

Set the specified public field value of an object

```
#region License and Copyright
```

```
/*
```

```
* Dotnet Commons Reflection
```

```
*
```

```
* This library is free software; you can redistribute it  
and/or modify it
```

```
* under the terms of the GNU Lesser General Public License as  
published by
```

```
* the Free Software Foundation; either version 2.1 of the  
License, or
```

```
* (at your option) any later version.
```

```
*
```

```
* This library is distributed in the hope that it will be  
useful, but
```

```
* WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY
```

```
* or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser  
General Public License
```

```
* for more details.
```

```
*
```

```
* You should have received a copy of the GNU Lesser General  
Public License
```

```
* along with this library; if not, write to the
```

```
* Free Software Foundation, Inc.,
```

```
* 59 Temple Place,
```

```
* Suite 330,
```

```
* Boston,
```

```
* MA 02111-1307
```

```
* USA
```

```
*
```

```
*/
```

```
#endregion
```

```
using System;  
using System.Collections;  
using System.IO;  
using System.Reflection;  
using System.Runtime.Serialization.Formatters.Binary;  
using System.Text;
```

```
namespace Dotnet.Commons.Reflection
```

```
{
```

```
/// _____
```

```
///
```

```
/// This utility class contains a rich sets of utility methods  
/// that perform operations
```

```
/// on objects during runtime such as copying of property and  
/// field values
```

```
/// between 2 objects, deep cloning of objects, etc.
```

```
///
```

```
/// _____
```

```
public abstract class ObjectUtils
```

```
{
```

```
/// _____
```

```
///
```

```
/// Set the specified public field value of an object
```

```
///
```

```
/// the Value Object on which setting is to be performed ///
```

```
Field name /// Value to be set /// _____
```

```
public static object SetField(object vo, string fieldName,  
Object fieldValue)
```

```
{
```

```
if (vo == null)
```

```
throw new System.ArgumentNullException("No object specified to  
set.");
```

```
if (fieldName == null)
```

```
throw new System.ArgumentNullException("No field name
```

```
specified.");
```

```
if ((fieldName == String.Empty) || (fieldName.Length < 1))  
throw new System.ArgumentException("Field name cannot be  
empty.");          FieldInfo          fieldInfo          =  
vo.GetType().GetField(fieldName); if (fieldInfo == null) throw  
new System.ArgumentException("The class '" + vo.GetType().Name  
+ "' does not have the field '"+ fieldName + "'"); // Set the  
value          vo.GetType().InvokeMember(fieldInfo.Name,  
BindingFlags.SetField, null, vo, new object[] {fieldValue});  
return vo; } } } [/csharp]
```