

02

## Servindo os produtos

Nosso cliente ainda não consegue consumir os produtos, na verdade nem a gente consegue sem acessar diretamente o banco de dados! Vamos criar então uma rota para servir os produtos?

```
GET /api/produtos controllers.ApiController.todosOsProdutos
```

Como faremos um ponto de acesso, a lógica ficará em um novo controller.

```
public class ApiController extends Controller {
    @Inject private ProdutoDAO produtoDAO;
    public Result todosOsProdutos() {
        List<Produto> produtos = produtoDAO.todos();
        return ok(produtos.toString());
    }
}
```

Com a lógica do controller feita, podemos criar o método `todos()` no **ProdutoDAO** que retorna todos os produtos do banco!

```
public List<Produto> todos() {
    return produtos.all();
}
```

Mas a lista convertida para String é ilegível e portanto inútil. Não dá pra fazer nada com esses dados. Vamos converter então para um formato comum em APIs, o formato JSON, tem suporte embutido no Play! com a biblioteca Jackson. Basta fazer a conversão para JSON!

```
public Result todosOsProdutos() {
    List<Produto> produtos = produtoDAO.todos();
    return ok(Json.toJson(produtos));
}
```

Agora sim estamos servindo os produtos de um modo útil! Porém, se não houver nenhum produto no banco, o sistema retorna `[]`, que não representa nada sozinho. Faremos assim então: entregar pro cliente um envelope onde está escrito "produtos", com uma lista de produtos dentro. Então vamos modelar esse envelope?

```
package models;
public class EnvelopeDeProdutos {
    private List<Produto> produtos;
    public EnvelopeDeProdutos (List<Produto> produtos) {
        this.produtos = produtos;
    }
    public List<Produto> getProdutos() {
        return this.produtos;
    }
}
```

```
    }  
}
```

Agora basta inserir os produtos no envelope antes da conversão!

```
EnvelopeDeProdutos envelope = new EnvelopeDeProdutos(productoDAO.todos());  
return ok(Json.toJson(envelope));
```

E pronto, servimos um envelope bonitinho, rotulado e com uma lista de produtos dentro. O cliente sabe que ele recebeu uma lista de produtos, mesmo que esta esteja vazia!