- If we want to implement auditing data when we create a new object or update a
  object, we can implement methods in the entity and create columns in the entity
  table.
- First, implement the attributes in the entity

```
@Id // the primary key of the table will be the attribute id from the client
@GeneratedValue(strategy = GenerationType.IDENTITY) // auto increment id (primary key)
private Long id;
private String name;
private String cpf;
private Double income;
private Instant birthDate;
private Integer children;

private Instant createdAt;
private Instant updatedAt:
```

Now, we have to put a annotation in the attributes to inform that we want store this
data in the UTC form.

```
@Id // the primary key of the table will be the attribute id from the client
@GeneratedValue(strategy = GenerationType.IDENTITY) // auto increment id (primary key)
private Long id;
private String name;
private String cpf;
private Double income;
private Instant birthDate;
private Integer children;

@Column(columnDefinition = "TIMESTAMP WITHOUT TIME ZONE")
private Instant createdAt;

@Column[columnDefinition = "TIMESTAMP WITHOUT TIME ZONE")
private Instant updatedAt;

public Client() {
}
```

• Now, we have to create the getters and setters methods. We will not create the setters methods, because we don't want to set methods for the auditing data, just get.

```
public void setChildren(Integer chldren) {
    this.children = chldren;
}

public Instant getCreatedAt() {
    return createdAt;
}

public Instant getUpdatedAt() {
    return updatedAt;
}
```

 To get the instant time when create or update, we will create a method to store the instant time.

```
public Instant getCreatedAt() {
    return createdAt;
}

public Instant getUpdatedAt() {
    return updatedAt;
}

public void prePersist() {
    createdAt = Instant.now();
}

public void preUpdate() {
    updatedAt = Instant.now();
}
```

• To make this method return the instant, we put a annotation.

```
public Instant getCreatedAt() {
    return createdAt;
}

public Instant getUpdatedAt() {
    return updatedAt;
}

@PrePersist
public void prePersist() {
    createdAt = Instant.now();
}

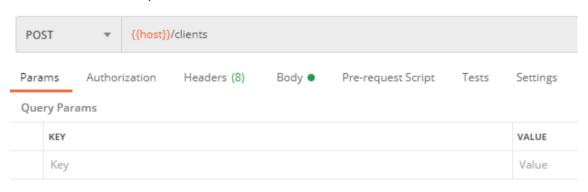
@PreUpdate
public void preUpdate() {
    updatedAt = Instant.now();
}
```

• We don't create the attributes in the DTO package, because we don't want to inform to the user the instant data.

• When we run the app, the tables was created



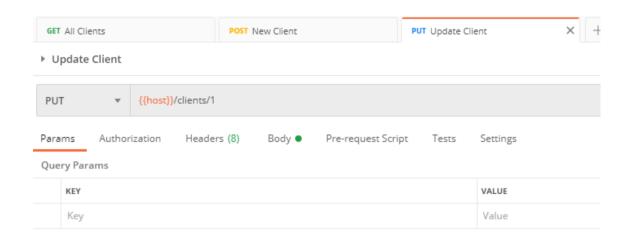
• Let's test the Update and Create







SELECT * FROM TB_CLIENT;											
ID	BIRTH_DATE	CHILDREN	CPF	CREATED_AT	INCOME	NAME	UPDATED_AT				
1	2011-12-03 08:15:30	1	111.111.111-11	null	1000.0	JOAO	null				
2	2011-12-03 08:15:30	2	222.222.222-22	null	2000.0	MARIA	null				
3	2011-12-03 08:15:30	3	333.333.333-33	null	3000.0	JOSE	null				
4	1994-07-20 07:30:00	2	12345678901	2021-01-24 18:54:26.806998	6500.0	Maria Silva	null				
(4 rows, 4 ms)											





SEL	SELECT * FROM TB_CLIENT;											
ID	BIRTH_DATE	CHILDREN	CPF	CREATED_AT	INCOME	NAME	UPDATED_AT					
1	1994-07-20 07:30:00	2	12345678901	null	6500.0	Maria Silvaaa	2021-01-24 18:55:53.081488					
2	2011-12-03 08:15:30	2	222.222.222-22	null	2000.0	MARIA	null					
3	2011-12-03 08:15:30	3	333.333.333-33	null	3000.0	JOSE	null					
4	1994-07-20 07:30:00	2	12345678901	2021-01-24 18:54:26.806998	6500.0	Maria Silva	null					
(4 rows, 1 ms)												