

## ARTIGO: 11379

### Usando Lazy Load

#### Introdução

Nesse artigo veremos como usar o Lazy Load do Velster Framework.

#### Como usar

Para usar o recurso de lazy load do Velster, nossa classe de domínio deve estender *br.com.mobilemind.velster.orm.model.EntityLazy*, e no método *get* do atributo que sofrerá o lazy, devemos chamar o método *lazy("attributeName")*, passando como parametro o atributo que sofre o lazy.

#### Java Code

```
public class EntityImpl extends EntityLazy {

    @Id()
    @Column()
    private Long id;
    private boolean loaded;

    @Override
    public Long getId() {
        return id;
    }

    @Override
    public void setId(Long id) {
        this.id = id;
    }

    @Override
    public boolean isLoaded() {
        return this.loaded;
    }

    @Override
    public void setLoaded(boolean loaded) {
        this.loaded = loaded;
    }

    @Override
    public boolean equals(Object o) {
        if (o instanceof EntityImpl && this.id != null) {
            return this.id.equals(((Entity) o).getId());
        }
        return false;
    }

    @Override
    public int hashCode() {
        int hash = 7;
        hash = 29 * hash + (this.id != null ? this.id.hashCode() : 0);
        return hash;
    }

    @Override
    public String toString() {
        return this.getClass().getName() + " {" + this.id + "}";
    }
}
```

#### Java Code

```
@Table
public class EntityInsertCascadeLazy extends EntityImpl {

    @Column
    private String name = "teste";
    @Column
    @JoinColumn( cascadeOnInsert = true)
    private PersonGroup grupo;

    public PersonGroup getGrupo() {
        lazy("grupo");
        return grupo;
    }

    public void setGrupo(PersonGroup grupo) {
        this.grupo = grupo;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```