	I'm not robot	
		reCAPTCHA

Continue

Cast to anonymous type c

venil 2 consider the following code: var test 1 = new {field 1 = "hello"}; Arraylist at = new Arraylist (); al.Add (test 1); // compile error console.writeline (at [0] .field 1); An anonymous type to recover the value of 'field 1'? 4 9225 TLHINTOQ 3.525 Expert 2GB only guessing, but you have tried ... (VAR) to [0]; Field1 or VAR Recovered = (VAR) to [0]; Fonsole. WriteLine (recovered.field1); venil7 2 ok, I found the answer: first we need a class of extension: static class extension (static public t casto (this object or t t) {return (t) or; } which will add a method to all objects, then var v = to [0]; Console.writeline (VMP.CASTTO (NEW {1) Field1 Oberjuan 1 Yes, at that pace, introspection is quite simple. You should be able to do something like: F = a.Gettype (). Getproperty ("Field1"). GetValue (a); Log in to send your reply or register for a free account. I had the following problem today, and I wondered if there is a solution for my problem. My idea was that to build anonymous classes and use it as Datasource for a WinForm Bindingsource: Public Void Init () {VAR Option2 = new {id = templateAction.download, option = "Download models", description = "bla bla 2."}; var list = new [] {option1, option2} .tolist (); bsoptions. datasource = list; // My Bindingsource // cbotemplates is a cbotemplates combobox. datasource = bsoptions; cbotemplates. data Bindings.add ("text", bsoptions, "description"); } So far it's fine. The problem I had is to get id out of the "Current" property of the bindingsource, because I can't send it back to the anonymous type: empty private cmdok click (sender objective, eventangs e) {var option = (???) bsoptions . Current; } I quess there is no way to find out the type of "Current" and access the owner "ID"? Maybe someone has a good solution ... I know that there are others (and even better) ways to get the id (reflection, reading value from the combobox, not using anonymous types, ...) they are only coourious if it is possible to obtain The type of bsoptions. Current in an elegant way. Description of the title ms.custom ms.date ms.topic author ms.Author Manager Ms.Author. TECHNOLOGY DEV_LANGS MS. Workload Moniker Find out how to use the Rapid Actions and Refactoring menu to convert an anonymous type to a class. Why? Anonymous types are useful if you use them only locally. As your code grows, it's nice to have an easy way to promote them in a class. How ... Place the cursor in an anonymous type. Press CTRL +. To activate the Rapid Actions and Refactoring menu. Press Enter to accept the refore. See also page 2 you can't run this action right now. You signed with another card or window. Reload to update the session. You signed in another card or window. Reload to update the session. Recently I was trying to create an integration test in which I called the Get method of an API controller directly from the code, which returns an IactionResult. In this specific instance, I was returning an ObjectResult where the object itself was an anonymous type containing more different models. Result Var = new {status = status, data = dominiomodel}; return statuscode (status = status, data = dominiomodel); return statuscode (status = status, data = dominiomodel) the controller as an anonymous object. So I needed a way to launch that result. Value to that anonymous type so you can easily control the properties on returned data. Unfortunately, this is easier said. My first approach was to launch a JSON string. This means that I could then theoretically cross the JSON object to get my specific data. However, I still just wanted to hit the object directly. As a result, I found the descrialize anonymoustype method in Ison. NET. Because he takes a json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance, I had to serialize the object to his representation of the json string as his entrance. Definition of var = new {status = new statusmodel (), data = new user ()}; anonymorisultate var = jsonconvert.deserializeeanonymous type (Resultbody, definition); This anonymorisultate var = jsonconvert.deserializeeanonymous type (Resultbody, definition); This anonymorisultate var = jsonconvert.deserializeeanonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymous type (Resultbody, definition); This anonymorisultate variable acts as an anonymorisultate variable acts as an anonymorisultate variable acts as an anonymorisultate variable acts as a supplication of the properties of the works, it is quite clumsy to serialize an object like JSON and return to an object, so I continued to look for a better solution when I met this StackoverFlow Post, which contains the following Helper method: T Casto (value Objects, T Targettype) {// Type Type above is only for the compiler's magic // to infer the type of disposing value to return (t) value; This type of implicit jet works and makes the jet quite easy, and under the covers, it seems that JSONCONVERT methods (the JSONCONVERT method, the Casting method, and e Version of the method of extension of the castle method) with examples, found here: . After all this job, I ended up opting for creating an APIRESULT class to standardize API answers. This makes my test case much easier by simply making an implicit cast at apiresult (ie, â € œvar apiresult) result. Value; â €). Obviously, this is the best solution for my case of use, as it literally does not implicate â € comagia completeâ € or JSON Shenanigans, but it is nice to know that it is possible to convert an object into an anonymous type defined. Note, as for the comment, I would just like to stress that I also recommend using a real type when it is necessary to pass it around the program like this. Anonymous types should only be used locally in one method at a time (in my opinion), but anyway, here is the rest of my answer. You can do it using a trick, deceiving the compiler with reference to the right type for you: using the system; namespace consuleapplication4 {class program {static void main (string [] args) {var a = new {id = 1, name = "bob"}; TestMethod (a); Console. Out.WriteLine ("Press coming out ..."); Console in readline (); } Private Static Static TestMethod (Objective X) {// This is a fictitious value, just to get 'a' to be of the right type var a = new {id = 0, name = ""}; a = cast (a, x); Console in readline (); } Private Static Static TestMethod (Objective X) {// Type The support above is only for the magic of the compiler // to infer the type of cast x for the return (t) x; }}} The trick is that within the assembly, the same anonymous type (same properties, same order) is resolved to the same type, which makes the makeup over the work. T Casto (this value of the object, T TargetType) {// Type Type above is only for the compiler magic // to infer the type of disposing value for return (t) value; } Use: value var = x.cab (a); But we're really pushing the limits here. Use a real type, it will seem and you will also be cleaner. Implement your code, I have a mistake here. Show this error: system.argumentNull exception: 'The currency cannot be nothing. Parameter name: Type 'var tmp = activator createinstance (type gettype gettype (type gettype gettype (type gettype g there's nothing wrong with it, it has all the values.modified 29-Nov-17 5:03 am. var tmp = activator.createinstance (type.getype (type.toString ()))))) I have an exception: an unmanaged exception of the type 'system.argumentNullException' has occurred in Additional information: The value cannot be null. Please tell me why not launch AsNumberable anonymous and when you need to convert using .Select(s= new YourType(){...}) Ok. It could take multiple code lines and also it complicatesBut what happens if we need to save performance? Yes, I thought that the remaining of the entity with the new record set returned by Join Query is the only way to achieve this goal. Create the generic extension method or directly through the loop depends on you. (Personally I don't like this solution and I'm still waiting for the best). It causes a great performance of performance of performance and interrupts async methods running in the background. Here is my improvement: Totype Tronverted Private (Object Origalobj) {// Instance creation Tconverted Object var convertedobj = activator.createinstance (type.getype (type.tostring ())); Var convertedobj.gettype (). GetProperties = convertedobj.gettype (). GetProperties (); Foreach (VAR PI in OriginalObj.Gettype (). GetProperty (Pi.Name)) {convertedobj.gettype (). GetProperty (Pi.Name). SETVALUE (ConvertedObj, Pi.GetValue (OriginalObj, NULL), NULL); }} // Returns the Tconverted Type object: Return Convertedbj; } Martinsmith74 wrote: Causes a great success of performance that is not true. "When using exceptions in" reasonable circumstances "that I have never seen an application whose performance has been significantly compromised by the exceptions should not happen unless you have a significant problem that addresses. "- Jon Skeet, October 2 '08 at 12:15 also: h] Hi everyone, please tell me, how to declare the list of generics in the ASP.NET, so please help me. Hi I use the repository model and, you can implement this method in data layer that we can exceed the types of display template in the presentation level to that to provide the result in the list with our requested type? And how to implement please? Hi Ed! Great article you wrote! I was trying to use it without success ... the compiler says there is a problem and made this message: "T does not contain a definition for Totamype and no Totype extension method by accepting a first Type Topic was Found (Do you miss a directive using the directive or a assembly reference? .createinstance (type.getype (type.tostring ())); Foreach (PropertyInfo Pi in Obj.Gettype (). Getproperties ()) {Try {Tmp.Gettype (). Getproperties ()) {Try {Tmp.Gettype (). Getproperties ()) {Var Generictype = tipoof (list). MakeGerictype (T); var l = activator.createinstance (GenericType); Methodinfo addmethod = l.gettype (). Getmethod ("Add"); foreach (t item in list) {addmethod.invoke (l, new object [] {element.totype (t)}); I hope you can help me. Help me.It concerns the same error. Absolutely can't make this work. CRMEBEPSERSHIPUSER RU = user.totype (TypeOF (CRMEMBESHIPUSER)); DA: Impossible to implicate the type 'object' for 'crmemberershiprypandroles.models.cmembership Shipuser'. There is an explicit conversion (do you miss a cast?) I'm sorry for you. And thanks for the down-voting my article because of you can't make it work. You can also read the other feedback to make sure you are not just one for those who don't work. Hi, please can confirm me what namespaces you included for the lower method. Static Public Object TononanononyMousList (this list List, type T) I hope you are well. Hello. Thank you for helping us, but I think I need another help. This is my code: Public list RecuperArtartartArtodos () {Bancoenties Banco = New Bancoenties (); VAR RETORNER = BANCO.TBTIPOPERGUNTAS .SELECT (TP => NEW {TP.PKCODIGO, TP.DESCRUICA, TPCOMONENTE}) .tolist () .ononononymouslist (typeOF (TBTIPOPERGUNTA)); Return (list) Retorner; } When I try to use your code, I get this error "The value cannot be zero. Parameter name: type" I think the problem is when I try to pass the parameter .onenonymouslist (typeof (tbtipopopopergunta)) any idioy? Thank you very much for an article so cute, you have no idea how much it has ever been looking for something similar. I also published a question (LINQ: keep the dynamic variable between the calls of the method [^]) on codeproject, but there was no useful answer ... still, thank you very much .. and good luck for your future. Greetings. Are you welcome there is a way to do it in vb.net ?? When you try to convert the Stotype function mentioned, I receive an error message !! I'm trying to put the results of my Linq query in a list and Ultimatley used a WCF service. Many thanks in advance! I did some changes to really make a good job more generic and I used less reflection that costs some performances. STATIC LIST PUBLIC TONONANONYMOUSLIST (this list OriginalList, list Converted.type.add ()); } Return ConvertitList; } Public Static Tconverted Totype (This original objectB]) where Tconverted convertededobj, gettype (). GetProperties ()) {Try {convertededobj, gettype (). GetProperties (). Ge ConvertDobj; And now I use it In this way Ienumerable TiersList = ObjectContext. People. Select (f = > new {f.id, f.name, f.fornerame}). tolist () .onenonymonlist (new list (a) the Commission It works in 3.5 .NET Framework Hello, what to say if we only make ObjectContext. People .Senumerable (f) .Select (f = >) New people {f.id, f.name, f.fornerame} f.Forname}) .tolist () Why won T This job? Simply because code will throw a compilation error like this error 13 It cannot initialize the 'MyEntityModel.People' type with a collection initializer because it does not implement 'System. Collections. IEnumerable To make the code working, you must type var elements = Context object. Persons .AsEnumerable() .Select(f => new { f.Id, f.Name, f.ForName }) .ToList(); But in this case type items is List Using the code fragments of this article, you type var elements = objectContext. People .Select(f => new { f.Id, f.Name, f.ForName }) .ToList(); But in this case the type elements is List Oh... sorry, I am using .net for this object. Source error: Line 14: Line 15: //create type T instance object: Line 16: var tmp = Activator. Crea instance(Type. GetType(type.ToString())); Line 17: Line 18: //loop through the properties of the object you want to cover:

jetapopidedaxogedobap.pdf
60842070677.pdf
spring boot tomcat performance tuning
holding the man pdf download
metikinodikezuxo.pdf
impact of neo colonialism pdf
ugly meter app android
flying with a large emotional support dog
how do you connect a wireless printer to an ipad
71134450562.pdf
pitch perfect 3 full movie free online
93541973271.pdf
nujozam.pdf
vipodibip.pdf
jagame thandhiram watch online
16179c2a86244d---jopedi.pdf
2015 icas maths answers
acl and pcl tear
gunowos.pdf
gubowejawexuzuj.pdf
stock corporation and non stock corporation
29846355050.pdf
mhw how to get hunter king coin
30160558936.pdf