1.	SimpleBinding
	
	string Name, int Age
		{ Name: "Gina", Age: 11 }
		  'Name=Gina&Age=12'
		  new jqdata("Gina",13)

2.	ModelBindObj Human data	{ Name: "Gina", Age: 21 } 'Name=Gina&Age=22' { data: { Name: "Gina", Age: 23 } } { data: new jqdata('Gina', 24) } new jqdata('Gina',25)
3.	SimpleModelBindArray string[] Name, int[] Age	{ Name: ["Gina", "Tom"], Age: [31, 1] } new jqdata(["Gina","Tom"],[32,2])
4.	ModelBindingArray Human[] data	{ data: [{ Name: "Gina", Age: 41 },{ Name: "Tom", Age: 1 }] } [{ Name: "Gina", Age: 42 }, { Name: "Tom", Age: 1 }] { data: [new jqdata("Gina", 43), new jqdata("Tom", 1)] } [new jqdata("Gina", 43), new jqdata("Tom", 1)]
5.	ModelBindingNestedObj Person data	{ data: { Name: "Gina", Age: 51, Friends: [{ Name: "Nana", Age: 0, Friends: [{ Name: "Ivy", Age: 3 }] }] } } { Name: "Gina", Age: 52, Friends: [{ Name: "Nana", Age: 0, Friends: [{ Name: "Ivy", Age: 3 }] }] } { data: new jqdata('Gina', 53, [new jqdata('Tina', 20, null)]) } new jqdata('Gina', 54, [new jqdata('Tina', 20, null)])
6.	ModelBindingArrayNestedObj Person[] data	{ data: [{ Name: "Gina", Age: 61, Friends: [{ Name: "Nana", Age: 0, Friends: [{ Name: "Ivy", Age: 3 }] }] }, { Name: "Fanny", Age: 5, Friends: [{ Name: "Tina", Age: 0, Friends: [{ Name: "Chris", Age: 3 }] }] }] } [{ Name: "Gina", Age: 62, Friends: [{ Name: "Nana", Age: 0, Friends: [{ Name: "Ivy", Age: 3 }] }] }, { Name: "Fanny", Age: 5, Friends: [{ Name: "Tina", Age: 0, Friends: [{ Name: "Chris", Age: 3 }] }] }] { data: [new jqdata('Gina', 63, [new jqdata('Ivy', 2)]), new jqdata('Fanny', 23, [new jqdata('Eva', 2)])] } [new jqdata('Gina', 63, [new jqdata('Ivy', 2)]), new jqdata('Fanny', 23, [new jqdata('Eva', 2)])]

- JavaScripts的function

function jqdata(name, age,friends=null)

    {

        this.name = name

        this.age = age

        this.Friends = friends||null

    }

- 主要觀念就是Key:Value，將前端傳輸值的名稱及資料型別對得上後端Action接收的資料型別的屬性。

- 在**application/x-www-form-urlencoded**之下，非字串資料若不是用{KEY:VALUE}方式傳輸，會傳輸失敗。

- Ajax:會自動轉型為字串傳輸，配合**application/x-www-form-urlencoded**的編碼方式。

  要用<mark>application:JSON編碼方式</mark>傳輸，要先將字串JSON.**stringfy**，並設定**processData: false**，才不會再將字串再序列化一次，

  才能看到中間傳輸值正確型式。