@fpftech.educacional

FPFtech
Escola Tecnológica

Requisições Simuladas e Manipulação de JSON

Objetivo: Simular consumo de API usando Gson, converter JSON para objetos Kotlin, e exibir os dados com Jetpack Compose.

JSON: Formato leve para trocar dados entre sistemas. Muito usado em APIs e apps móveis. Estrutura: chave-valor

Gson no Kotlin: É uma biblioteca que transforma: JSON → Objeto Kotlin | Objeto Kotlin → JSON

Atenção:

A estrutura JSON **precisa bater** com a data class.

```
val json = """{"nome":"Ana", "idade":28}"""
val cliente = gson.fromJson(json, Cliente::class.java)
```

```
val novoJson = gson.toJson(cliente)
```

Campos nulos ou extras devem ser tratados com atenção.

Para usar Gson no projeto

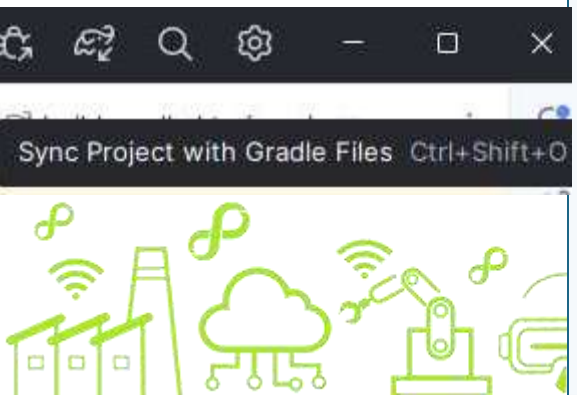
1 – vá para **build.gradle.kts (Module) > dependencies**

Acrescente

a linha de

implementação.

Rebuild gradle.



Para usar Gson no projeto

2 – Depois de instalar a dependência, basta importar a classe no seu código:

```
import com.google.gson.Gson
```

Agora você já pode usar:

```
val gson = Gson()

val cliente = gson.fromJson(json, Cliente::class.java)
```

Integração com Compose

Usando dados dinâmicos em @**Composable:**

```
@Composable
fun ExibirCliente(clienteJson: String) {
    val gson = remember { Gson() }
    val cliente = gson.fromJson(clienteJson, Cliente::class.java)

    Column(modifier = Modifier.padding(16.dp)) {
        Text("Nome: ${cliente.nome}")
        Text("Idade: ${cliente.idade}")
    }
}
```

Remember: objeto será criado apenas uma vez, e nas recomposições seguintes, a mesma instância será reutilizada.

Integração com Compose

De Classe para JSON:

```
val novoJson = gson.toJson(cliente)
Text("JSON gerado: $novoJson")
```

1 - Criando STRING JSON:

```
val jsonProdutos = """
[
  {"nome": "Caneta", "preco": 2.0},
  {"nome": "Caderno", "preco": 10.0},
  {"nome": "Lápis", "preco": 1.5}
]
""".trimIndent()
```

2 - Converter e exibir:

```
val gson = remember { Gson() }
val lista = gson.fromJson(jsonProdutos, Array<Produto>::class.java).toList()

LazyColumn {
    items(lista.size) { i ->
        val item = lista[i]
        Text("${item.nome} - R$ ${item.preco}")
    }
}
```

Mão na Massa

Vamos separar em pastas **models, services, views**

Models -> as classes até agora.

Services -> crie um novo chamado *ProdutoService*

Views -> Todas as listagens (exceto o MainActivity)

Utilizar ClassProduto

Criar string JSON com produtos

Converter string JSON em lista de objetos

Exibir os produtos em uma LazyColumn

Gerar novo JSON a partir de um objeto

Mão na Massa

1 - ProdutoService

Mão na Mass

2- ListaAPIprodutos

ifests
n+java
om.example.test
models
services
ui.theme
views
ListaAPIprodutos.kt
ListaProdutos.kt
PessoaCard.kt
PreviewFicha.kt
MainActivity.kt
om.example.test (androidT
om.example.test (test)
(generated)
Scripts

```
1  package com.example.test.views
2
3  import androidx.compose.foundation.layout.Column
4  import androidx.compose.foundation.layout.Spacer
5  import androidx.compose.foundation.layout.fillMaxWidth
6  import androidx.compose.foundation.layout.height
7  import androidx.compose.foundation.layout.padding
8  import androidx.compose.foundation.lazy.LazyColumn
9  import androidx.compose.material3.Card
10 import androidx.compose.material3.CardDefaults
11 import androidx.compose.material3.MaterialTheme
12 import androidx.compose.material3.Text
13 import androidx.compose.runtime.Composable
14 import androidx.compose.runtime.remember
15 import androidx.compose.ui.Modifier
16 import androidx.compose.ui.text.font.FontStyle
17 import androidx.compose.ui.tooling.preview.Preview
18 import androidx.compose.ui.unit.dp
19 import com.example.test.models.Produto
20 import com.example.test.services.jsonProdutos
21 import com.example.test.services.objProdudos
22 import com.google.gson.Gson
```

Mão na Massa

2 - ListaAPIprodutos

```
@Composable
fun ExibirProduto(produtos: List<Produto>) {
    Column(modifier = Modifier.padding(8.dp)) {
        val gson = remember { Gson() }

        produtos.forEach { produto ->
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(vertical = 4.dp),
                elevation = CardDefaults.cardElevation(defaultElevation = 4.dp)
            ) {
                Column(modifier = Modifier.padding(16.dp)) {
                    Text(
                        text = "Produto: ${produto.nome}",
                        style = MaterialTheme.typography.titleMedium
                    )
                    Text(
```

Mão na Massa

2 - ListaAPIprodutos

```
            )
            Text(
                text = "Preço: R$${produto.preco}",
                style = MaterialTheme.typography.bodyLarge
            )
            Spacer(modifier = Modifier.height(8.dp))
            Text(
                text = "JSON: ${gson.toJson(produto)}",
                style = MaterialTheme.typography.bodySmall,
                fontStyle = FontStyle.Italic
            )
        }
    }
}
}
```

Mão na Massa

2- ListaAPIprodutos

```
}
@Preview
@Composable
fun listaAPIprodutosPreview(){
    listaAPIprodutos(jsonProdutos);
}
@Preview
@Composable
fun ExibirProdutoPreview(){
    ExibirProduto(objProdudos)
}
```

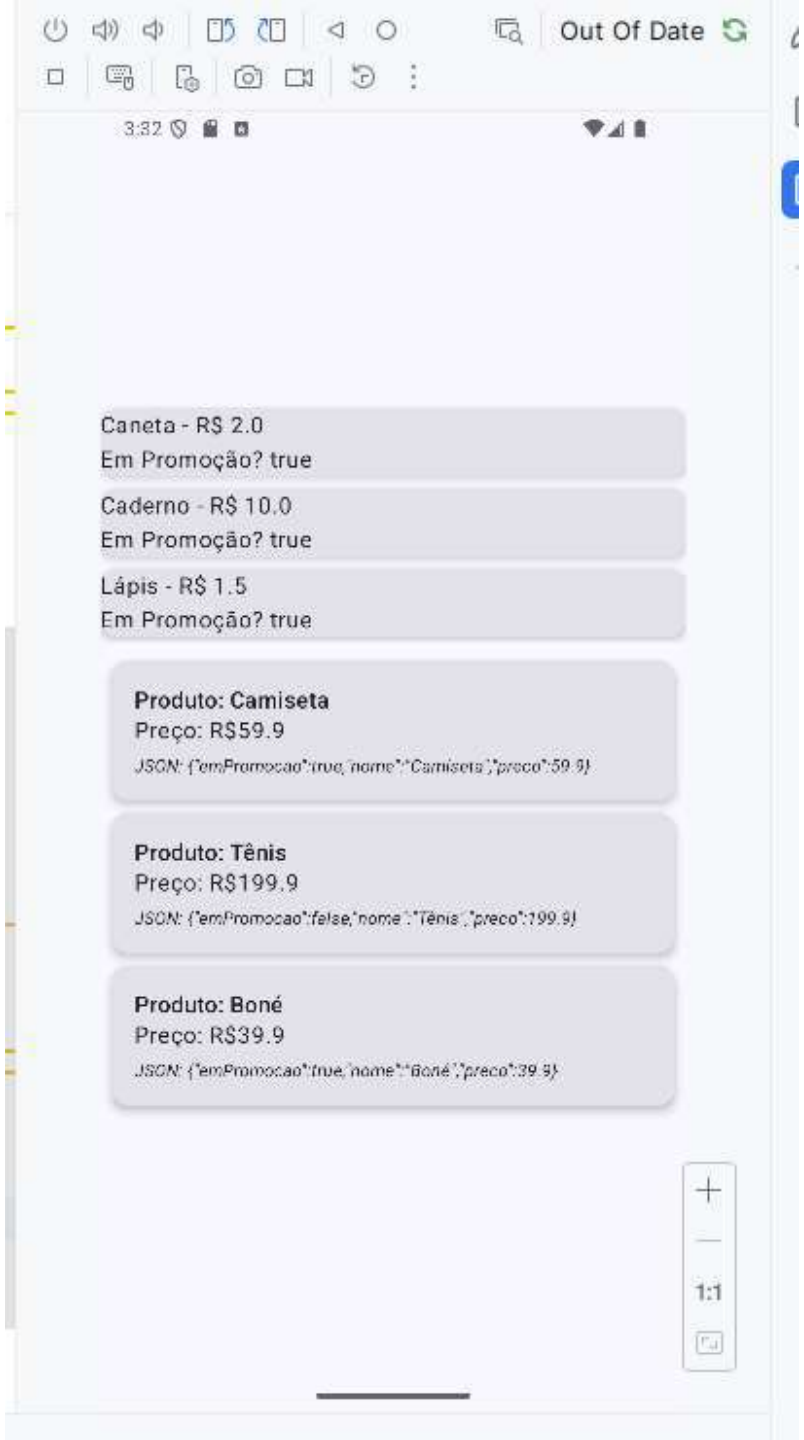
Mão na Massa

3 - Main

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column(
        modifier = modifier.then(Modifier.fillMaxSize()),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        listaAPIprodutosPreview();
        ExibirProdutoPreview();
    }
}
```

Mão na Massa

4 - Resultado



FPFtech
Escola Tecnológica

Obrigado!

FPFtech
Escola Tecnológica

@fpftech.educacional