

Syntaxe des expressions de requête pour les opérateurs de requête standard

Visual Studio 2012

Certains des opérateurs de requête standard les plus courants possèdent une syntaxe de mots clés du langage C# et Visual Basic dédiés qui leur permet d'être appelés dans le cadre d'une expression de requête. Une expression de requête est une forme différente d'expression d'une requête plus lisible que son équivalent fondé sur une méthode. Les clauses d'expression de requête sont traduites en appels aux méthodes de requête lors de la compilation.

[Tableau de la syntaxe des expressions de requête](#)

Le tableau suivant répertorie les opérateurs de requête standard qui comportent des clauses d'expression de requête équivalentes. Les langages de programmation C# et Visual Basic ne fournissent pas de syntaxe d'expression de requête dédiée pour les mêmes méthodes. Le tableau liste la syntaxe de ces deux langages.

Méthode	Syntaxe d'expression de requête C#	Syntaxe d'expression de requête Visual Basic
All<TSource>	non applicable	Aggregate ... In ... Into All(...) (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).
Any	non applicable	Aggregate ... In ... Into Any() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).
Average	non applicable	Aggregate ... In ... Into Average() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).

Cast<TResult>	<p>Utilisez une variable de portée explicitement typée, par exemple :</p> <p>from int i in numbers</p> <p>(Pour plus d'informations, consultez from, clause (Référence C#).)</p>	<p>From ... As ...</p> <p>(Pour plus d'informations, consultez From, clause (Visual Basic)).</p>
Count	non applicable	<p>Aggregate ... In ... Into Count()</p> <p>(Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).</p>
Distinct<TSource>(IEnumerable<TSource>)	non applicable	<p>Distinct</p> <p>(Pour plus d'informations, consultez Distinct, clause (Visual Basic)).</p>
GroupBy	<p>group ... by</p> <p>ou</p> <p>group ... by ... into ...</p> <p>(Pour plus d'informations, consultez group, clause (Référence C#).)</p>	<p>Group ... By ... Into ...</p> <p>(Pour plus d'informations, consultez Group By, clause (Visual Basic)).</p>
GroupJoin<TOuter, TInner, TKey, TResult>(IEnumerable<TOuter>, IEnumerable<TInner>, Func<TOuter, TKey>, Func<TInner, TKey>, Func<TOuter, IEnumerable<TInner>, TResult>)	<p>join ... in ... on ... equals ... into ...</p> <p>(Pour plus d'informations, consultez join, clause</p>	<p>Group Join ... In ... On ...</p> <p>(Pour plus d'informations, consultez Group Join, clause</p>

	(Référence C#) .	(Visual Basic) .
Join<TOuter, TInner, TKey, TResult>(IEnumerable<TOuter>, IEnumerable<TInner>, Func<TOuter, TKey>, Func<TInner, TKey>, Func<TOuter, TInner, TResult>)	join ... in ... on ... equals ... (Pour plus d'informations, consultez join, clause (Référence C#)).	From x In ..., y In ... Where x.a = b.a ou Join ... [As ...] In ... On ... (Pour plus d'informations, consultez Join, clause (Visual Basic)).
LongCount	non applicable	Aggregate ... In ... Into LongCount() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).
Max	non applicable	Aggregate ... In ... Into Max() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).
Min	non applicable	Aggregate ... In ... Into Min() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic)).
OrderBy<TSource, TKey>(IEnumerable<TSource>, Func<TSource, TKey>)	orderby (Pour plus d'informations, consultez orderby, clause (Référence C#)).	Order By (Pour plus d'informations, consultez Order By, clause (Visual Basic)).
OrderByDescending<TSource,	orderby ...	Order By ...

TKey>(IEnumerable<TSource>, Func<TSource, TKey>)	descending (Pour plus d'informations, consultez orderby, clause (Référence C#) .)	Descending (Pour plus d'informations, consultez Order By, clause (Visual Basic) .)
Select	select (Pour plus d'informations, consultez select, clause (Référence C#) .)	Select (Pour plus d'informations, consultez Select, clause (Visual Basic) .)
SelectMany	Plusieurs clauses from. (Pour plus d'informations, consultez from, clause (Référence C#) .)	Plusieurs clauses From (Pour plus d'informations, consultez From, clause (Visual Basic) .)
Skip<TSource>	non applicable	Skip (Pour plus d'informations, consultez Skip, clause (Visual Basic) .)
SkipWhile	non applicable	Skip While (Pour plus d'informations, consultez Skip While, clause (Visual Basic) .)
Sum	non applicable	Aggregate ... In ... Into Sum() (Pour plus d'informations, consultez Aggregate, clause (Visual Basic) .)
Take<TSource>	non applicable	Take (Pour plus d'informations,

		consultez Take, clause (Visual Basic)).
TakeWhile	non applicable	Take While (Pour plus d'informations, consultez Take While, clause (Visual Basic)).
ThenBy<TSource, TKey>(IEnumerable<TSource>, Func<TSource, TKey>)	orderby ..., ... (Pour plus d'informations, consultez orderby, clause (Référence C#) .)	Order By ..., ... (Pour plus d'informations, consultez Order By, clause (Visual Basic) .)
ThenByDescending<TSource, TKey>(IEnumerable<TSource>, Func<TSource, TKey>)	orderby ..., ... descending (Pour plus d'informations, consultez orderby, clause (Référence C#) .)	Order By ..., ... Descending (Pour plus d'informations, consultez Order By, clause (Visual Basic) .)
Where	where (Pour plus d'informations, consultez where, clause (Référence C#) .)	Where (Pour plus d'informations, consultez Where, clause (Visual Basic) .)