Lazy Load Pattern in C#



Filip Ekberg
PRINCIPAL CONSULTANT & CEO
@fekberg fekberg.com



Lazy Loading



Don't eagerly load data you won't use!



Different Flavors of Lazy Loading

Lazy Initialization

Virtual proxies

Value holders

Ghost objects



Lazy Initialization



Lazy Initialization

```
private byte[] profilePicture;

public byte[] ProfilePicture
{
    get {
        if (profilePicture == null) {
             profilePicture = ProfilePictureService.GetFor(Name);
        }
        return profilePicture;
    }
    set { profilePicture = value; }
}
```



Lazy Initialization

When the value is null we try to load the data

This requires the entity to know about accessing the services or databases



The entity is now coupled with logic to load additional data



Value Holders



Using a Value Holder



Virtual Proxies



Virtual Proxies

```
public class CustomerProxy : Customer
{
    public override byte[] ProfilePicture
    {
        get {
            if (base.ProfilePicture == null) {
                base.ProfilePicture = ProfilePictureService.GetFor(Name);
            }
            return base.ProfilePicture;
        }
    }
}
```



Using a Virtual Proxy

```
var customer = context.Customers
                      .Single(c => c.CustomerId == entity.CustomerId);
var proxy = new CustomerProxy
   Name = customer.Name,
   City = customer.City,
   PostalCode = customer.PostalCode,
   ShippingAddress = customer.ShippingAddress,
   Country = customer.Country,
return proxy;
```



Virtual Proxies

The repository can map the entity to a proxy class to return to its caller

This will allow the proxy to intercept calls to a property and load the data when necessary



Ghost Objects



Ghost Objects

```
public class CustomerGhost : Customer
   public override string Name {
        get {
            Load();
            return base.Name;
        set
            Load();
            base.Name = value;
```



Ghost Objects

The entity is loaded in a partial state which

It is fully loaded when a property is accessed





Only load necessary data

Different patterns depending on your situation

Lazy<T> provides great flexibility and is easy to use

Most ORMs like Entity Framework comes with lazy loading built in



Thank You!

