

# Bootcamp Crisalis

## Orange Team

Faccipieri, Federico

Fleitas, Gonzalo

Flores, Elio

Haoy, Sebastián

Pepa Degani, Julián



# Excepciones generales

Responsable: Pepa Degani, Julián

JIRA: [SCRUM-70](#) Backend

## Descripción

Lista de descuentos totales por cliente.

## Requerimientos

- Crear un stored procedure.
- Generar un back que se adapte al SP.
- Filtrar por fechas.

## Stored Procedure

Estructura del SP, recibe dos parámetros, fecha desde y fecha hasta, ambos de tipo varchar.

```

CREATE DEFINER='root'@'localhost' PROCEDURE `reportTotalDiscount`(IN from_date varchar(55), IN until_date varchar(55))
BEGIN
    DECLARE query_full TEXT;

    SET @SQL = "SELECT ce.business_name, ce.id,
ce.dtype, ce.first_name, ce.last_name, sg.name AS item_name, oe.date_created, od.discount FROM order_entity oe
INNER JOIN order_detail od ON oe.id = od.order_id
INNER JOIN client_entity ce ON ce.id = oe.client_id
INNER JOIN sellable_good sg ON od.sellable_good_id = sg.id
WHERE sg.type = 'SERVICE' ";
    #SET @SQL = CONCAT(@SQL, " AND (oe.date_created >= '", from_date, "' AND oe.date_created <= '", until_date, "' ) ");
    IF until_date != '' THEN
        SET @SQL = CONCAT(@SQL, " AND (oe.date_created >= '", from_date, "' AND oe.date_created <= '", until_date, "' ) ");
    ELSE
        SET @SQL = CONCAT(@SQL, " AND (oe.date_created >= '", from_date, "' ) ");
    END IF;
    SET @SQL = CONCAT(@SQL, ' ORDER BY ce.business_name , oe.date_created, od.discount DESC;');
    SET query_full = @SQL;
    SET @query_full = query_full;
    PREPARE STMT FROM @query_full;
    EXECUTE STMT;
END

```

## Clase ReportTotalDiscount

La clase ReportTotalDiscount es una representación del Stored Procedure, contiene atributos idénticos.

```

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "reportTotalDiscount") // <--- nombre del stored procedure en la base de datos
@Immutable
public class ReportTotalDiscount implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", updatable = false, nullable = false)
    private Long id;
    @Column
    private String business_name;
    @Column
    private String first_name;
    @Column
    private String last_name;
    @Column
    private String dtype;
    @Column
    private String item_name;
    @Column
    private Date date_created;
    @Column
    private Double discount;
}

```

## IReportRepository

El método llama al SP con sus respectivos parámetros.

```

public interface IReportRepository extends JpaRepository<ReportTotalDiscount, Long> {
    1 usage ↗ JulianPepaDegani97
    @Transactional
    @Procedure(name = "reportTotalDiscount(:fromDate, :untilDate)")
    List<ReportTotalDiscount> reportTotalDiscount(@Param("fromDate") String fromDate, @Param("untilDate") String untilDate);
}

```

## Controller

El controlador de Informes tiene como base la ruta /report. En /report/total-discount se trae la lista de descuentos totales por cliente.

```
@RestController
@RequestMapping("/report")
public class ReportController {

    1 usage
    @Autowired
    private IReportService reportService;

    no usages  JulianPepaDegani97
    @GetMapping("/total-discount")
    @Transactional(readOnly = false)
    public List<ReportTotalDiscount> getTotalDiscountPerClient(@RequestParam("fromDate") String fromDate,
                                                                @RequestParam("untilDate") String untilDate) {
        return reportService.totalDiscountReport(fromDate, untilDate);
    }
}
```