

C# Reflection

Where to use Reflection

- Normally compiler decision of types
(**strongly typed language**)
- Sometimes the need
The type information at runtime => reflection
- For generic framework for do something -
examples
 - Json convert
 - Entity framework

What is Reflection

Metatype

- `Type t = obj.GetType();`

Kind of information

- `t.IsXXX` (class, interface, abstract)
- `t.GetProperties()` => `PropertyInfo` (name, type etc)
- `t.GetMethods()` => `MethodInfo` (name, parameters, return types etc) => invoke methods
- `t.BaseType`

Call methods in Reflection

Example:

```
// find
```

```
MethodInfo setIdMethod =
```

```
    t.GetMethods().First(m => m.Name == "set_Id");
```

```
// make call
```

```
setIdMethod.Invoke(o, new object[] { 12 });
```

```
// parameters in an array of values (here only one)
```

Extension Methods

Example:

```
public static string XXMethodName(this ExtensionToType t)
```

Small Demo

Opgave MyJsonConverter .