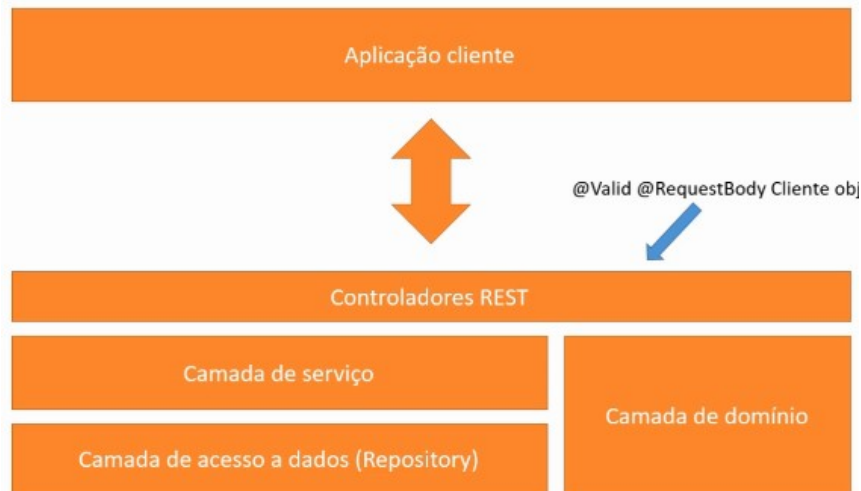


## 36. Validação sintática com Bean Validation

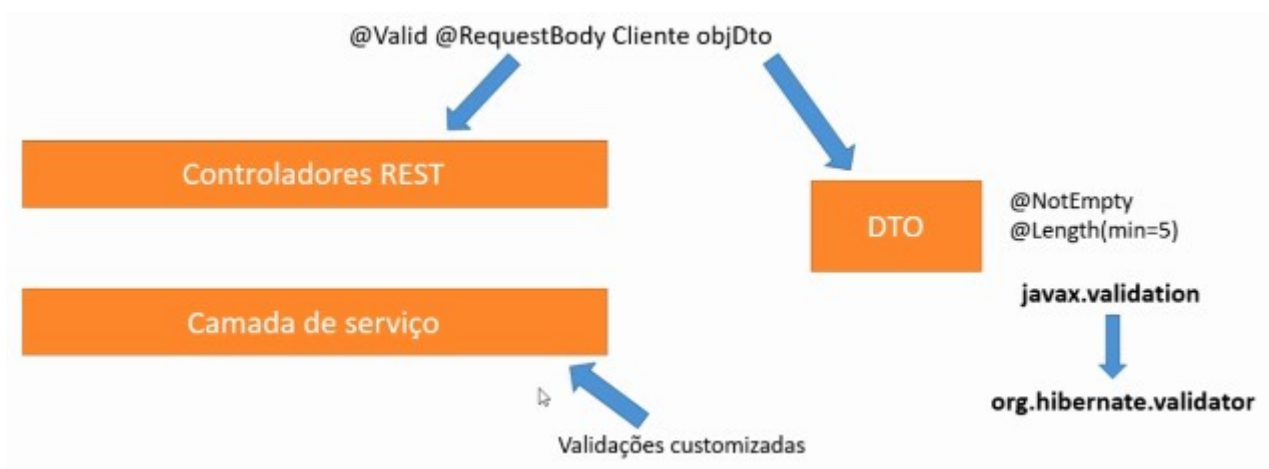
Na camada resource (Controladores REST) o framework intercepta o objeto DTO enviado pela requisição (POST ou PUT) para realizar sua validação antes que ele atinja a camada de serviço. Prezando pela prática de manter os Controladores REST simples, qualquer validação customizada será implementada em um pacote a parte.



O controlador REST recebe a anotação `@Valid` para interceptar a requisição e validar os dados do objeto DTO.

As regras de validação estarão nos próprios DTOs. Os campos que serão validados recebem anotações para serem validados.

Validações não contempladas por `javax.validation` e `hibernate.validator` que precisam acessar dados e verificar coisas mais complexas, são consideradas como serviços, logo são implementadas na camada de serviço.



### Validações sintáticas na entidade Categoria:

- nome não pode ser vazio.
- nome deve conter entre 5 e 80 caracteres.

Em **CategoriaDTO** inserir as anotações de validação no campo de interesse, neste caso o campo nome.

```
@NotEmpty(message = "Nome não pode estar vazio!")
@Length(min = 5, max = 80, message = "Nome deve ter pelo menos 5 caracteres e no máximo 80!")
private String nome;
```

Em **CategoriaResource** método insert( ) deverá ser atualizado para receber um parâmetro do tipo CategoriaDTO enquanto o atual é do tipo Categoria.

A atualização torna necessário converter o parâmetro CategoriaDTO no corpo do método em uma Categoria, para isso o método fromDto( ) é invocado de CategoriaService.

Também é preciso adicionar a anotação @Valid no parâmetro.

Antes...

```
@RequestMapping(method = RequestMethod.POST)
public ResponseEntity<Void> insert(@RequestBody Categoria obj) {
    obj = categoriaService.insert(obj);
    URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(obj.getId());
    return ResponseEntity.created(uri).build();
}
```

Depois...

```
@RequestMapping(method = RequestMethod.POST)
public ResponseEntity<Void> insert(@Valid @RequestBody CategoriaDTO objDto) {
    Categoria obj = categoriaService.fromDto(objDto);
    obj = categoriaService.insert(obj);
    URI uri = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(obj.getId());
    return ResponseEntity.created(uri).build();
}
```

Em **CategoriaService** criar um método auxiliar para converter um objeto CategoriaDTO em Categoria. O método recebe uma CategoriaDTO e retorna uma Categoria que recebe os dados de CategoriaDTO.

```
public Categoria fromDto(CategoriaDTO objDto) {
    return new Categoria(objDto.getId(), objDto.getNome());
}
```

Links:

<https://docs.oracle.com/javaee/7/tutorial/bean-validation001.htm>

<https://docs.jboss.org/hibernate/validator/4.1/api/org/hibernate/validator/constraints/package-summary.html>