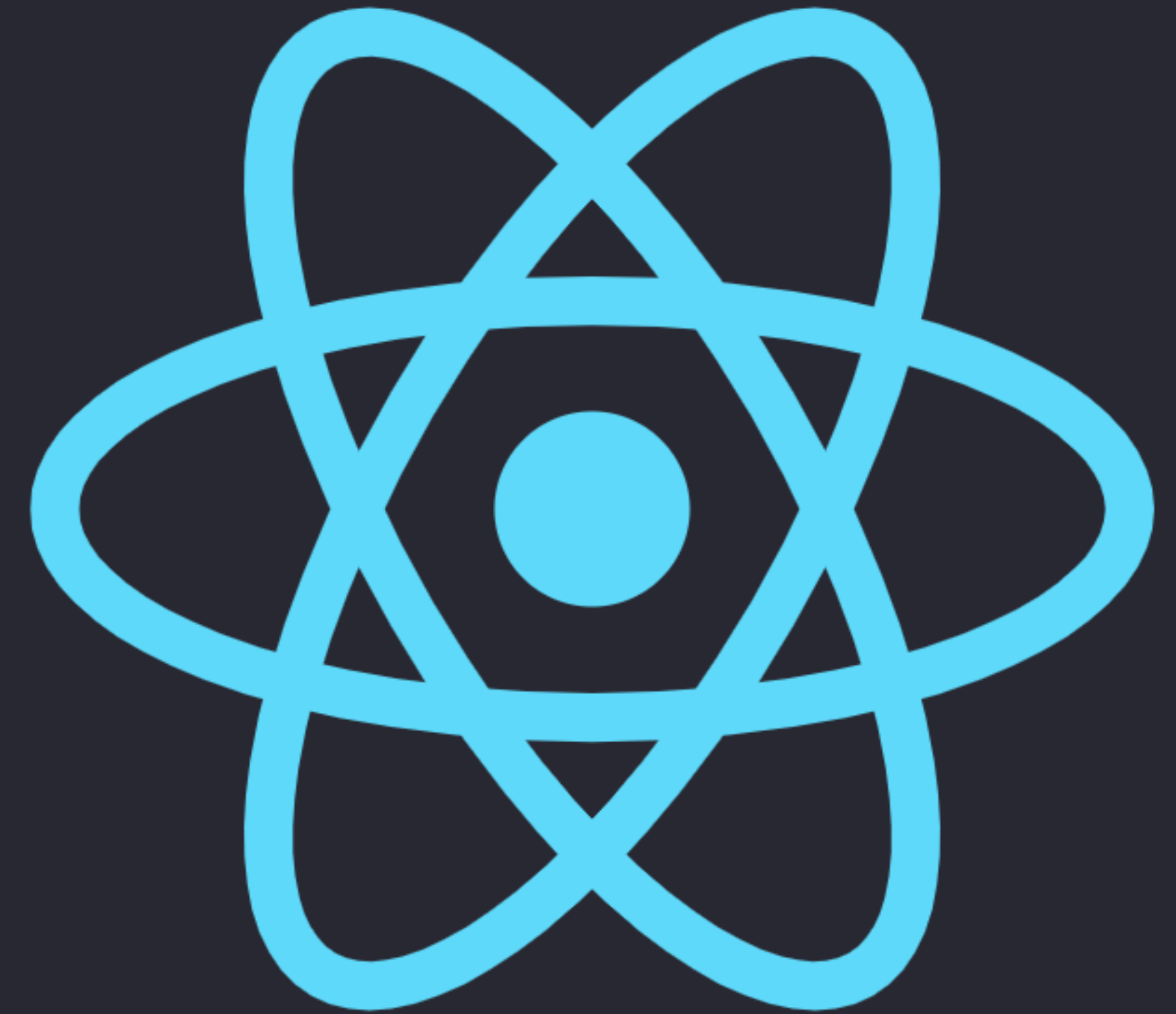


React Native

Aprenda uma vez, escreva em qualquer lugar.



Chamadas para API

React Native

Networking

Muitos aplicativos móveis precisam carregar recursos de uma URL remota. Você pode querer fazer uma solicitação POST para uma API REST ou pode precisar buscar um pedaço de conteúdo estático de outro servidor.

Usando Fetch

React Native fornece a [API Fetch](#) para suas necessidades de rede. A busca parecerá familiar se você já usou `XMLHttpRequest` ou outras APIs de rede antes. Você pode consultar o guia do MDN sobre como [usar o Fetch](#) para obter informações adicionais.

React Native

Fazendo solicitações

Para buscar conteúdo de um URL arbitrário, você pode passar o URL para buscar:

```
fetch('https://mywebsite.com/mydata.json');
```

Fetch também leva um segundo argumento opcional que permite personalizar a solicitação HTTP. Você pode especificar cabeçalhos adicionais ou fazer uma solicitação POST:

```
fetch('https://mywebsite.com/endpoint/', {
  method: 'POST',
  headers: {
    Accept: 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    firstParam: 'yourValue',
    secondParam: 'yourOtherValue'
  })
});
```

Dê uma olhada nos [documentos Fetch Request](#) para uma lista completa de propriedades.

React Native

Lidando com a resposta

Os exemplos acima mostram como você pode fazer uma solicitação. Em muitos casos, você desejará fazer algo com a resposta.

A rede é uma operação inerentemente assíncrona. O método Fetch retornará uma promessa que torna simples escrever um código que funcione de maneira assíncrona:

```
const getMoviesFromApi = () => {  
  return fetch('https://reactnative.dev/movies.json')  
    .then((response) => response.json())  
    .then((json) => {  
      return json.movies;  
    })  
    .catch((error) => {  
      console.error(error);  
    });  
};
```


React Native

Você também pode usar a sintaxe `async` / `await` em um aplicativo React Native:

```
const getMoviesFromApiAsync = async () => {  
  try {  
    const response = await fetch(  
      'https://reactnative.dev/movies.json'  
    );  
    const json = await response.json();  
    return json.movies;  
  } catch (error) {  
    console.error(error);  
  }  
};
```

Não se esqueça de detectar quaisquer erros que possam ocorrer `fetch`, caso contrário, eles serão ignorados silenciosamente.

api-call

Entrada de Datos

React Native

TextInput ⓘ ↗

^ Expo

```
import React from "react";
import { SafeAreaView, StyleSheet, TextInput } from "react-native";

const UselessTextInput = () => {
  const [text, onChangeText] = React.useState("Useless Text");
  const [number, onChangeNumber] = React.useState(null);

  return (
    <SafeAreaView>
      <TextInput
        style={styles.input}
        onChangeText={onChangeText}
        value={text}
      />
      <TextInput
        style={styles.input}
        onChangeText={onChangeNumber}
        value={number}
        placeholder="useless placeholder"
        keyboardType="numeric"
      />
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  input: {
```

Teste

Testless

Preview ☒ My Device ☐ iOS ☐ Android ☒ Web

Utilizando o **STATE** em conjunto com o **TextInput** para a entrada de dados

Input-fields

Envio de Dados para a API

React Native

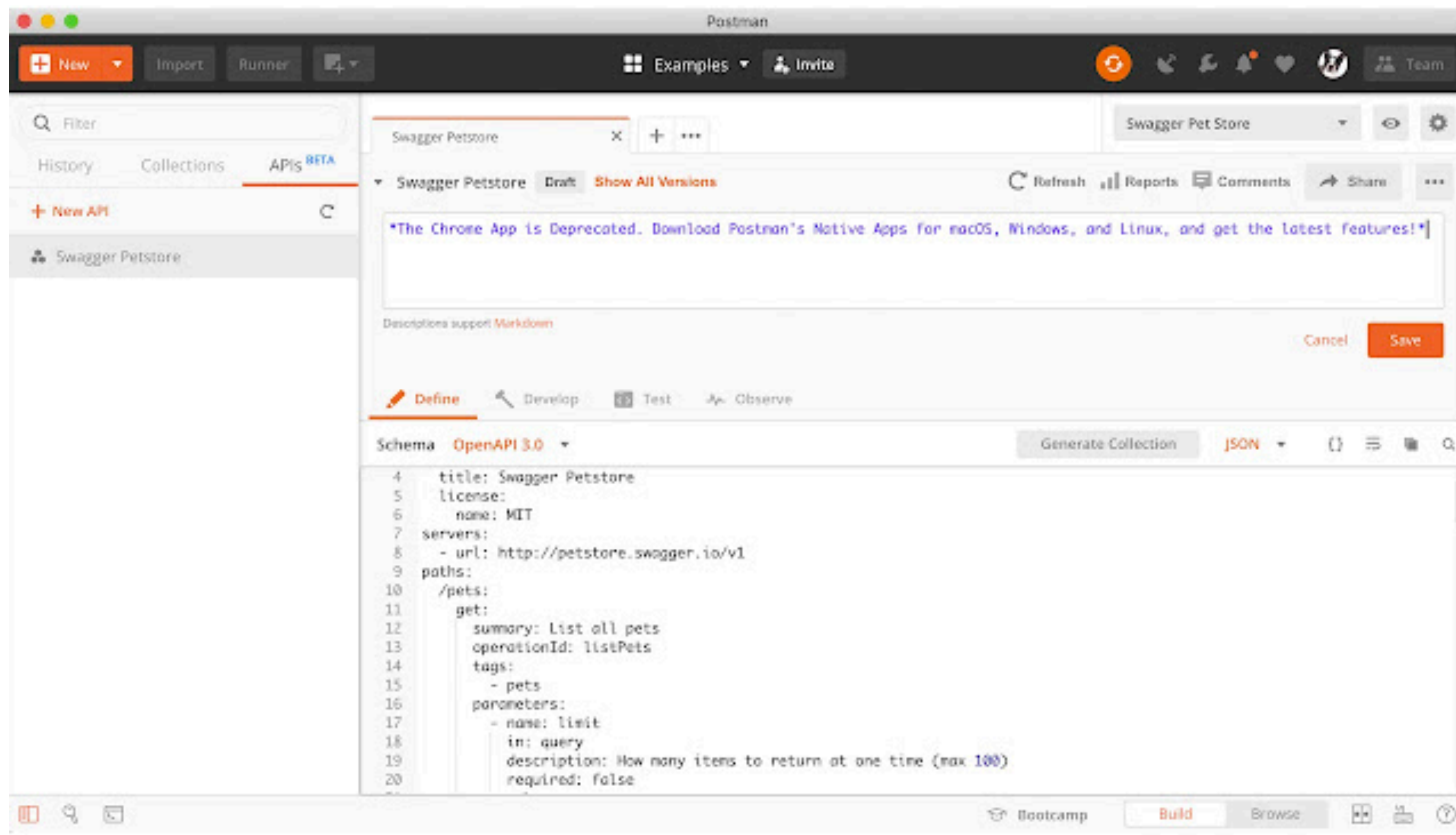
Top 7
Programming
Languages for
**Backend
Development**



Top 7 Programming Languages for Backend Web Development

- JavaScript.
- Python. ←
- PHP.
- Java.
- Ruby.
- Golang.
- C#

React Native



React Native

```
const getMessages = async () => {  
  try {  
  
    const rawResponse = await fetch('http://localhost:5000/api/messages', {  
      method: 'POST',  
      headers: {  
        'Accept': 'application/json',  
        'Content-Type': 'application/json'  
      },  
      body: JSON.stringify({message: 'Hello World'})  
    });  
  
    const content = await rawResponse.json();  
    setData(content.messages);  
  } catch (error) {  
    console.error(error);  
  } finally {  
    setLoading(false);  
  }  
}
```

Envio de dados
utilizando JSON

api-call-data

Utilizando Dynamic Inputs

React Native

Dynamic Form Fields in React

First Name

Last Name

- +

Save

```
[
  {
    "firstName": "",
    "lastName": ""
  }
]
```

dynamic-fields

<https://codesandbox.io/s/react-dynamic-form-fields-3fjbd?from-embed=&file=/src/index.js>