



Fetch Type



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >



Fetch Type

Quando usamos qualquer associação (relacionamento) e a informação é recuperada do banco de dados e carregada em memória devemos especificar um Fetch Type.

Quando o Hibernate acessa uma entidade que está relacionada a outra por meio de associações, ao recuperar a informação, esse comportamento pode ser configurado usando esses tipos de FetchType:

- **EAGER**
- **LAZY**





```
@Entity
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String street;
    private int houseNumber;
    private String city;
    private int zipCode;
    @ManyToOne(fetch = FetchType.LAZY)
    private Person person;
}
```



Eager

Carrega os dados da entidade e de **todas as entidades que estão relacionadas.**





Eager Loading - Carregamento ansioso

No Eager Loading ou carregamento ansioso, os registros são carregados em memória, assim como os registros associados mapeados por meio de suas entidades são carregados realizando uma consulta.

No exemplo a seguir, podemos ver o carregamento ansioso configurado nas classes Carrinho e Itens:

```
@OneToMany(mappedBy = "carrinho", cascade = CascadeType.ALL, fetch  
= FetchType.EAGER)  
private Set <Item> items;
```





Resultados

No exemplo do carrinho e itens, quando fazemos uma pesquisa por id e ela está **configurada com carregamento ansioso**, podemos ver no console a seguinte saída:

```
{} Hibernate: select carrinho0_.id as id1_1_0_,  
itens1_.carrinho_id as carrinho_id4_4_1_, itens1_.id as  
id1_4_1_, itens1_.id as id1_4_2_, itens1_.quantidade as  
quantidade2_4_2_, itens1_.carrinho_id as carrinho_id4_4_2_,  
itens1_.descricao as descricao3_4_2_ from carrinho carrinho0_  
left outer join itens itens1_ on  
carrinho0_.id=itens1_.carrinho_id where carrinho0_.id=?
```



Lazy

Carrega os dados da entidade, mas **não carrega os dados das entidades que estão relacionadas.**





Lazy Loading - Carregamento preguiçoso

No Lazy Loading ou carregamento preguiçoso, os registros são carregados em memória, mas os registros associados mapeados por meio de suas entidades não são carregados até que ocorra uma chamada explícita ao método get.

No exemplo a seguir, podemos ver o carregamento preguiçoso configurado nas classes Carrinho e Itens:

```
@OneToMany(mappedBy = "carrinho", cascade = CascadeType.ALL, fetch  
= FetchType.LAZY)  
private Set <Item> items;
```




Resultados

No exemplo do carrinho e itens, quando fazemos uma pesquisa por id e ela está **configurada com carregamento preguiçoso**, podemos ver no console a seguinte saída:

```
Hibernate: select carrinho0_.id as id1_1_0_ from carrinho  
carrinho0_ where carrinho0_.id=?
```

```
{}
```

```
Hibernate: select itens0_.carrinho_id as carrinho_id4_4_0_,  
itens0_.id as id1_4_0_, itens0_.id as id1_4_1_, itens0_.quantidade  
as quantidade2_4_1_, itens0_.carrinho_id as carrinho_id4_4_1_,  
itens0_.descricao as descricao3_4_1_ from itens itens0_ where  
itens0_.carrinho_id=?
```



Eager vs Lazy

Eager Loading

Usar o Eager Loading em cenários onde ocorra o carregamento desnecessário de informações poderá gerar problemas de desempenho.



Lazy Loading

O Lazy Loading fornece tempo menores para carregamento e menor consumo de memória em comparação com o Eager Loading. O Lazy Loading é o padrão para @OneToMany e @ManyToMany.



DigitalHouse>