Bootcamp Crisalis

Orange Team

Faccipieri, Federico

Fleitas, Gonzalo

Flores, Elio

Haoys, Sebastián

Pepa Degani, Julián

Backend de Impuestos

Responsable: Pepa Degani, Julián

JIRA: SCRUM-31 Backend

Descripción

Creación de las entidades necesarias para el funcionamiento de los impuestos.

Desarrollo del CRUD de Tax con accesos limitados por roles.

Requerimientos

- Creación de la entidad Tax.
- Creación de la tabla intermedia ItemTax, entre Tax e Item.
- Relación ‘uno a muchos’ entre Tax e ItemTax.
- Creación de repositorio de Tax.
- Creación de dto de Tax.
- Desarrollo de servicio e implementación de Tax.
- Desarrollo de controlador de Tax.

Clase Tax

Tiene como atributos el nombre (taxName), el porcentaje (taxPercentage), isActive por defecto en true, sirve para dar de baja y reactivar. Y la relación ‘uno a muchos’ con ItemTaxes.

```java
public class Tax {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @NotNull
    private String taxName;

    @NotNull
    private Double taxPercentage;

    private boolean isActive = true;

    @OneToMany(mappedBy = "tax")
    @Getter(value = AccessLevel.NONE)
    private Set<ItemTax> itemTaxes;

    no usages  new *
    public Tax(String taxName, Double taxPercentage) {
        this.taxName = taxName;
        this.taxPercentage = taxPercentage;
    }
}
```

Clase ItemTax

Tabla intermedia entre Tax e Item.

```java
public class ItemTax {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "tax_id")
    private Tax tax;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "detail_id")
    private Item item;

    no usages  new *
    public ItemTax(Tax tax, Item item) {
        this.tax = tax;
        this.item = item;
    }
}
```

Controlador de Tax

Se toma como base la ruta /tax.

```java
@RestController
@RequestMapping("/tax")
public class TaxController {
```

Métodos GET del controlador

En la misma ruta de la base, el método GET trae la lista de impuestos. Accesible para usuarios con roles de user y/o admin.
Al agregar un ID a esta ruta, se hace la búsqueda de un impuesto en particular.

```java
@PreAuthorize("hasAnyRole('USER' ,'ADMIN')")
@GetMapping
public List<TaxDto> getTaxes() {
    List<TaxDto> taxList = taxService.getTaxes();
    return taxList;
}

no usages  new *
@PreAuthorize("hasAnyRole('USER' ,'ADMIN')")
@GetMapping("/{id}")
public TaxDto getTaxById(@PathVariable("id") Integer id) { return taxService.getTaxById(id); }
```

Método para crear impuesto

En la ruta /tax/create, solo los usuarios con rol admin pueden acceder. Se pasa un TaxDto y se adapta para convertirlo en Tax.

```
@PreAuthorize("hasRole('ADMIN')")
@Transactional
@PostMapping("/create")
public ResponseEntity<Object> createTax(@Valid @RequestBody TaxDto taxDto, BindingResult bindingResult) {
    if(bindingResult.hasErrors()) {
        return new ResponseEntity<Object>( body: "Hubo un error en el envío", HttpStatus.BAD_REQUEST);
    }
    taxService.saveTax(taxDto);
    return new ResponseEntity<>( body: "Creado con éxito", HttpStatus.OK);
}
```

Método para actualizar impuesto

Ruta /tax/update/{id}, solo con permiso de admin..

```
@PreAuthorize("hasRole('ADMIN')")
@Transactional
@PutMapping("/update/{id}")
public ResponseEntity<Object> updateTax(@RequestBody TaxDto taxDto, @PathVariable("id") Integer id, BindingResult bindingResult) {
    if(bindingResult.hasErrors()) {
        return new ResponseEntity<Object>( body: "Hubo un error en el envío", HttpStatus.BAD_REQUEST);
    }
    taxService.updateTax(taxDto, id);
    return new ResponseEntity<>( body: "Creado con éxito", HttpStatus.OK);
}
```

Método para eliminar impuesto

Ruta /tax/delete/{id}, solo con acceso de admin.
Se hace un borrado lógico, se cambia el atributo isActive de Tax, a false.

```
@PreAuthorize("hasAnyRole('ADMIN')")
@PatchMapping("/delete/{id}")
public ResponseEntity<Object> deleteTax(@PathVariable("id") Integer id) {
    taxService.deleteTax(id);
    return new ResponseEntity<>( body: "Impuesto eliminado", HttpStatus.OK);
}
```