

C# Reflection

Where to use Reflection

- Normally compiler decision of types
(**strongly typed language**)
- Sometimes we need the type information at runtime => reflection
- An examples Generic framework to do something
 - Json convert (e.g. Newtonsoft)
 - 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:

```
Type t = o.GetType();
```

```
// find
```

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

```
// make call
```

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

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

Small Demo

Exercise MyJsonConverter .