

Introdução

Nesse artigo vamos ver as diretrizes que devem ser seguidas para a criação de uma entidade controlada pelo Veloster Framework.

Modelos

Toda entidade gerenciada pelo Veloster, além das anotações de identificação, devem seguir um padrão para criação, ou seja, elas devem implementar uma interface conhecida pelo framework ou estender uma classe com algumas funcionalidades pré-definidas.

Abaixo a interface modelo a ser seguida para a criação de uma entidade. Essa interface define algumas propriedades e métodos necessárias para o funcionamento do framework.

Java Code

```
br.com.mobilemind.veloster.orm.model.Entity<T>
```

Abaixo a implementação padrão para a interface padrão (uso recomendado):

Java Code

```
br.com.mobilemind.veloster.orm.model.EntityLazy<T extends Entity>
```

Essa classe abstrata já implementa as funcionalidades básicas, como identificador de carregamento e lazy load de atributos do tipo @JoinColumn. É recomendado que essa classe sempre seja usada como modelo para entidades.

Implementação Básica

Agora vamos construir uma implementação básica de um modelo para usarmos em nossas aplicações:

Java Code

```
import br.com.mobilemind.veloster.orm.model.Entity;
import br.com.mobilemind.veloster.orm.annotations.Column;
import br.com.mobilemind.veloster.orm.annotations.Id;
import br.com.mobilemind.veloster.orm.model.EntityLazy;

public class EntityImpl extends EntityLazy {

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

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

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

    @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 + "}";
    }
}
```

Para usarmos nossa implementação:

Java Code

```
import br.com.mobilemind.veloster.orm.annotations.Column;
import br.com.mobilemind.veloster.orm.annotations.Table;
```

```
/**
 *
 * @author Ricardo Bocchi
 */
@Table(name = "groups")
public class Group extends EntityImpl {

    @Column(length = 100)
    private String name;

    public String getName() {
        return name;
    }

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