Setup a Database**

Let’s create a database for this application using Entity Framework Core. For that we’ve to install corresponding NuGet Packages. Right click on project from solution explorer, select Manage NuGet Packages\_,\_ From browse tab, install following 3 packages.

Now let’s define DB model class file – */Models/TransactionModel.cs*.

public class TransactionModel

{

[Key]

public int TransactionId { get; set; }

[Column(TypeName ="nvarchar(12)")]

[DisplayName("Account Number")]

[Required(ErrorMessage ="This Field is required.")]

[MaxLength(12,ErrorMessage ="Maximum 12 characters only")]

public string AccountNumber { get; set; }

[Column(TypeName ="nvarchar(100)")]

[DisplayName("Beneficiary Name")]

[Required(ErrorMessage = "This Field is required.")]

public string BeneficiaryName { get; set; }

[Column(TypeName ="nvarchar(100)")]

[DisplayName("Bank Name")]

[Required(ErrorMessage = "This Field is required.")]

public string BankName { get; set; }

[Column(TypeName ="nvarchar(11)")]

[DisplayName("SWIFT Code")]

[Required(ErrorMessage = "This Field is required.")]

[MaxLength(11)]

public string SWIFTCode { get; set; }

[DisplayName("Amount")]

[Required(ErrorMessage = "This Field is required.")]

public int Amount { get; set; }

[DisplayFormat(DataFormatString = "{0:MM/dd/yyyy}")]

public DateTime Date { get; set; }

}

C#Copy

Here we’ve defined model properties for the transaction with proper validation. Now let’s define  DbContextclass for EF Core.

#asp.net core article #asp.net core #add loading spinner in asp.net core #asp.net core crud without reloading #asp.net core jquery ajax form #asp.net core modal dialog #asp.net core mvc crud using jquery ajax #asp.net core mvc with jquery and ajax #asp.net core popup window #bootstrap modal popup in asp.net core mvc. bootstrap modal popup in asp.net core #delete and viewall in asp.net core #jquery ajax - insert #jquery ajax form post #modal popup dialog in asp.net core #no direct access action method #update #validation in modal popup