

@ReadOnly only work with JS remoting. If you direct call the method using class instance then it wouldn't work

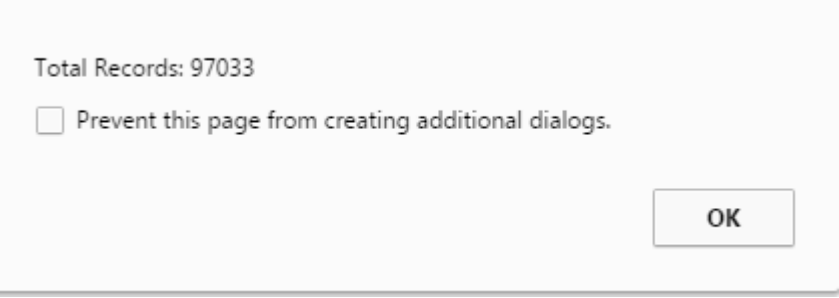
Here is the example(IN this example I have 97000 Task records.)

```
<apex:page controller="TestVFpage">

    <div id="totalAccounts"></div>
    <script>
        Visualforce.remoting.Manager.invokeAction(
            '{$RemoteAction.TestVFpage.getTask}',
            function(result){
                alert('Total Records: '+result.length);
            }
        );
    </script>
</apex:page>
```

Controller

```
public class TestVFpage {
    @RemoteAction
    @readOnly
    public static list<Task> getTask(){
        return [SELECT Id FROM Task];
    }
}
```



alert message

Another way to count records.

```
<apex:page>
    <input name="getList" class="btn" value="Get Account list" type="button"
onclick="getSubjectList()" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
</script>
    <script>
        var $j = jQuery.noConflict();
        var sessionId = '{$Api.Session_ID}';
        function getSubjectList() {
            $j.ajax({
                type: "GET",
                url: "/services/data/v35.0/query?q="+'+'+'{!URLENCODE('SELECT count(Id) totalTask
FROM Task'}}",
                headers: {"Authorization" : "OAuth " + sessionId},
                crossDomain : true,
                dataType: 'application/json',
                success: function (responseData) {
                    console.log(responseData);
                },
                error: function (request, status, error) {
                    console.log(request.responseText);
                }
            });
        }
    </script>
</apex:page>
```

check browser console

