

Questionário sobre Networking em Flutter

Instruções: Para cada questão, selecione a opção que você considera correta.

1. Qual é o principal propósito do pacote `http` no desenvolvimento Flutter?
 - a) Para criar interfaces de usuário interativas.
 - b) Para realizar requisições HTTP e comunicação com servidores externos.
 - c) Para gerenciar o estado local de um aplicativo.
 - d) Para desenvolver animações complexas.

2. De acordo com as fontes, o que é uma requisição HTTP?
 - a) Um método para armazenar dados localmente no dispositivo.
 - b) A forma como serviços e clientes se comunicam na internet.
 - c) Um tipo de widget usado para layout de tela.
 - d) Uma ferramenta para depurar aplicativos Flutter.

3. Como se instala o pacote `http` em um projeto Flutter?
 - a) Executando o comando `flutter create --package http`.
 - b) Baixando o arquivo diretamente de um navegador.
 - c) Adicionando-o como uma dependência no arquivo `pubspec.yaml` e executando `flutter pub get`.
 - d) Ele já vem pré-instalado com o SDK do Flutter.

4. Para que o site `randomuser.me` foi utilizado como exemplo nas fontes?
 - a) Para hospedar o aplicativo Flutter finalizado.
 - b) Para fornecer dados de usuário simulados (como um "Lorem Ipsum" para pessoas).

- c) Para servir como um console de depuração para requisições de rede.
 - d) Para converter objetos Dart em formato JSON.
5. Em que formato de dados a API `randomuser.me` retorna as informações dos usuários, conforme o vídeo?
- a) XML
 - b) HTML
 - c) YAML
 - d) **JSON**
6. Qual a principal função do construtor `fromJson` (ou `fromJson`) em um modelo de dados Dart?
- a) Converter um objeto Dart em uma string JSON.
 - b) **Criar uma nova instância de um objeto Dart a partir de um mapa JSON.**
 - c) Validar a sintaxe de um arquivo JSON.
 - d) Enviar dados JSON para um servidor.
7. O que a propriedade `response.body` contém após uma requisição HTTP bem-sucedida?
- a) O código de status HTTP da resposta.
 - b) **A string JSON bruta recebida da API.**
 - c) Um objeto Dart já decodificado e pronto para uso.
 - d) O URL original da requisição.
8. Qual pacote Dart é essencial para decodificar uma string JSON em um mapa ou lista Dart?
- a) `dart:io`
 - b) `dart:async`
 - c) **`dart:convert`**

d) `dart:math`

9. Qual código de status HTTP geralmente indica que uma requisição foi bem-sucedida e que tudo funcionou corretamente?

a) 404 (Not Found)

b) 500 (Internal Server Error)

c) **200 (OK)**

d) 301 (Moved Permanently)

10. No contexto de requisições de rede, o que os modificadores `async` e `await` permitem no Dart/Flutter?

a) Execução síncrona do código, bloqueando a interface do usuário.

b) **Esperar que uma `Future` seja concluída sem bloquear a interface do usuário.**

c) Definir métodos estáticos dentro de uma classe.

d) Criar novas threads para processamento paralelo.

11. Qual das seguintes é considerada uma boa prática ao estruturar a lógica de rede em um aplicativo Flutter?

a) Incorporar a lógica de rede diretamente nos widgets da interface do usuário.

b) **Separar a lógica de rede em arquivos ou classes de serviço próprios.**

c) Ignorar o tratamento de erros para simplificar o código.

d) Usar apenas strings para representar dados JSON, sem criar modelos de dados.

12. Além do pacote `http`, qual outro pacote é mencionado nas fontes como sendo mais avançado para networking no Flutter, oferecendo suporte a interceptadores e timeout? a) `connectivity` b) **`dio`** c) `url_launcher` d) `shared_preferences`

13. Para que um aplicativo Android possa acessar a internet, qual etapa é obrigatória no arquivo `AndroidManifest.xml`?

a) Nenhuma declaração é necessária, o acesso à internet é padrão.

b) Adicionar uma tag `<internet-access>` dentro da tag `<application>`.

c) **Declarar a permissão `<uses-permission android:name="android.permission.INTERNET" />`.**

d) Configurar uma propriedade específica no arquivo `pubspec.yaml`.

14. Qual é o tipo de retorno esperado para uma função que busca uma lista de objetos `Person` de uma API de forma assíncrona no Flutter?

a) `List<Person>`

b) `void`

c) **`Future<List<Person>>`**

d) `String`

15. O que o `FutureBuilder` no Flutter faz em relação às requisições de rede?

a) É um widget que faz a requisição de rede em segundo plano.

b) **É um widget que constrói sua interface de usuário com base no estado de um `Future` (como dados recebidos de uma rede).**

c) É uma classe para gerenciar o cache de dados de rede.

d) É uma função para converter dados JSON em objetos Dart.

16. Qual das seguintes NÃO é uma boa prática de networking mencionada nas fontes?

a) Sempre tratar erros de rede. b) Usar modelos de dados (classes) para representar o JSON. c) Separar a lógica de rede em arquivos/serviços próprios. d) **Manter todo o código de requisição e parsing JSON em uma única função `main`.**

17. Para que serve o método `.timeout()` em uma requisição HTTP, como demonstrado na fonte em PDF?

a) Para atrasar o início da requisição em um número específico de segundos.

b) **Para definir um tempo limite para a resposta da requisição, lançando um erro se excedido.**

c) Para repetir automaticamente a requisição após um certo período de tempo.

d) Para cancelar a requisição imediatamente após ser chamada.

18. Como os parâmetros (como a quantidade de usuários a serem buscados) são tipicamente adicionados a uma URL em uma requisição GET?
- a) Inserindo-os no corpo da requisição (body).
 - b) **Adicionando-os após um ponto de interrogação (?) na URL, no formato `chave=valor`.**
 - c) Através de cabeçalhos HTTP personalizados.
 - d) Declarando-os como variáveis globais no código Dart.
19. Qual é o propósito de usar a sintaxe `import 'package:http/http.dart' as http;`?
- a) Para evitar a necessidade de instalar o pacote `http`.
 - b) Para renomear o arquivo `http.dart` para `http`.
 - c) **Para criar um prefixo `http`, que permite acessar as funções e classes do pacote, evitando conflitos de nome.**
 - d) Para indicar que o pacote `http` é opcional no projeto.
20. Ao criar um modelo `Person` a partir de um JSON, como você acessaria campos aninhados como `name` que contém `title`, `first` e `last`, conforme explicado no vídeo?
- a) Acessando diretamente `json['name']` e assumindo que é uma string.
 - b) **Acessando `json['name']['first']` (e outros campos aninhados) e combinando-os.**
 - c) Usando um método `toJson()` dentro da classe `Person`.
 - d) O construtor `fromJson` ignora campos aninhados para simplificar o processo.

Respostas

1. **b) Para realizar requisições HTTP e comunicação com servidores externos.**
2. **b) A forma como serviços e clientes se comunicam na internet.**
3. **c) Adicionando-o como uma dependência no arquivo `pubspec.yaml` e executando `flutter pub get`.**

4. b) Para fornecer dados de usuário simulados (como um "Lorem Ipsum" para pessoas).
5. d) JSON
6. b) Criar uma nova instância de um objeto Dart a partir de um mapa JSON.
7. b) A string JSON bruta recebida da API.
8. c) `dart:convert`
9. c) 200 (OK)
10. b) Esperar que uma `Future` seja concluída sem bloquear a interface do usuário.
11. b) Separar a lógica de rede em arquivos ou classes de serviço próprios.
12. b) `dio`
13. c) Declarar a permissão `<uses-permission android:name="android.permission.INTERNET" />`.
14. c) `Future<List<Person>>`
15. b) É um widget que constrói sua interface de usuário com base no estado de um `Future` (como dados recebidos de uma rede).
16. d) Manter todo o código de requisição e parsing JSON em uma única função `main`.
17. b) Para definir um tempo limite para a resposta da requisição, lançando um erro se excedido.
18. b) Adicionando-os após um ponto de interrogação (?) na URL, no formato `chave=valor`.
19. c) Para criar um prefixo `http.` que permite acessar as funções e classes do pacote, evitando conflitos de nome.
20. b) Acessando `json['name']['first']` (e outros campos aninhados) e combinando-os.