

Description: Coffee shop that has a employee and coffee

Supervisor:

```
@Pattern(regex = "(salles|cachear)")
```

String department

```
@NotNull(message = "name can't be empty")
@Size(min = 5, message = "name length must be more than 4 characters.")
@Pattern(regex = "([a-zA-Z]+)", message = "name should only contains alphabets characters")
```

String name

```
@NotNull(message = "id can't be empty")
@Size(min = 3, message = "id - length must be more than 2 characters.")
```

String id

```
@NotNull(message = "employees can't be empty")
```

ArrayList<Employee>employees

Employee:

```
@NotNull(message = "name can't be empty")
@Size(min = 5, message = "name length must be more than 4 characters.")
@Pattern(regex = "([a-zA-Z]+)", message = "name should only contains alphabets characters")
```

name

```
@NotNull(message = "id can't be empty")
@Size(min = 3, message = "id - length must be more than 2 characters.")
```

id

```
@Pattern(regex = "(salles|cachear)")
```

department

```
@Pattern(regex = "(0.) (\\d)+")
```

double salseBonuse

```
@Pattern(regex = "(0-9){2}")
@Min(18)
@Positive
```

Int age

Branches:

```
@NotNull(message = "address can't be empty")
@Pattern(regex = "([a-zA-Z0-9]{10,20})", message = "address should only contains alphabets characters and numbers")
```

address

Drinks:

```
@Pattern(regexp = "(hot|cold)")
```

coffeeType

```
Pattern(regexp = "(small|medium|large)")
```

size

Donets:

```
@Pattern(regexp = "(chocolate|sugar glaized)")
```

donetType