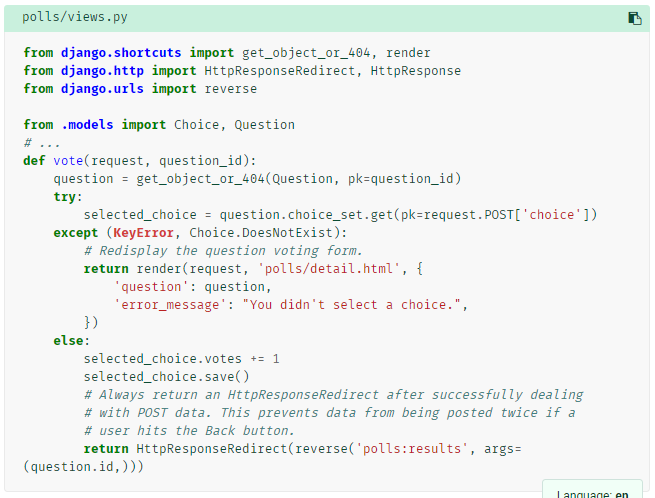
**Django app, part 4**



更改detail的html檔，新增一個post方法的form在網頁上，在form中加入{% csrf\_token %}，這是為了避免被跨站請求攻擊，因此使用django所提供的這項功能。

forloop.counter語法可測得迴圈已跑的次數，應用在這便可知道目前跑到的id為多少。

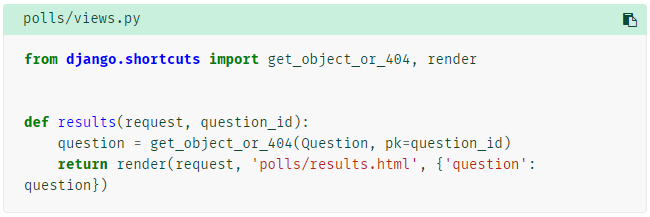


在view.py中修改vote函式，從http引入的功能改為HttpResponse及HttpResponseRedirect，urls引入reverse。

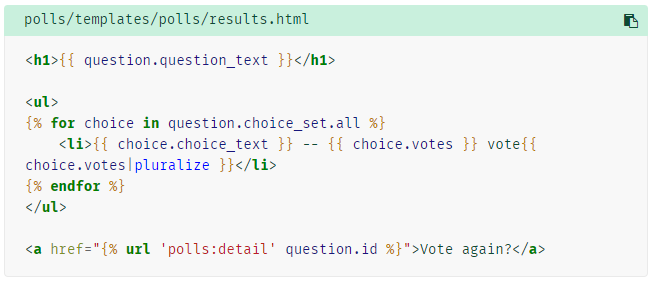
指定selected\_choice為從question對應的choice取得pk=被選擇的choice所回傳的id(request.POST值皆為字串型態)。

若無選擇choice則會呈現錯誤訊息(在render中有設定錯誤訊息為”You didn’t select a choice.”)。

使用HttpResponseRedirect中的reverse()功能可以避免當url改變時，還需在view中做更動，使用方法為第一項放所設定的URLconf，使其回傳對應的url，第二項放的是對應的值，如上為回傳question的id。



修改view.py，以之前在part3對detail修改的步驟修改results函式。



在templates/polls下新增results.html檔，改成上述的樣子。

pluralize此為設定，當值非為1時則回傳s，如：1 vote、2 votes、3 votes…如此，也可自行設定為pluralize: “es” or pluralize: “y, ies”不同的形式。



在urls.py中修改urlpatterns，改成generic view的形式，因為有一些呈現方式是有一套規則的，django中已有提供，如果套用的話可以少寫一些程式碼。



在views.py中從views引入generic功能，更改Index、Detail及Results三項，vote並不需要做變動。這裡使用到兩個generic view：ListView(呈現list的物件)、DetailView(呈現物件的detail的頁面)

使用generic view需要指定參數，model為告訴django要取得的模型為何，template\_name為指定要使用的template，context\_object\_name為取得被指定模型的資料的變數名稱。