# Flutter App with Back4App Integration

Create the Flutter Application using the Backend as service with Back4App Integration with help of Parse SDK and create the TODO Application which performs basic CRUD operations with Add, Update, Delete and Selection functionalities and Real time connection with Back4App Data server for real time CRUD operations.

Cross Platform Application Development (SEZG585)

Assignment - 1

Submitted By – Kenil Doshi (2022MT93719)

(2022mt93719@wilp.bits-pilani.ac.in)

# Agenda to Objective and Activities

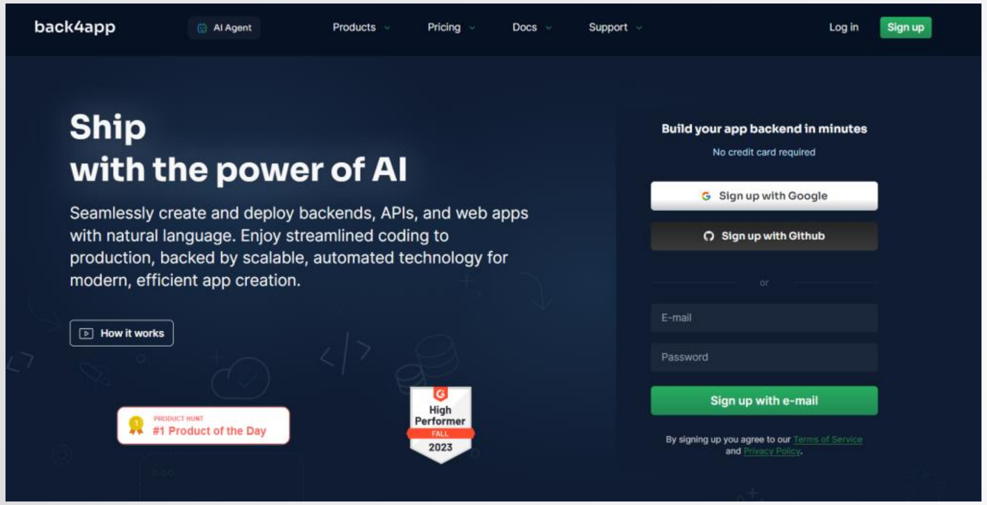
**Objective:** In this assignment, you will create a Flutter app that connects to Back4App, a Backend-as-a-Service (BaaS) platform, to manage tasks. You will be responsible for setting up the Back4App backend, creating the Flutter app, and implementing the necessary functionality to interact with the backend.

## Agenda to Activities:

- **Set Up Back4App** - Sign up for a Back4App account (if not already done), Create a new Back4App app. Create a class in Back4App named Task with columns title (String) and description (String).
- **Flutter Setup** - Create a new Flutter project, Add the required dependencies to your pubspec.yaml file. Initialize the Parse SDK in your Flutter app.
- **Task Creation** - Create a screen for adding new tasks. Implement functionality to create and save tasks to Back4App. Verify that newly created tasks appear in the task list.
- **Task List** - Create a screen in your Flutter app to display a list of tasks. Implement a function to fetch tasks from Back4App using the Back4App API. Display the tasks in a list view with titles and descriptions.
- **Task Details** - Add a feature to view task details when a task is tapped in the task list. Display the title and description of the selected task.
- **Bonus Features** - Add a feature to edit and update existing tasks. Implement a feature for task deletion and Add any additional features or enhancements:
  - On Long Press Task will be completed or convert to pending vice-versa.
  - Delete all the Task from Top Bar and show the snack bar for all the events performed.

# Set Up Back4App

Sign up for a Back4App account and Create a new Back4App app.



Ship with the power of AI

Seamlessly create and deploy backends, APIs, and web apps with natural language. Enjoy streamlined coding to production, backed by scalable, automated technology for modern, efficient app creation.

Build your app backend in minutes

No credit card required

Sign up with Google

Sign up with Github

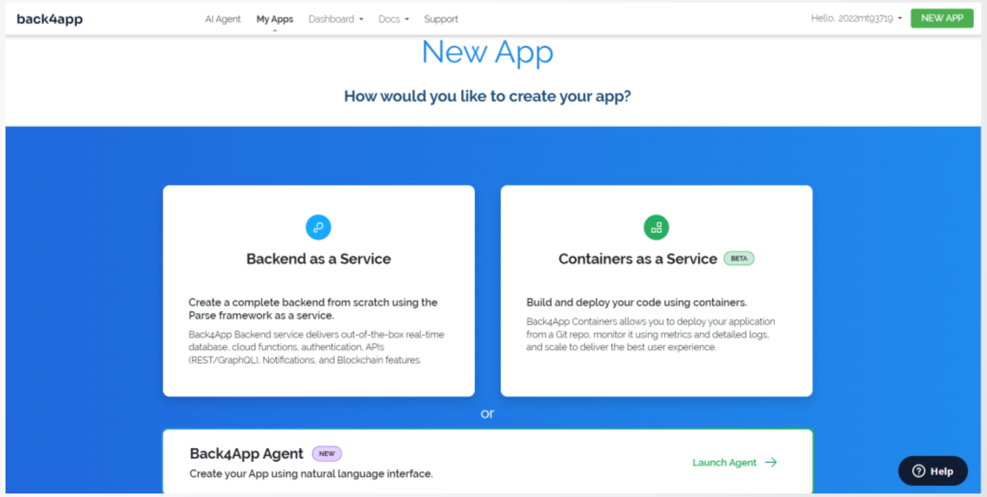
or

E-mail

Password

Sign up with e-mail

By signing up you agree to our Terms of Service and Privacy Policy.



New App

How would you like to create your app?

Backend as a Service

Create a complete backend from scratch using the Parse framework as a service.

Back4App Backend service delivers out-of-the-box real-time database, cloud functions, authentication, APIs (REST/GraphQL), Notifications, and Blockchain features.

Containers as a Service BETA

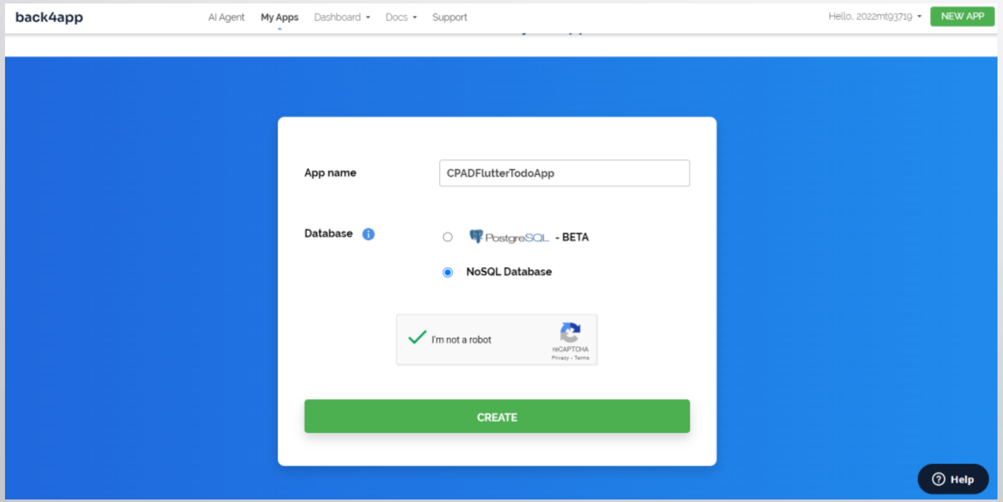
Build and deploy your code using containers.

Back4App Containers allows you to deploy your application from a Git repo, monitor it using metrics and detailed logs, and scale to deliver the best user experience.

or

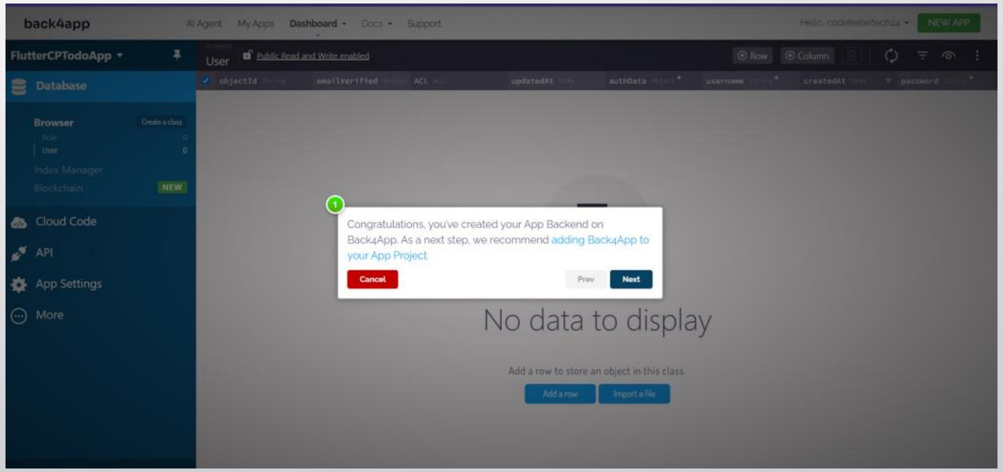
Back4App Agent NEW

Create your App using natural language interface.



App name: CPADFlutterTodoApp

Database: ( ) PostgreSQL - BETA [✓ NoSQL Database]



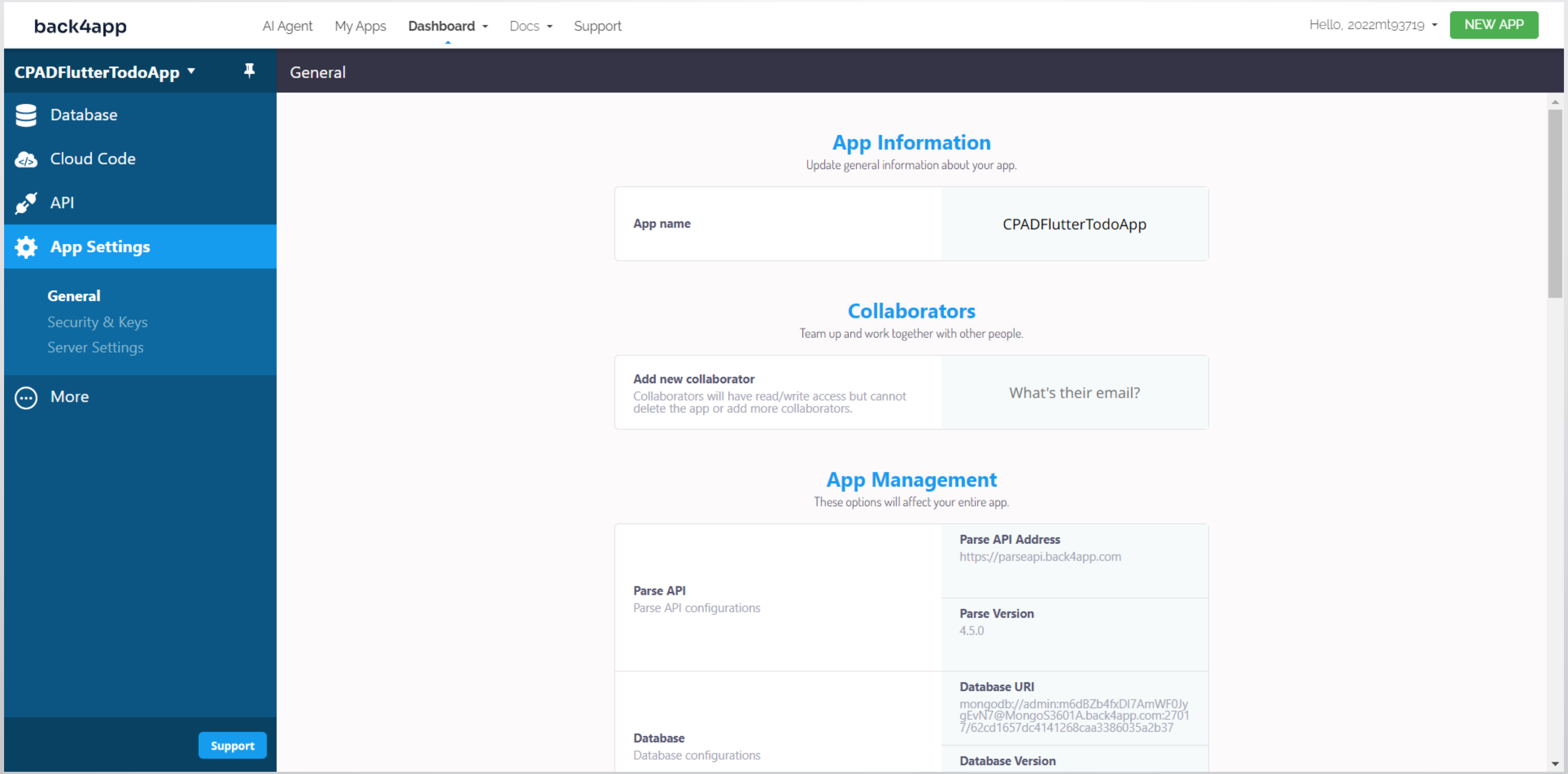
Congratulations, you've created your App Backend on Back4App. As a next step, we recommend adding Back4App to your App Project.

No data to display

Add a row to store an object in this class.

# Set Up Back4App

Back4App Credentials



General

**App Information**

Update general information about your app.

| | |
|---|---|
| **App name** | CPADFlutterTodoApp |

**Collaborators**

Team up and work together with other people.

| | |
|---|---|
| **Add new collaborator** Collaborators will have read/write access but cannot delete the app or add more collaborators. | What's their email? |

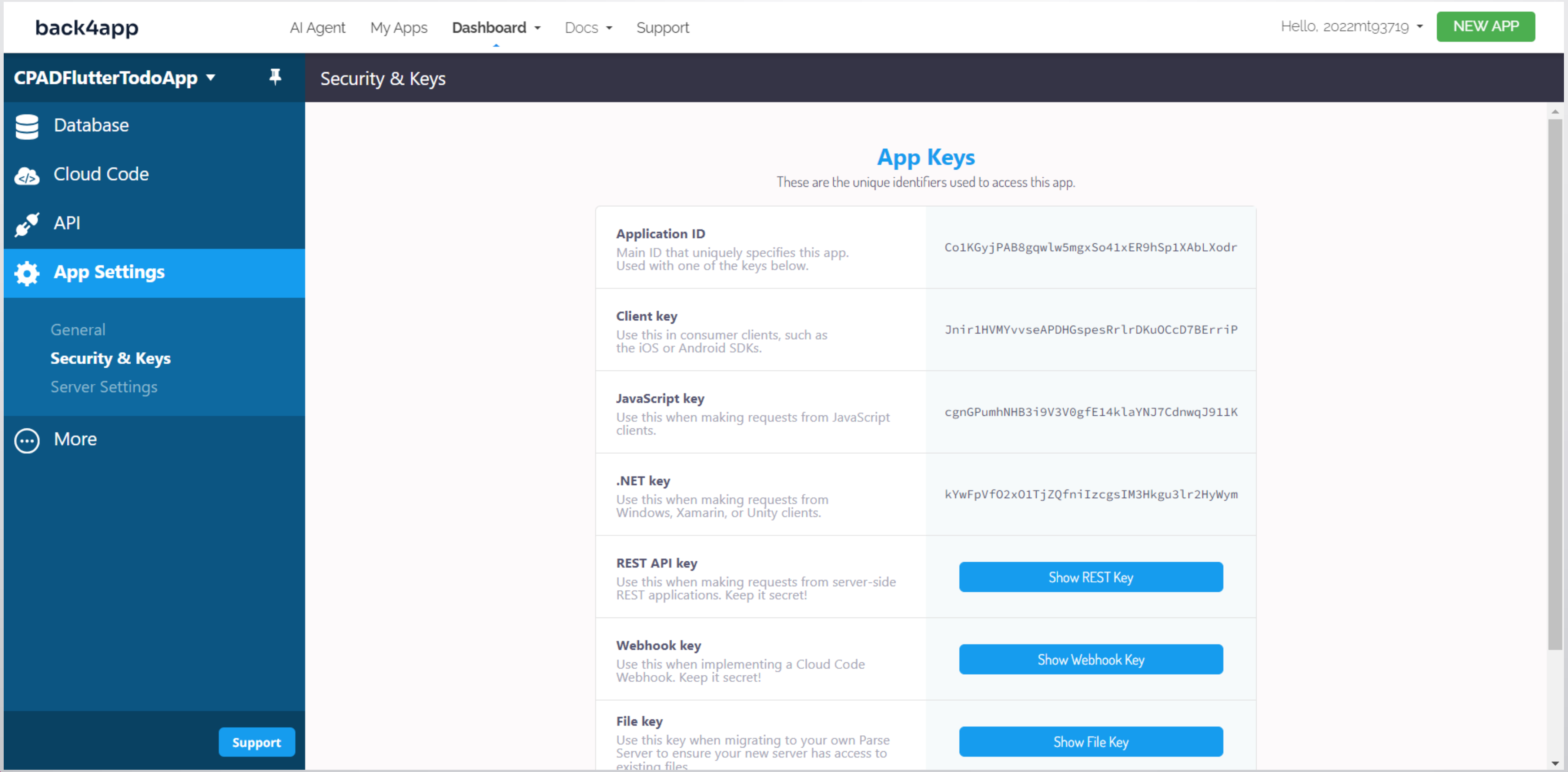
**App Management**

These options will affect your entire app.

| | |
|---|---|
| **Parse API** Parse API configurations | **Parse API Address** https://parseapi.back4app.com |
| | **Parse Version** 4.5.0 |
| **Database** Database configurations | **Database URI** mongodb://admin:m6dBZb4fxDI7AmWF0JygEvN7@MongoS3601A.back4app.com:27017/62cd1657dc4141268caa3386035a2b37 |
| | **Database Version** |

# Set Up Back4App

Back4App Credentials



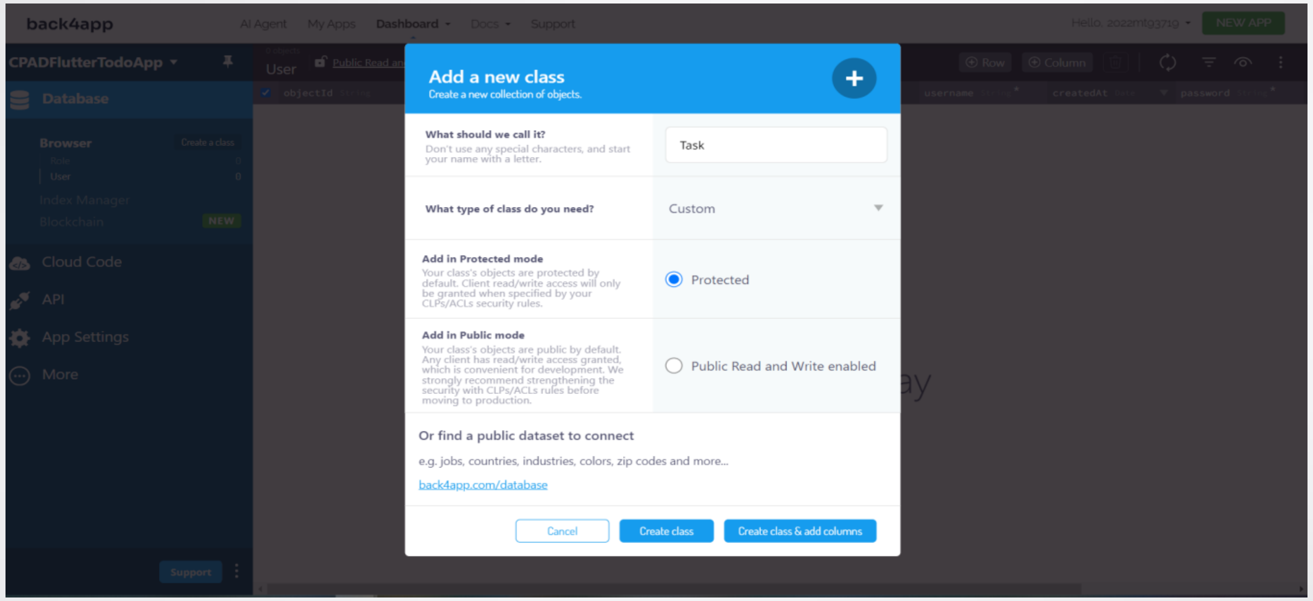
**App Keys**

These are the unique identifiers used to access this app.

| | |
|---|---|
| **Application ID** Main ID that uniquely specifies this app. Used with one of the keys below. | Co1KGyjPAB8gqwlw5mgxSo41xER9hSp1XAbLXodr |
| **Client key** Use this in consumer clients, such as the iOS or Android SDKs. | Jnir1HVMYvvseAPDHGspesRrlrDKuOCcD7BErriP |
| **JavaScript key** Use this when making requests from JavaScript clients. | cgnGPumhNHB3i9V3V0gfE14klaYNJ7CdnwqJ911K |
| **.NET key** Use this when making requests from Windows, Xamarin, or Unity clients. | kYwFpVfO2xO1TjZQfniIzcgsIM3Hkgu3lr2HyWym |
| **REST API key** Use this when making requests from server-side REST applications. Keep it secret! | Show REST Key |
| **Webhook key** Use this when implementing a Cloud Code Webhook. Keep it secret! | Show Webhook Key |
| **File key** Use this key when migrating to your own Parse Server to ensure your new server has access to existing files. | Show File Key |

# Set Up Back4App

Create a class in Back4App named Task with columns title (String) and description (String)



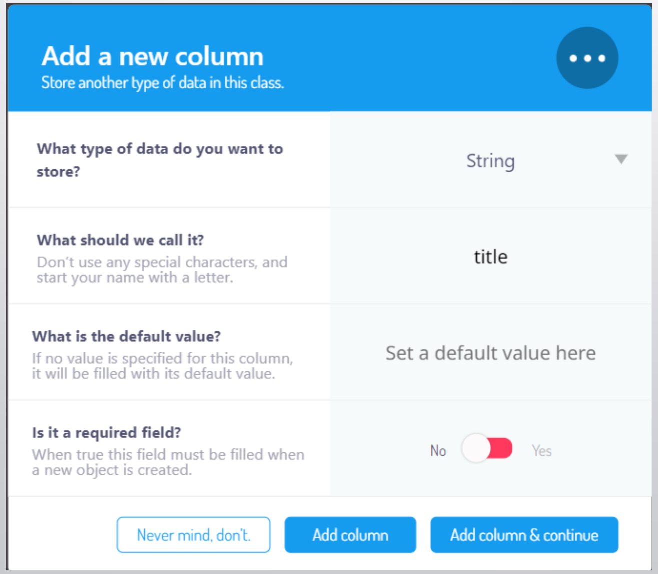
**Add a new class**
Create a new collection of objects.

| | |
|---|---|
| **What should we call it?** Don't use any special characters, and start your name with a letter. | Task |
| **What type of class do you need?** | Custom |
| **Add in Protected mode** Your class's objects are protected by default. Client read/write access will only be granted when specified by your CLPs/ACLs security rules. | [✓ Protected] |
| **Add in Public mode** Your class's objects are public by default. Any client has read/write access granted, which is convenient for development. We strongly recommend strengthening the security with CLPs/ACLs rules before moving to production. | Public Read and Write enabled |

Or find a public dataset to connect

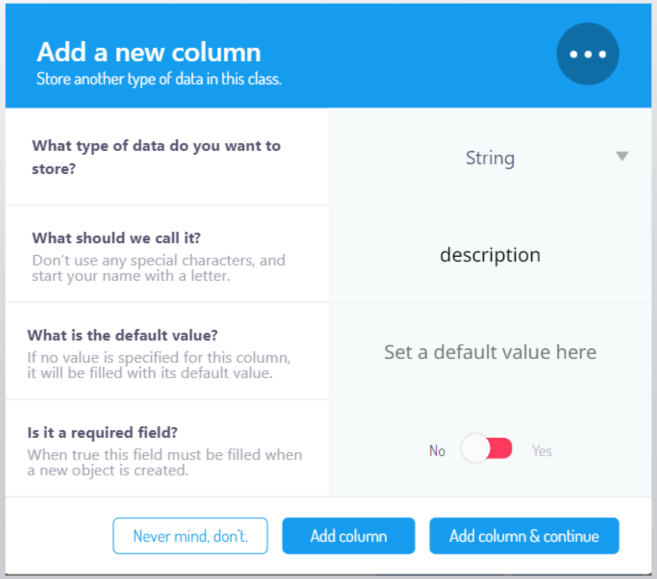
e.g. jobs, countries, industries, colors, zip codes and more...

back4app.com/database



**Add a new column**
Store another type of data in this class.

| | |
|---|---|
| **What type of data do you want to store?** | String |
| **What should we call it?** Don't use any special characters, and start your name with a letter. | title |
| **What is the default value?** If no value is specified for this column, it will be filled with its default value. | Set a default value here |
| **Is it a required field?** When true this field must be filled when a new object is created. | [✓ No] Yes |

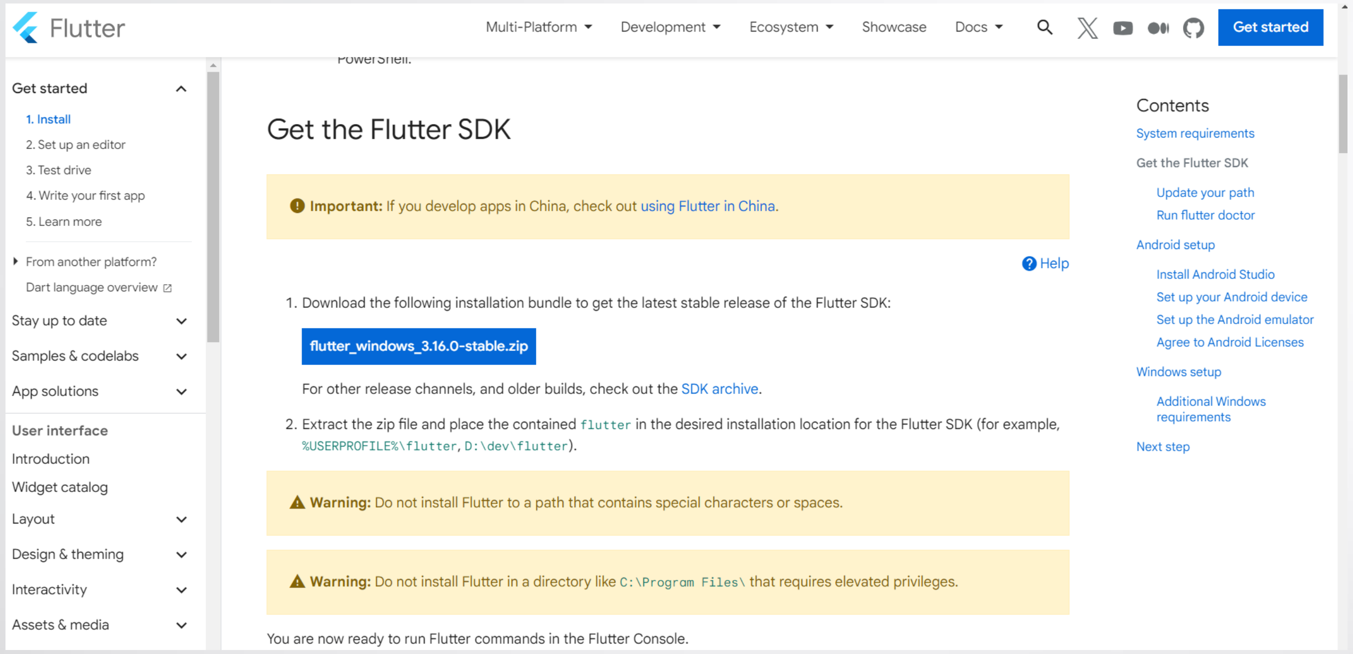


**Add a new column**
Store another type of data in this class.

| | |
|---|---|
| **What type of data do you want to store?** | String |
| **What should we call it?** Don't use any special characters, and start your name with a letter. | description |
| **What is the default value?** If no value is specified for this column, it will be filled with its default value. | Set a default value here |
| **Is it a required field?** When true this field must be filled when a new object is created. | [✓ No] Yes |

## Flutter Setup

Download and Install Flutter SDK to PC and set ENV Path to use it and Run Flutter Doctor.



Get the Flutter SDK

**Important:** If you develop apps in China, check out using Flutter in China.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

   flutter_windows_3.16.0-stable.zip

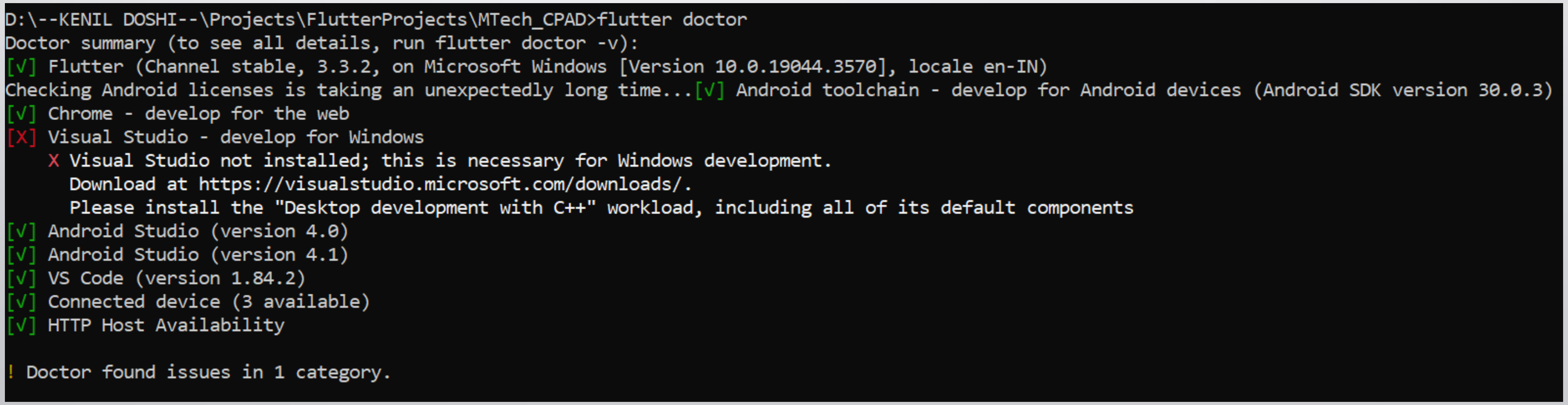
   For other release channels, and older builds, check out the SDK archive.

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `%USERPROFILE%\flutter`, `D:\dev\flutter`).

**Warning:** Do not install Flutter to a path that contains special characters or spaces.

**Warning:** Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

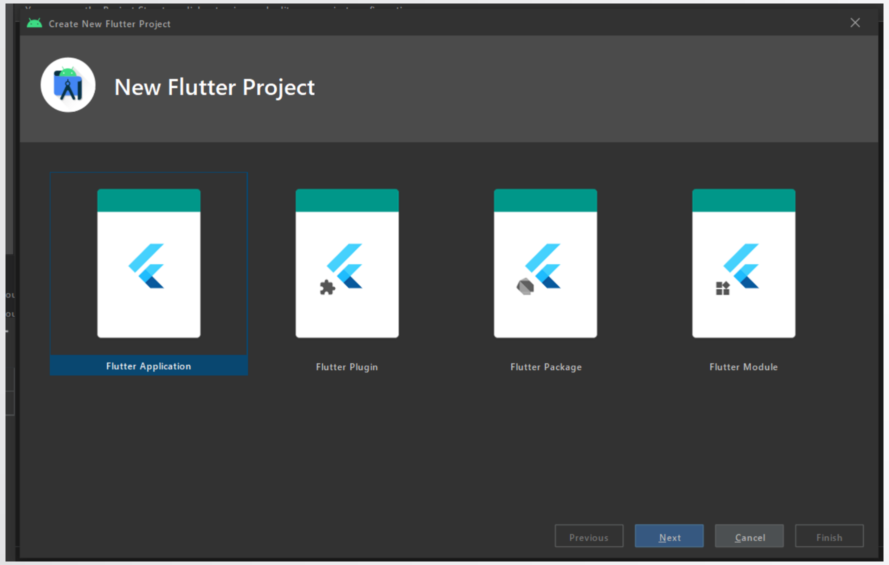
You are now ready to run Flutter commands in the Flutter Console.

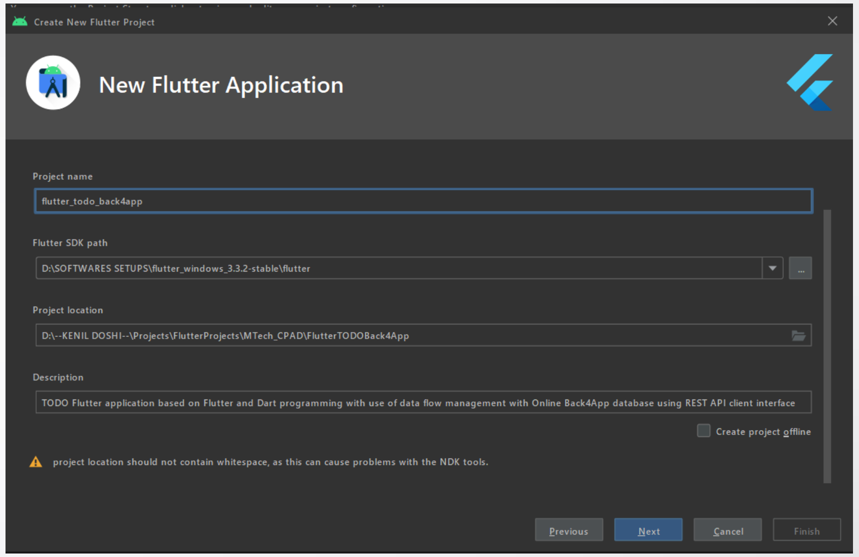


```
D:\--KENIL DOSHI--\Projects\FlutterProjects\MTech_CPAD>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.3.2, on Microsoft Windows [Version 10.0.19044.3570], locale en-IN)
Checking Android licenses is taking an unexpectedly long time...[√] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[√] Chrome - develop for the web
[X] Visual Studio - develop for Windows
    X Visual Studio not installed; this is necessary for Windows development.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[√] Android Studio (version 4.0)
[√] Android Studio (version 4.1)
[√] VS Code (version 1.84.2)
[√] Connected device (3 available)
[√] HTTP Host Availability

! Doctor found issues in 1 category.
```

# Flutter Setup

Create the Flutter Project in Android Studio and set the patch of the Flutter SDK and Install Plugins to Studio.





Project name: flutter_todo_back4app

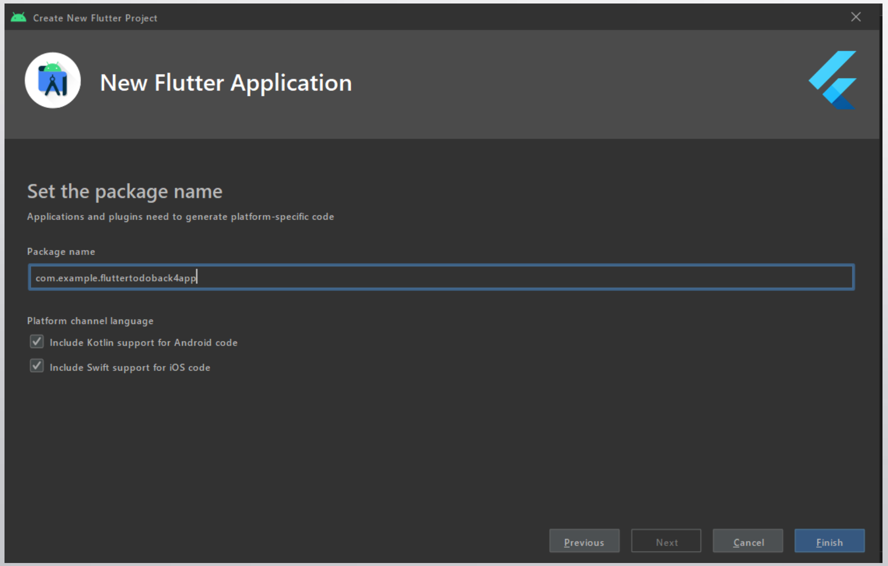
Flutter SDK path: D:\SOFTWARES SETUPS\flutter_windows_3.3.2-stable\flutter

Project location: D:\--KENIL DOSHI--\Projects\FlutterProjects\MTech_CPAD\FlutterTODOBack4App

Description: TODO Flutter application based on Flutter and Dart programming with use of data flow management with Online Back4App database using REST API client interface

Create project offline

project location should not contain whitespace, as this can cause problems with the NDK tools.



Set the package name

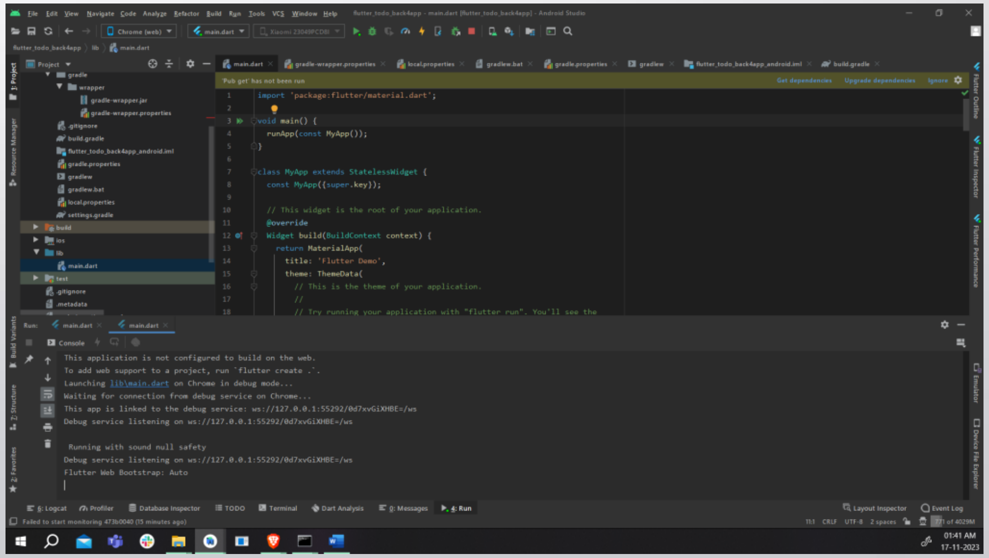
Applications and plugins need to generate platform-specific code

Package name: com.example.fluttertodoback4app

Platform channel language

✓ Include Kotlin support for Android code

✓ Include Swift support for iOS code



'Pub get' has not been run

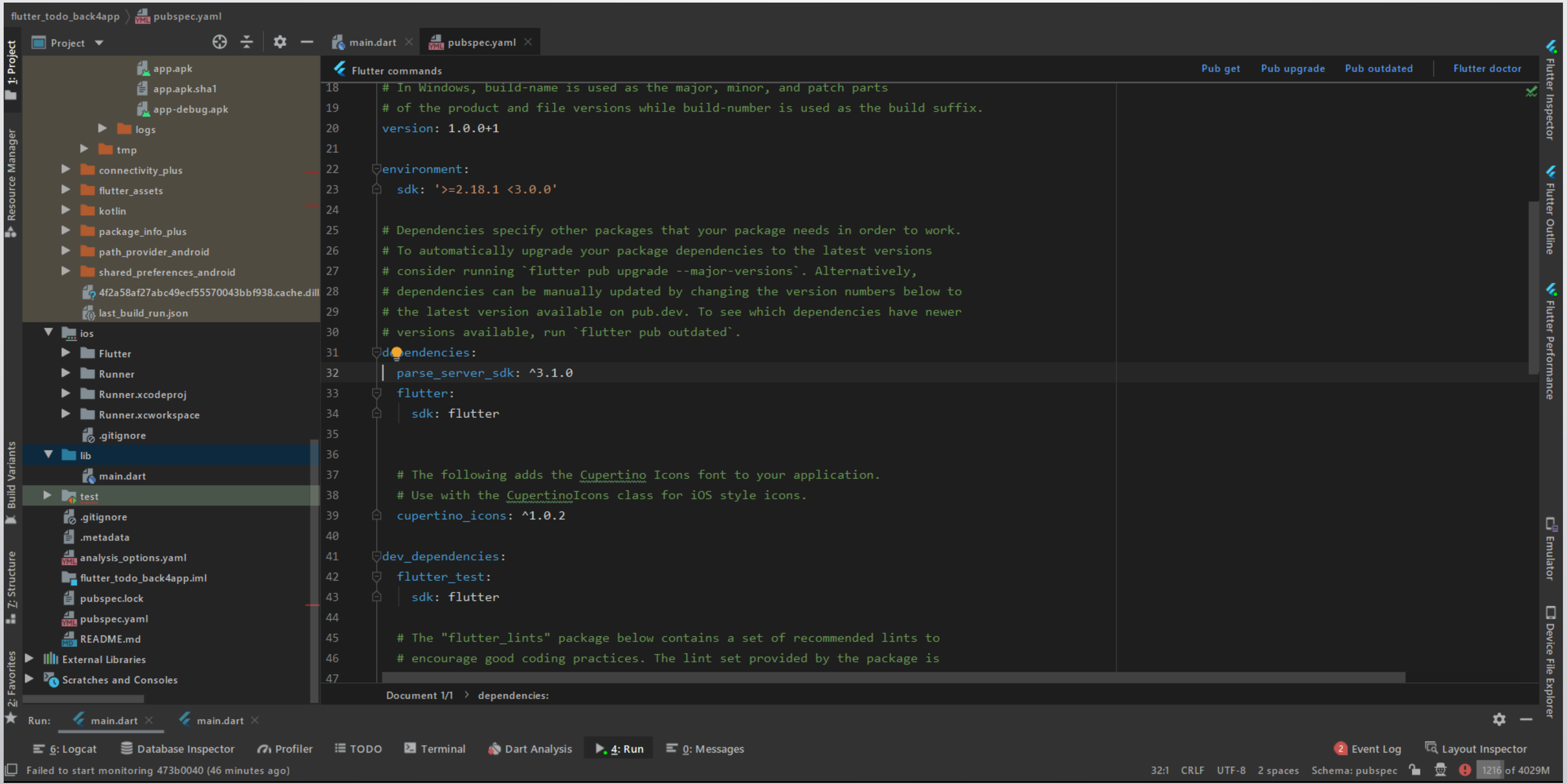
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
```

```
This application is not configured to build on the web.
To add web support to a project, run `flutter create .`.
Launching lib\main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...
This app is linked to the debug service: ws://127.0.0.1:55292/0d7xvGiXHBE=/ws
Debug service listening on ws://127.0.0.1:55292/0d7xvGiXHBE=/ws

 Running with sound null safety
Debug service listening on ws://127.0.0.1:55292/0d7xvGiXHBE=/ws
Flutter Web Bootstrap: Auto
```

Failed to start monitoring 473b0040 (15 minutes ago)

# Flutter Setup

Add the required dependencies to your pubspec.yaml file. Initialize the Parse SDK in your Flutter app.

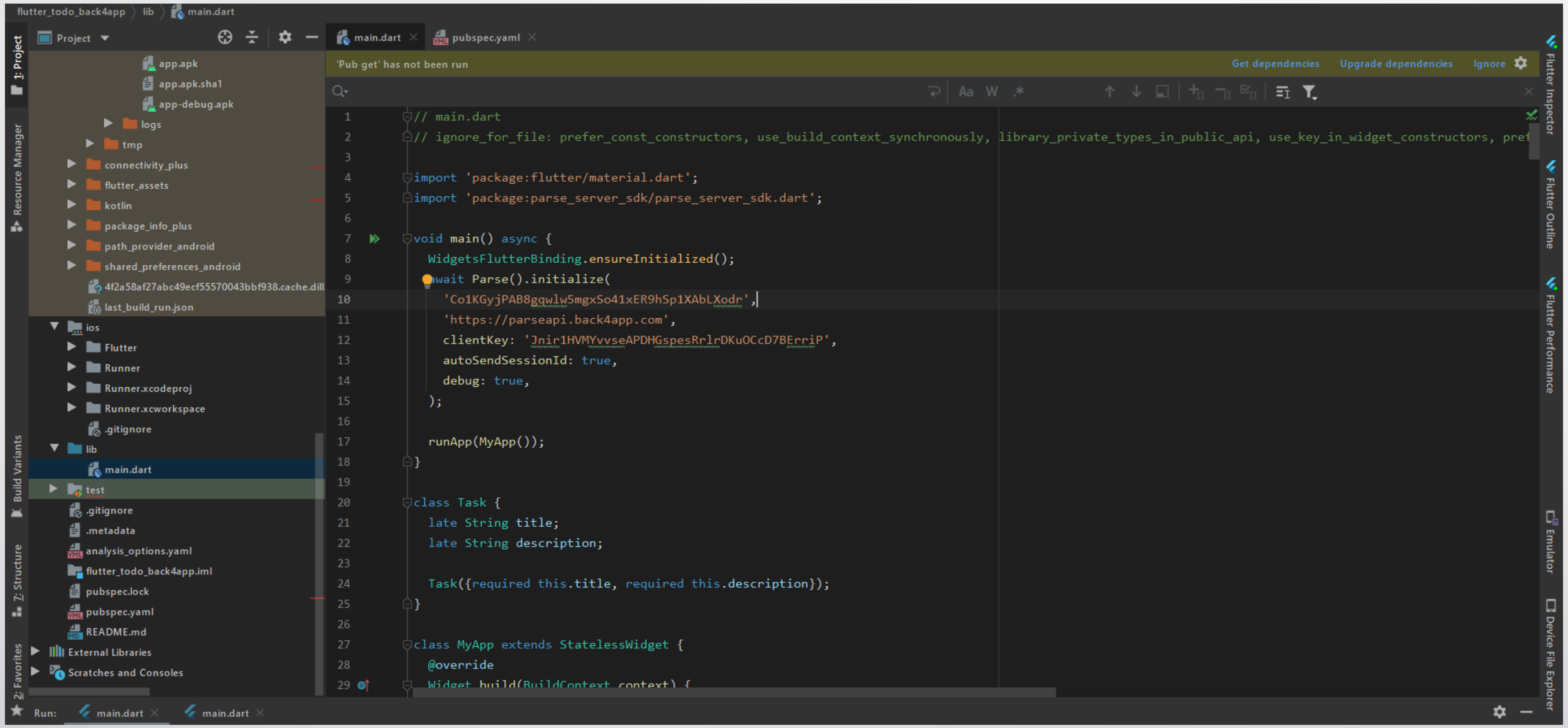


```yaml
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: '>=2.18.1 <3.0.0'

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  parse_server_sdk: ^3.1.0
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
```

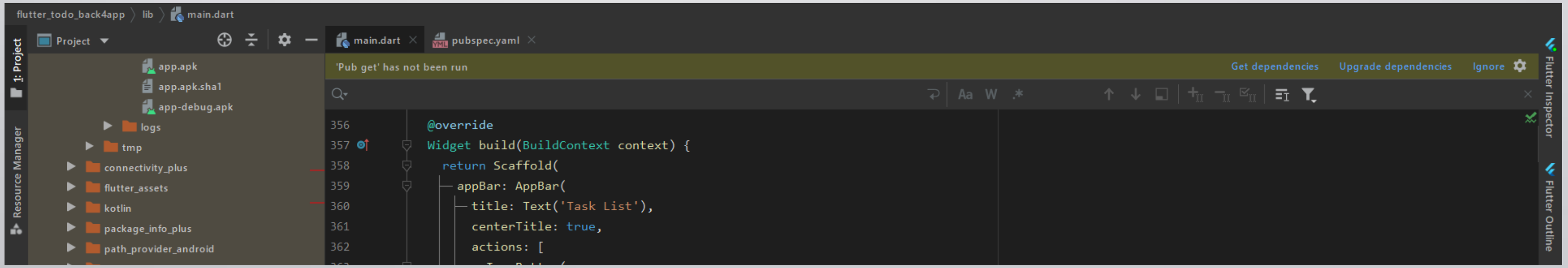
Failed to start monitoring 473b0040 (46 minutes ago)

# Task List

- Add the Back4App Credentials to Flutter App with Application ID, Client ID, Parse URL.
- Create a screen in your Flutter app to display a list of tasks.



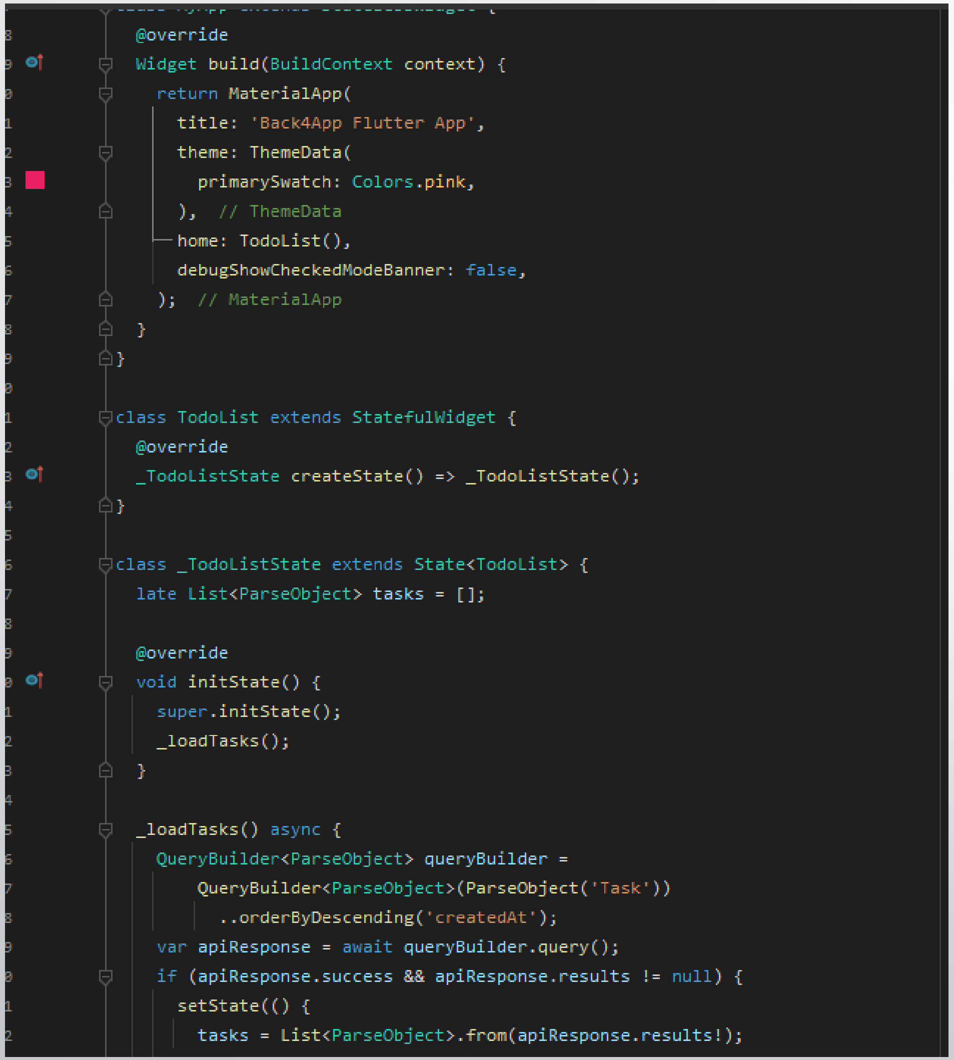
```dart
// main.dart
// ignore_for_file: prefer_const_constructors, use_build_context_synchronously, library_private_types_in_public_api, use_key_in_widget_constructors, pref

import 'package:flutter/material.dart';
import 'package:parse_server_sdk/parse_server_sdk.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Parse().initialize(
    'Co1KGyjPAB8gqwlw5mgxSo41xER9hSp1XAbLXodr',
    'https://parseapi.back4app.com',
    clientKey: 'Jnir1HVMYvvseAPDHGspesRrlrDKuOCcD7BErriP',
    autoSendSessionId: true,
    debug: true,
  );

  runApp(MyApp());
}

class Task {
  late String title;
  late String description;

  Task({required this.title, required this.description});
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
```



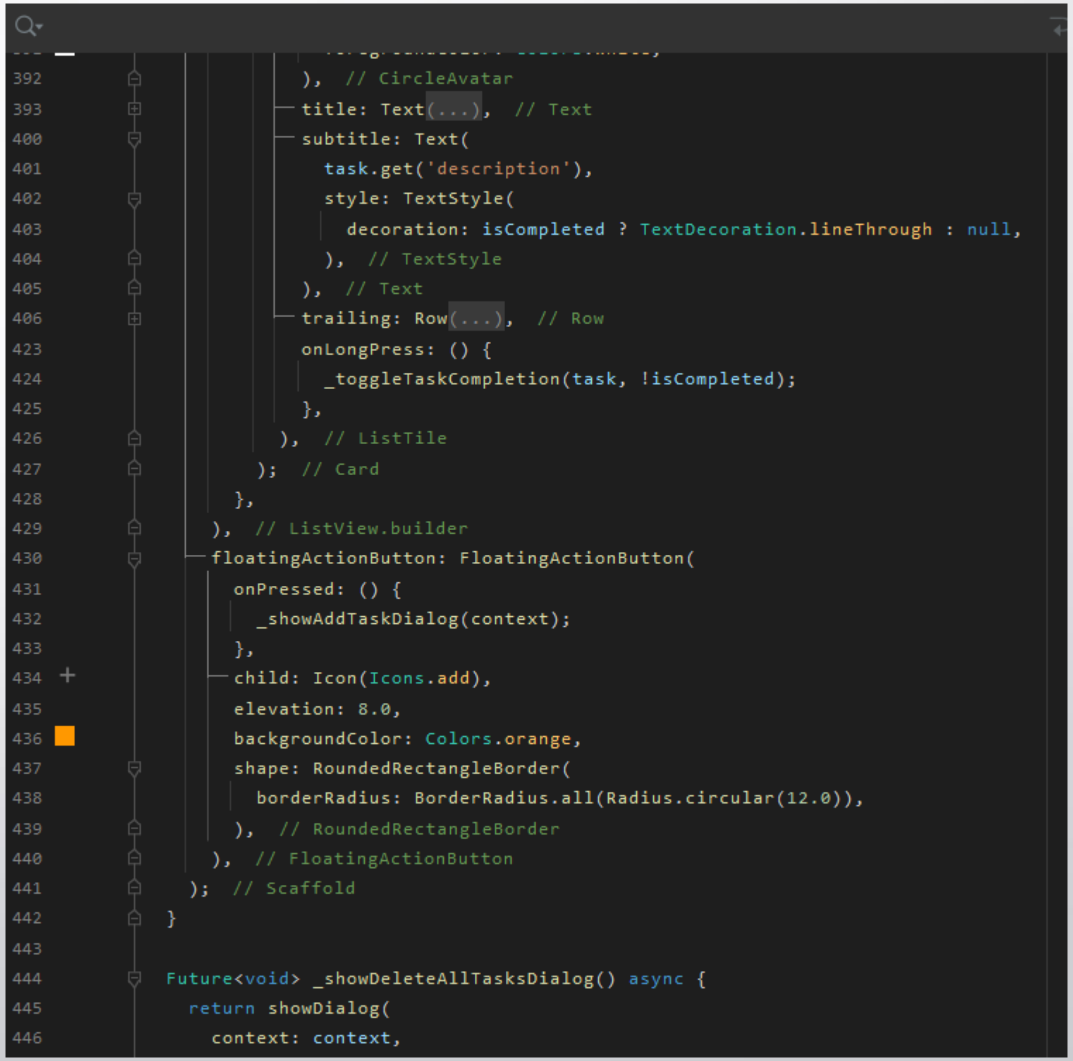
```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Task List'),
        centerTitle: true,
        actions: [
```

# Task List

- Implement a function to fetch tasks from Back4App using the Back4App API.
- Display the tasks in a list view with titles and descriptions.



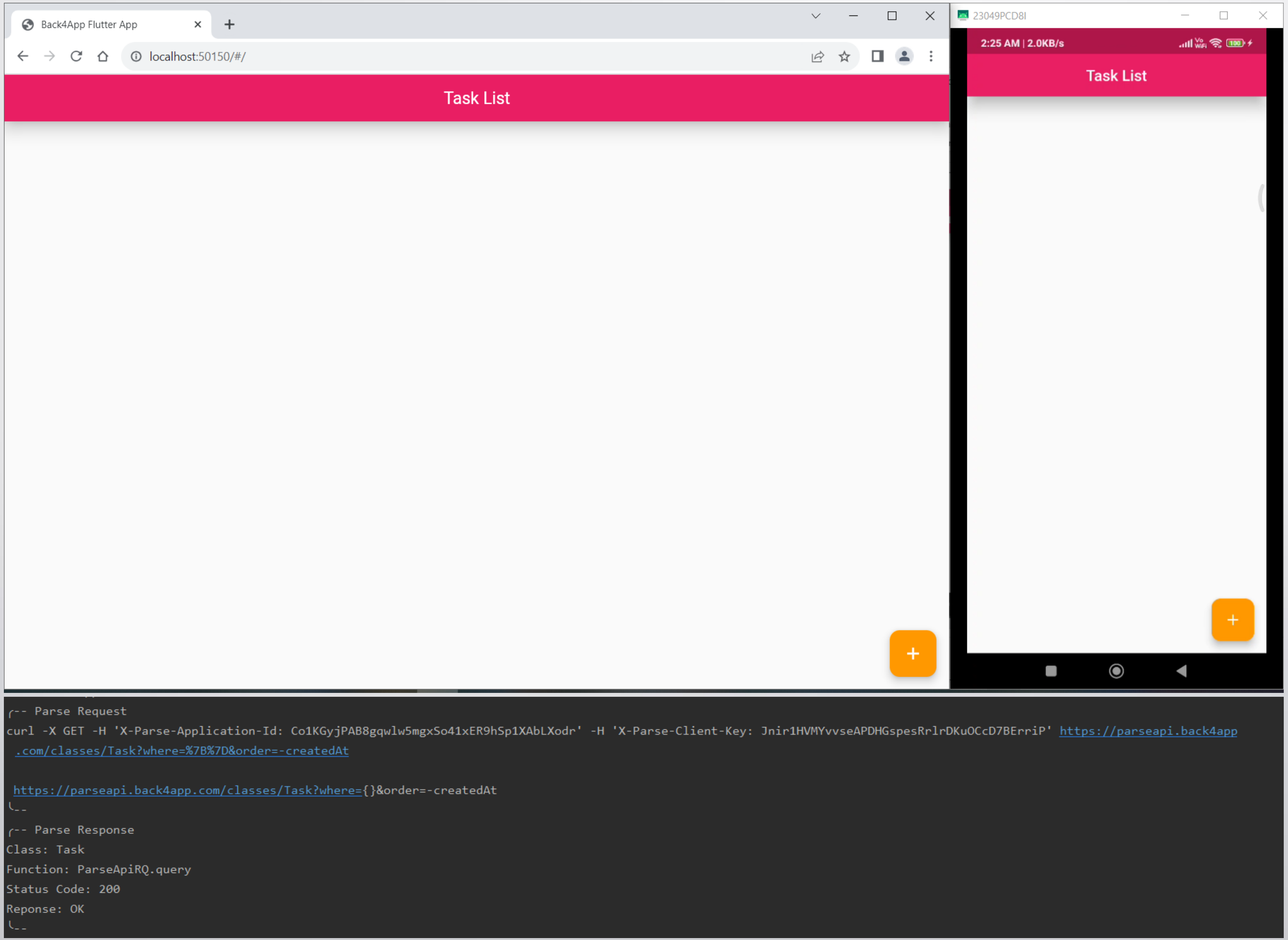
```dart
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Back4App Flutter App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ), // ThemeData
      home: TodoList(),
      debugShowCheckedModeBanner: false,
    ); // MaterialApp
  }
}

class TodoList extends StatefulWidget {
  @override
  _TodoListState createState() => _TodoListState();
}

class _TodoListState extends State<TodoList> {
  late List<ParseObject> tasks = [];

  @override
  void initState() {
    super.initState();
    _loadTasks();
  }

  _loadTasks() async {
    QueryBuilder<ParseObject> queryBuilder =
        QueryBuilder<ParseObject>(ParseObject('Task'))
          ..orderByDescending('createdAt');
    var apiResponse = await queryBuilder.query();
    if (apiResponse.success && apiResponse.results != null) {
      setState(() {
        tasks = List<ParseObject>.from(apiResponse.results!);
```



```dart
              ), // CircleAvatar
              title: Text(...), // Text
              subtitle: Text(
                task.get('description'),
                style: TextStyle(
                  decoration: isCompleted ? TextDecoration.lineThrough : null,
                ), // TextStyle
              ), // Text
              trailing: Row(...), // Row
              onLongPress: () {
                _toggleTaskCompletion(task, !isCompleted);
              },
            ), // ListTile
          ); // Card
        },
      ), // ListView.builder
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          _showAddTaskDialog(context);
        },
        child: Icon(Icons.add),
        elevation: 8.0,
        backgroundColor: Colors.orange,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.all(Radius.circular(12.0)),
        ), // RoundedRectangleBorder
      ), // FloatingActionButton
    ); // Scaffold
  }

  Future<void> _showDeleteAllTasksDialog() async {
    return showDialog(
      context: context,
```

# Task List

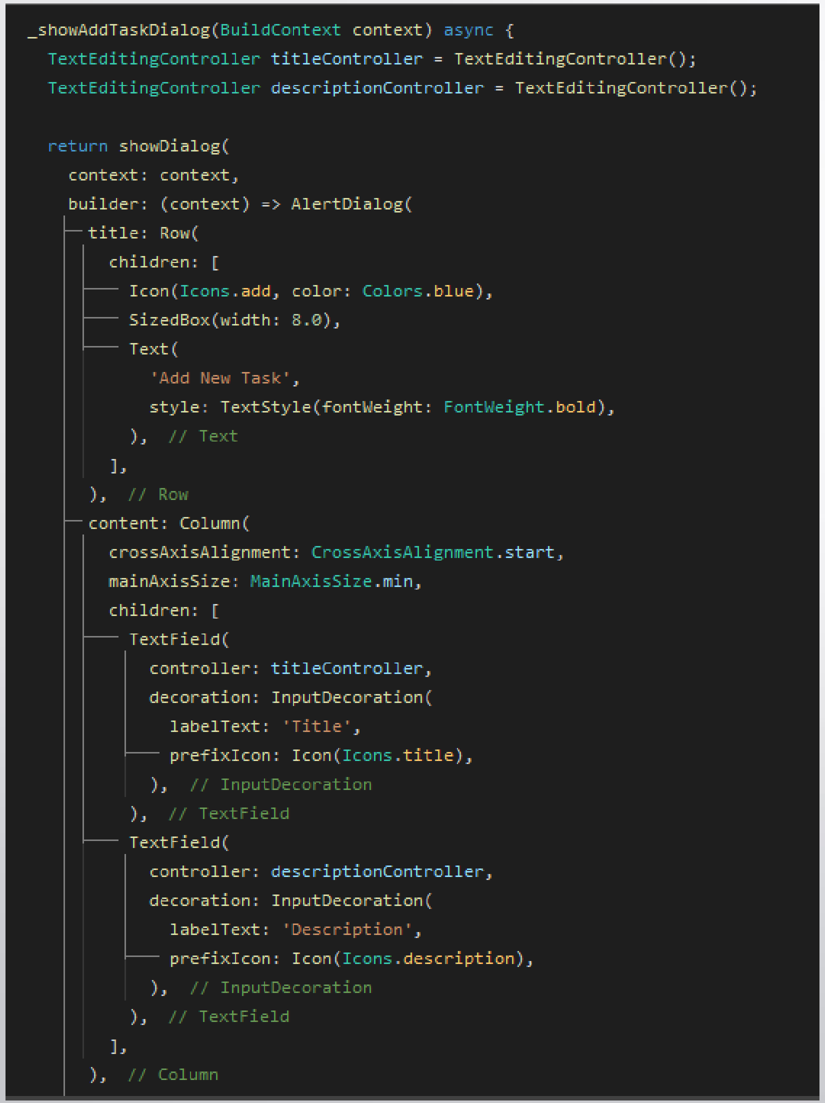
- Output in Web and Android and the Back4App API Logs simultaneously.



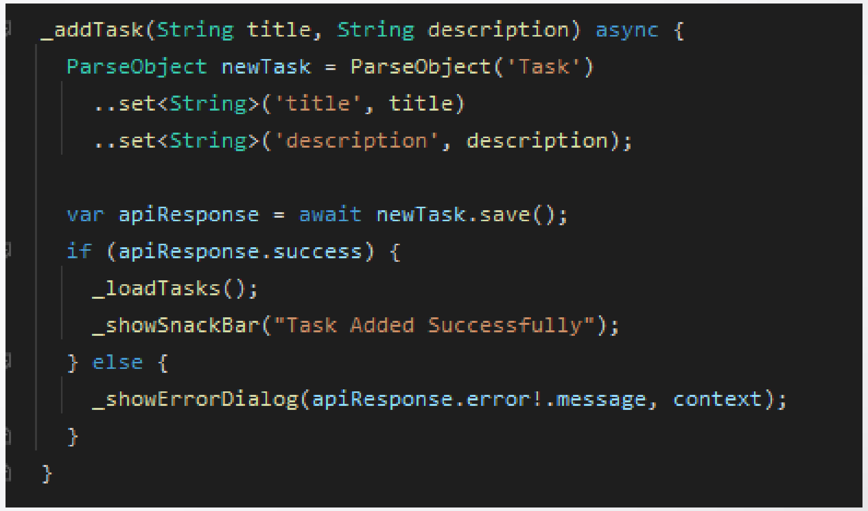
```
╭-- Parse Request
curl -X GET -H 'X-Parse-Application-Id: Co1KGyjPAB8gqwlw5mgxSo41xER9hSp1XAbLXodr' -H 'X-Parse-Client-Key: Jnir1HVMYvvseAPDHGspesRrlrDKuOCcD7BErriP' https://parseapi.back4app
 .com/classes/Task?where=%7B%7D&order=-createdAt

 https://parseapi.back4app.com/classes/Task?where={}&order=-createdAt
╰--
╭-- Parse Response
Class: Task
Function: ParseApiRQ.query
Status Code: 200
Reponse: OK
╰--
```

# Task Creation

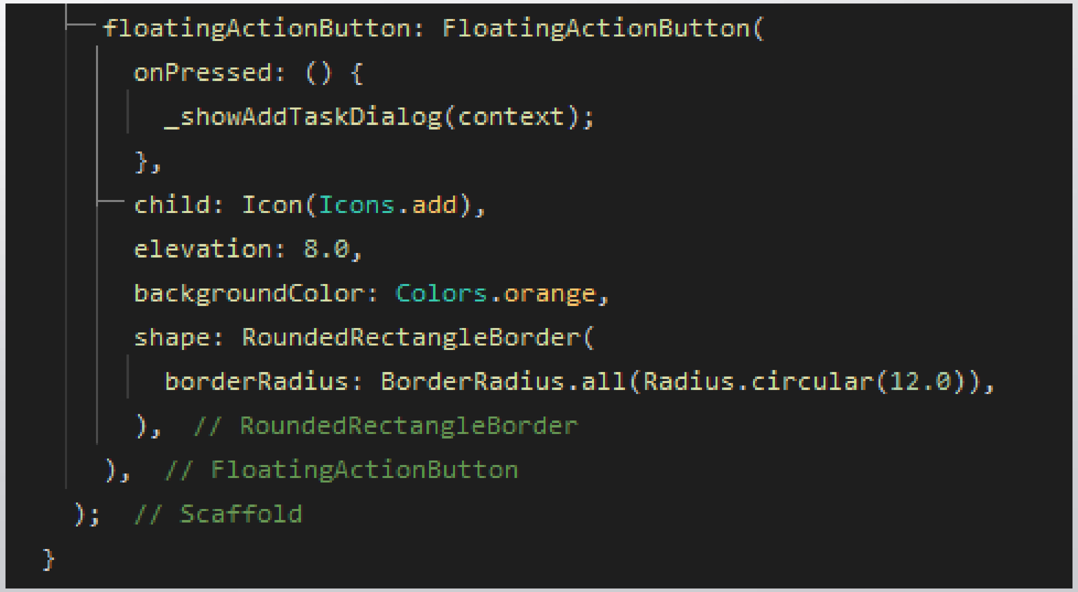
- Create a screen for adding new tasks.
- Implement functionality to create and save tasks to Back4App.
- Verify that newly created tasks appear in the task list.



```
_showAddTaskDialog(BuildContext context) async {
  TextEditingController titleController = TextEditingController();
  TextEditingController descriptionController = TextEditingController();

  return showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Row(
        children: [
          Icon(Icons.add, color: Colors.blue),
          SizedBox(width: 8.0),
          Text(
            'Add New Task',
            style: TextStyle(fontWeight: FontWeight.bold),
          ),  // Text
        ],
      ),  // Row
      content: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: [
          TextField(
            controller: titleController,
            decoration: InputDecoration(
              labelText: 'Title',
              prefixIcon: Icon(Icons.title),
            ),  // InputDecoration
          ),  // TextField
          TextField(
            controller: descriptionController,
            decoration: InputDecoration(
              labelText: 'Description',
              prefixIcon: Icon(Icons.description),
            ),  // InputDecoration
          ),  // TextField
        ],
      ),  // Column
```



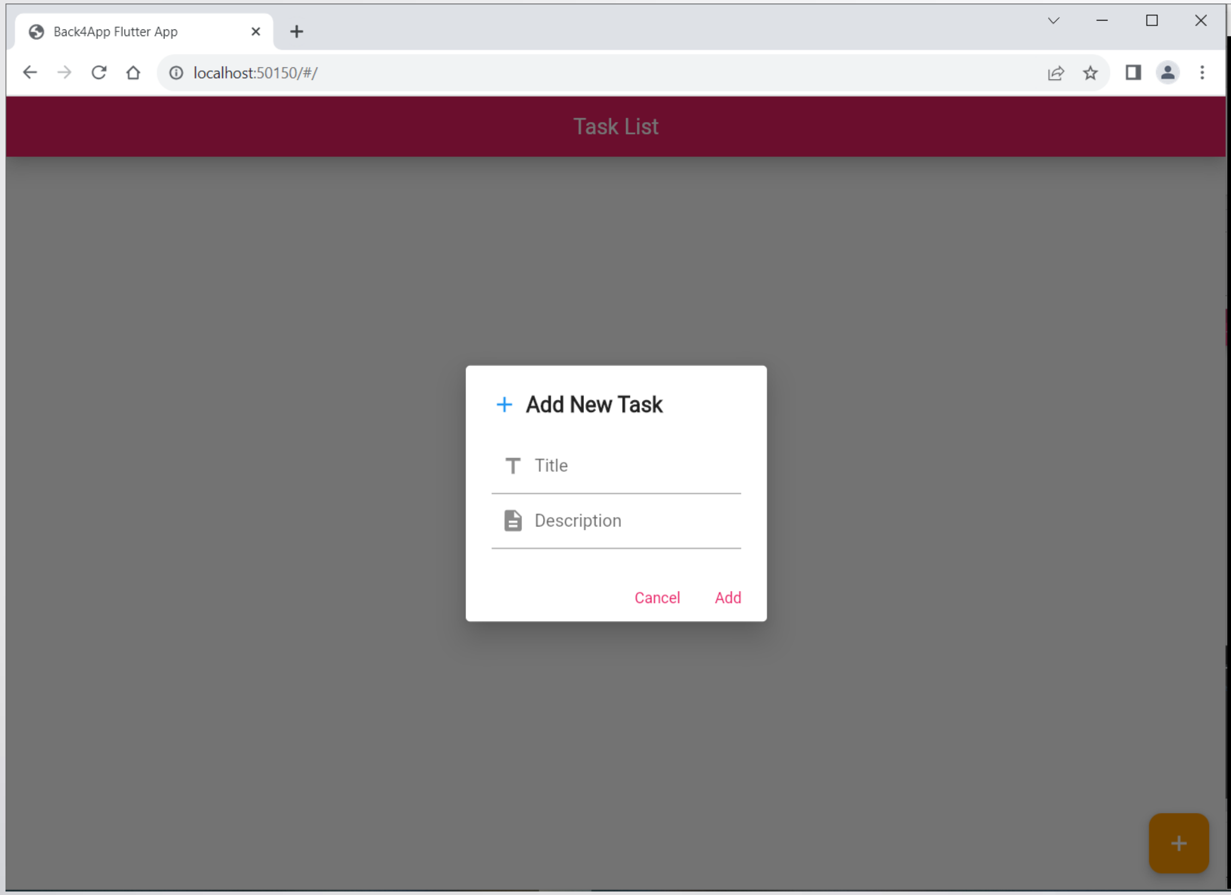
```
_addTask(String title, String description) async {
  ParseObject newTask = ParseObject('Task')
    ..set<String>('title', title)
    ..set<String>('description', description);

  var apiResponse = await newTask.save();
  if (apiResponse.success) {
    _loadTasks();
    _showSnackBar("Task Added Successfully");
  } else {
    _showErrorDialog(apiResponse.error!.message, context);
  }
}
```

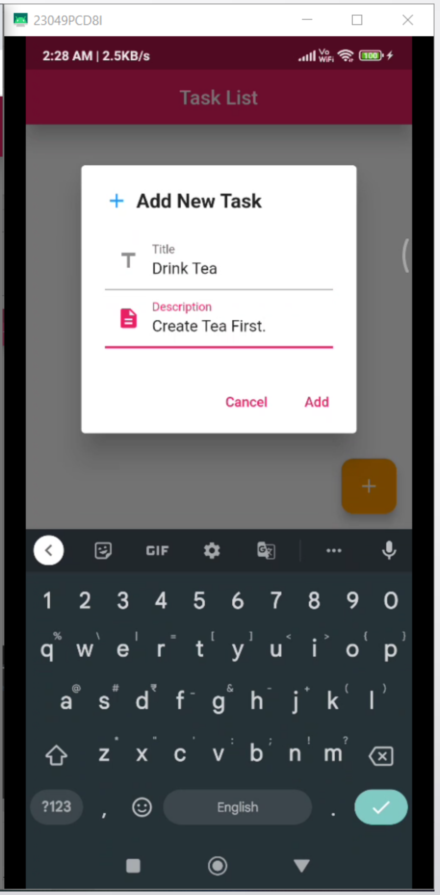


```
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        _showAddTaskDialog(context);
      },
      child: Icon(Icons.add),
      elevation: 8.0,
      backgroundColor: Colors.orange,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.all(Radius.circular(12.0)),
      ),  // RoundedRectangleBorder
    ),  // FloatingActionButton
  );  // Scaffold
}
```

# Task Creation

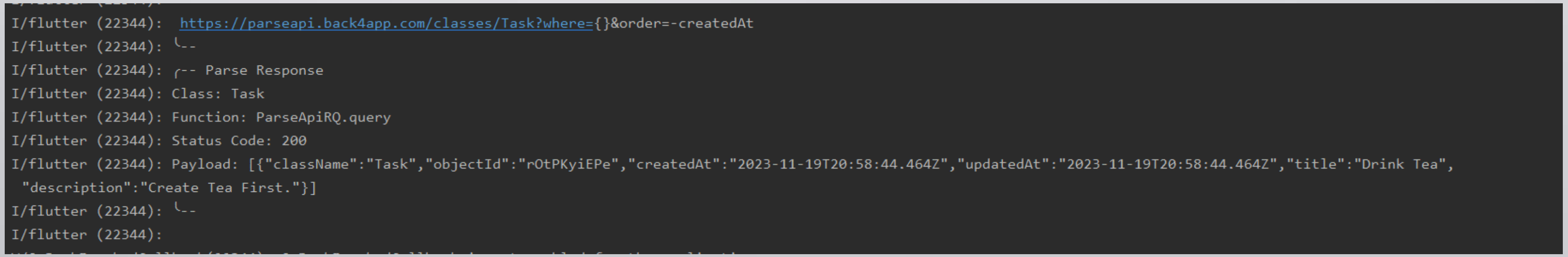
- Output in Web and Android and the Back4App API Logs simultaneously.





Title
Drink Tea

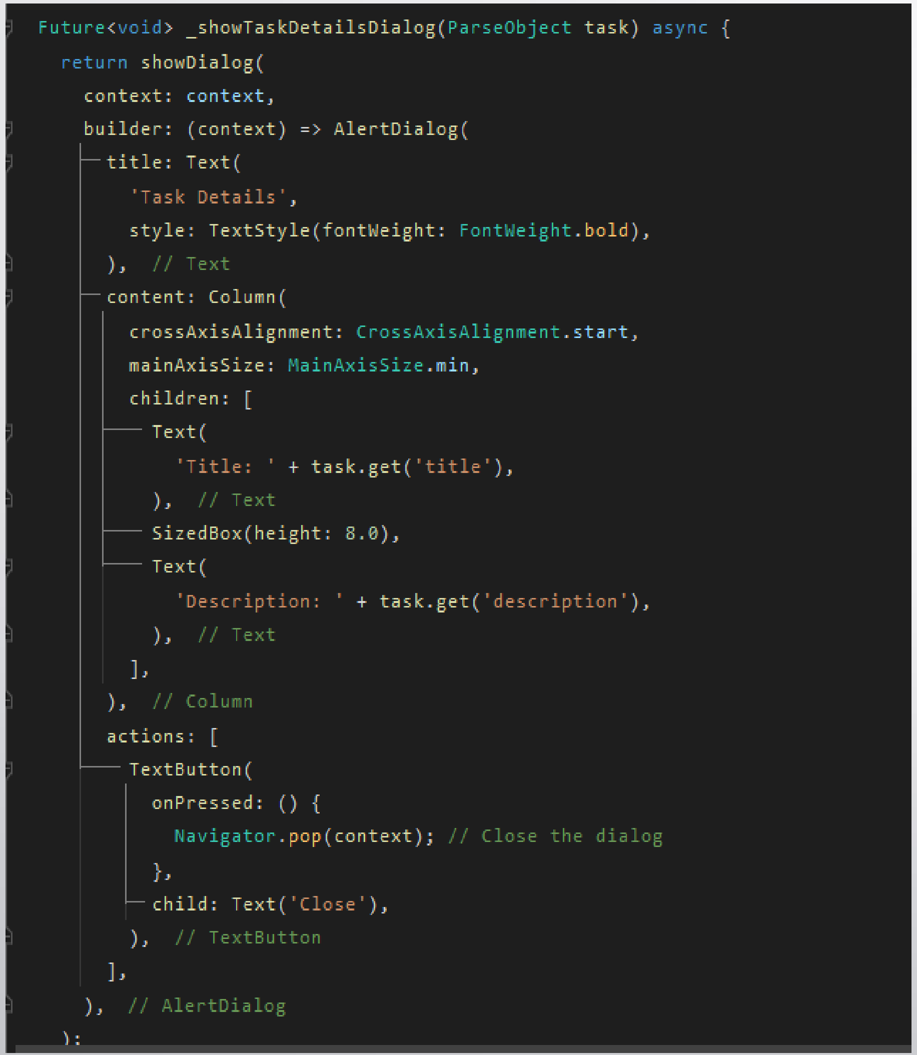
Description
Create Tea First.



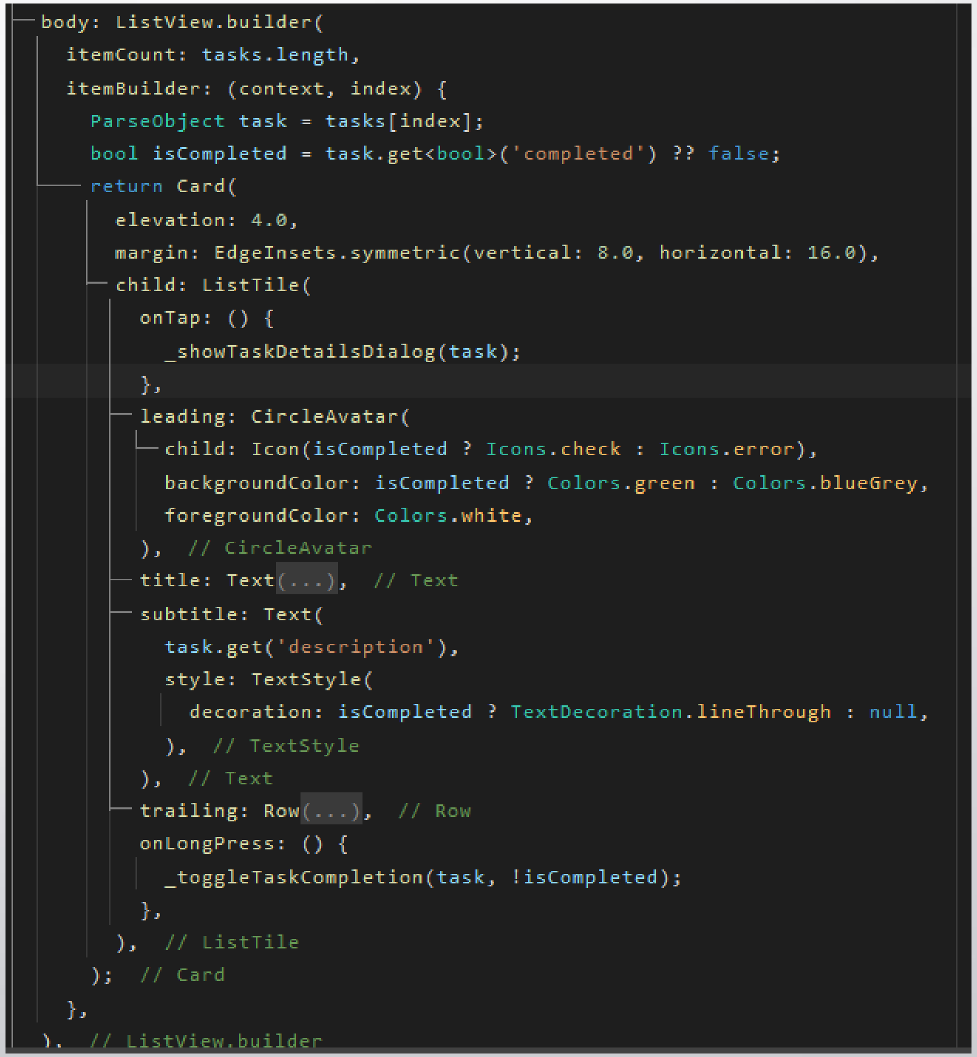
```
I/flutter (22344):  https://parseapi.back4app.com/classes/Task?where={}&order=-createdAt
I/flutter (22344): └--
I/flutter (22344): ┌-- Parse Response
I/flutter (22344): Class: Task
I/flutter (22344): Function: ParseApiRQ.query
I/flutter (22344): Status Code: 200
I/flutter (22344): Payload: [{"className":"Task","objectId":"rOtPKyiEPe","createdAt":"2023-11-19T20:58:44.464Z","updatedAt":"2023-11-19T20:58:44.464Z","title":"Drink Tea",
 "description":"Create Tea First."}]
I/flutter (22344): └--
I/flutter (22344):
```

# Task Details

- Add a feature to view task details when a task is tapped in the task list.
- Display the title and description of the selected task.



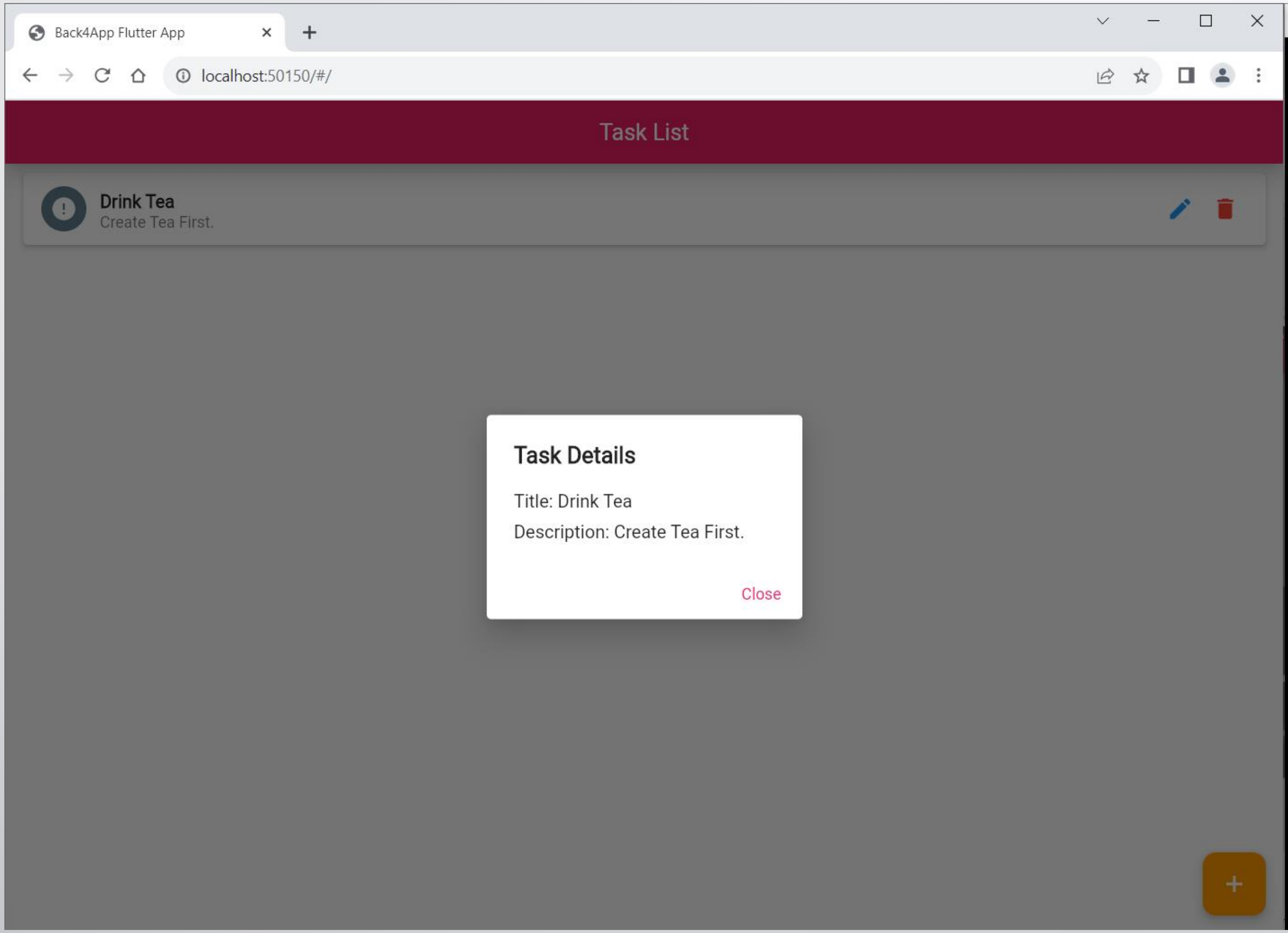
```dart
Future<void> _showTaskDetailsDialog(ParseObject task) async {
  return showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text(
        'Task Details',
        style: TextStyle(fontWeight: FontWeight.bold),
      ), // Text
      content: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: [
          Text(
            'Title: ' + task.get('title'),
          ), // Text
          SizedBox(height: 8.0),
          Text(
            'Description: ' + task.get('description'),
          ), // Text
        ],
      ), // Column
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context); // Close the dialog
          },
          child: Text('Close'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
```



```dart
body: ListView.builder(
  itemCount: tasks.length,
  itemBuilder: (context, index) {
    ParseObject task = tasks[index];
    bool isCompleted = task.get<bool>('completed') ?? false;
    return Card(
      elevation: 4.0,
      margin: EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),
      child: ListTile(
        onTap: () {
          _showTaskDetailsDialog(task);
        },
        leading: CircleAvatar(
          child: Icon(isCompleted ? Icons.check : Icons.error),
          backgroundColor: isCompleted ? Colors.green : Colors.blueGrey,
          foregroundColor: Colors.white,
        ), // CircleAvatar
        title: Text(...), // Text
        subtitle: Text(
          task.get('description'),
          style: TextStyle(
            decoration: isCompleted ? TextDecoration.lineThrough : null,
          ), // TextStyle
        ), // Text
        trailing: Row(...), // Row
        onLongPress: () {
          _toggleTaskCompletion(task, !isCompleted);
        },
      ), // ListTile
    ); // Card
  },
), // ListView.builder
```

# Task Details

- Output in Web and Android simultaneously.



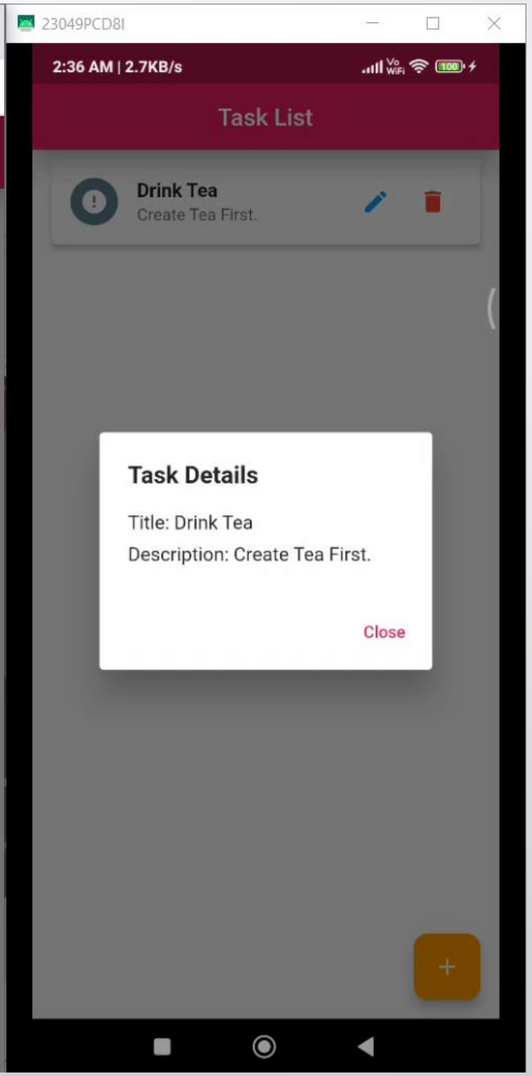
Drink Tea
Create Tea First.

**Task Details**

Title: Drink Tea

Description: Create Tea First.

Close



Drink Tea
Create Tea First.

**Task Details**

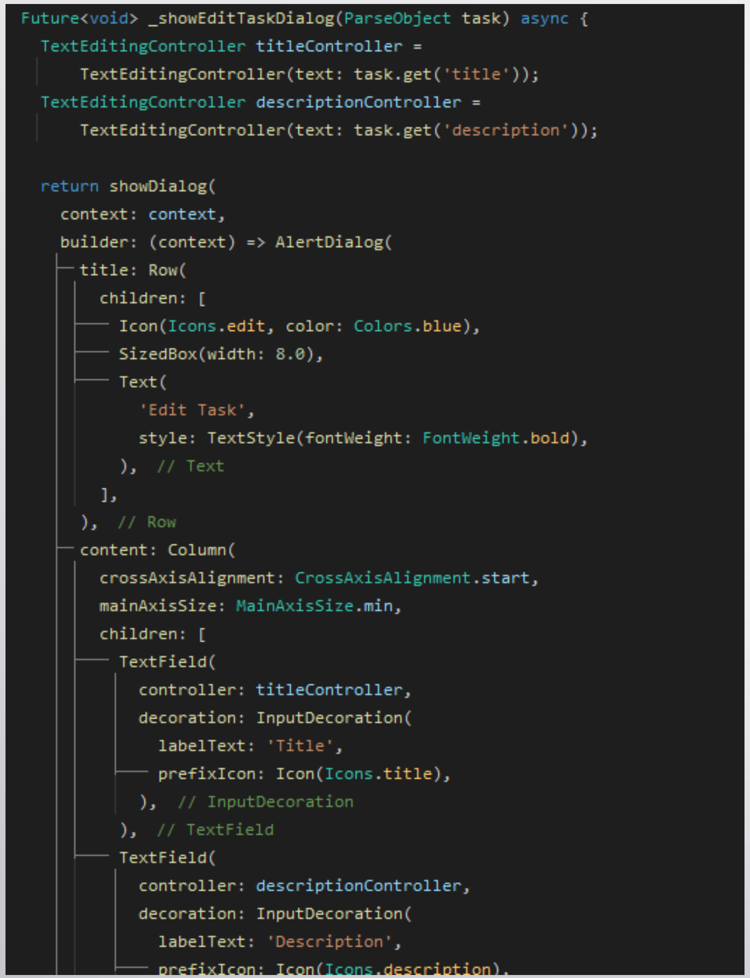
Title: Drink Tea

Description: Create Tea First.

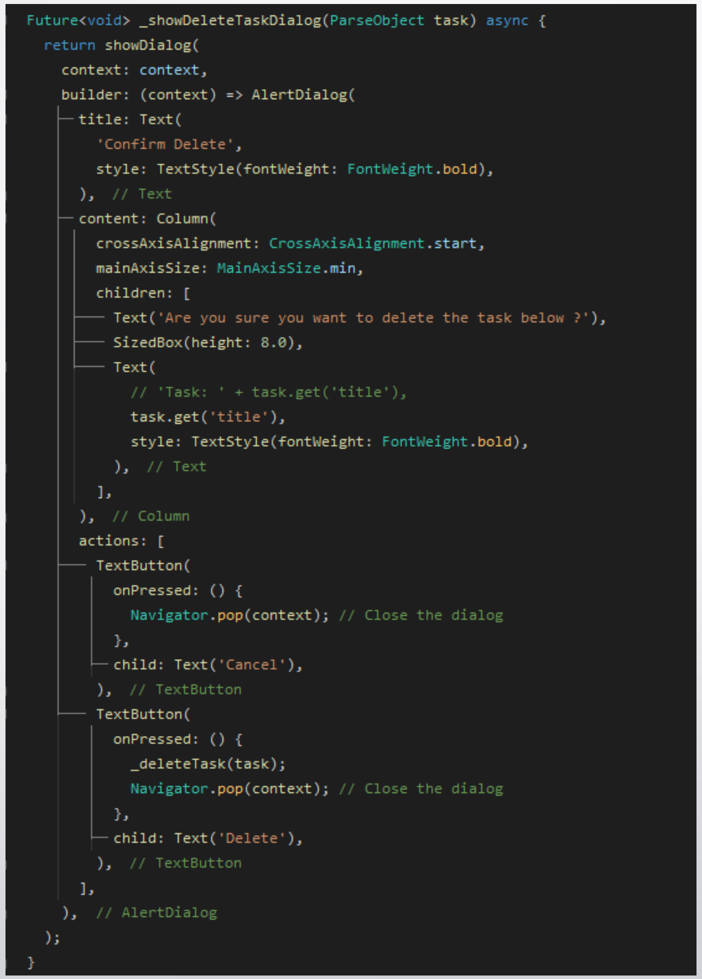
Close

# Bonus Features

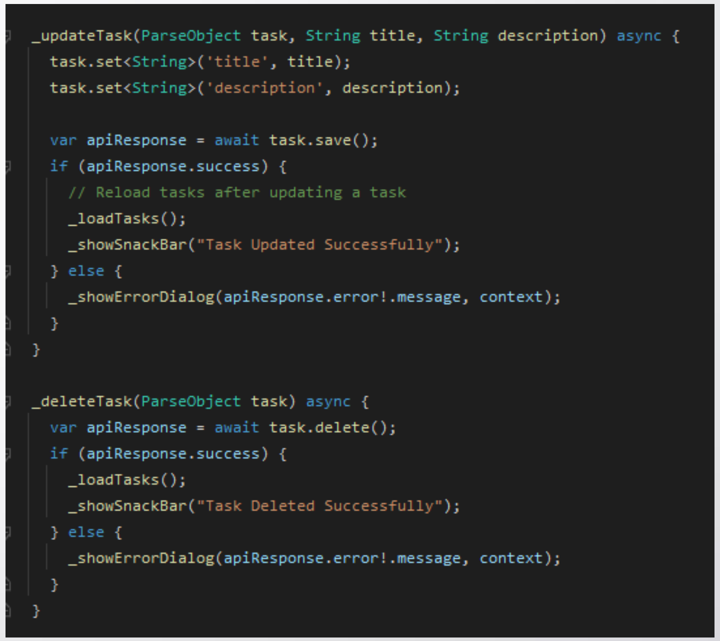
- Add a feature to edit and update existing tasks.
- Implement a feature for task deletion.



```dart
Future<void> _showEditTaskDialog(ParseObject task) async {
  TextEditingController titleController =
      TextEditingController(text: task.get('title'));
  TextEditingController descriptionController =
      TextEditingController(text: task.get('description'));

  return showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Row(
        children: [
          Icon(Icons.edit, color: Colors.blue),
          SizedBox(width: 8.0),
          Text(
            'Edit Task',
            style: TextStyle(fontWeight: FontWeight.bold),
          ), // Text
        ],
      ), // Row
      content: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: [
          TextField(
            controller: titleController,
            decoration: InputDecoration(
              labelText: 'Title',
              prefixIcon: Icon(Icons.title),
            ), // InputDecoration
          ), // TextField
          TextField(
            controller: descriptionController,
            decoration: InputDecoration(
              labelText: 'Description',
              prefixIcon: Icon(Icons.description)
```



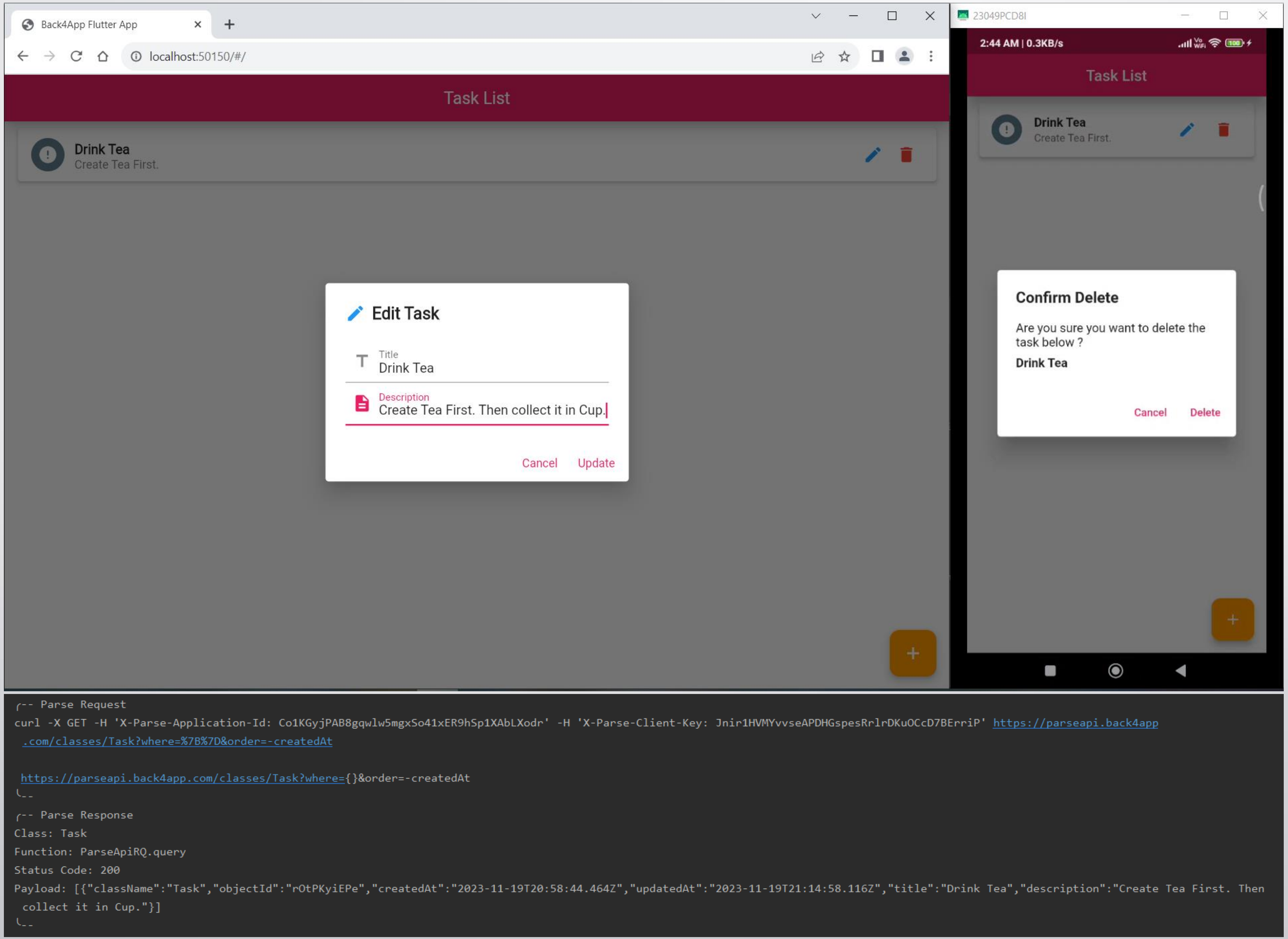
```dart
Future<void> _showDeleteTaskDialog(ParseObject task) async {
  return showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text(
        'Confirm Delete',
        style: TextStyle(fontWeight: FontWeight.bold),
      ), // Text
      content: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: [
          Text('Are you sure you want to delete the task below ?'),
          SizedBox(height: 8.0),
          Text(
            // 'Task: ' + task.get('title'),
            task.get('title'),
            style: TextStyle(fontWeight: FontWeight.bold),
          ), // Text
        ],
      ), // Column
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context); // Close the dialog
          },
          child: Text('Cancel'),
        ), // TextButton
        TextButton(
          onPressed: () {
            _deleteTask(task);
            Navigator.pop(context); // Close the dialog
          },
          child: Text('Delete'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
}
```



```dart
_updateTask(ParseObject task, String title, String description) async {
  task.set<String>('title', title);
  task.set<String>('description', description);

  var apiResponse = await task.save();
  if (apiResponse.success) {
    // Reload tasks after updating a task
    _loadTasks();
    _showSnackBar("Task Updated Successfully");
  } else {
    _showErrorDialog(apiResponse.error!.message, context);
  }
}

_deleteTask(ParseObject task) async {
  var apiResponse = await task.delete();
  if (apiResponse.success) {
    _loadTasks();
    _showSnackBar("Task Deleted Successfully");
  } else {
    _showErrorDialog(apiResponse.error!.message, context);
  }
}
```

# Bonus Features

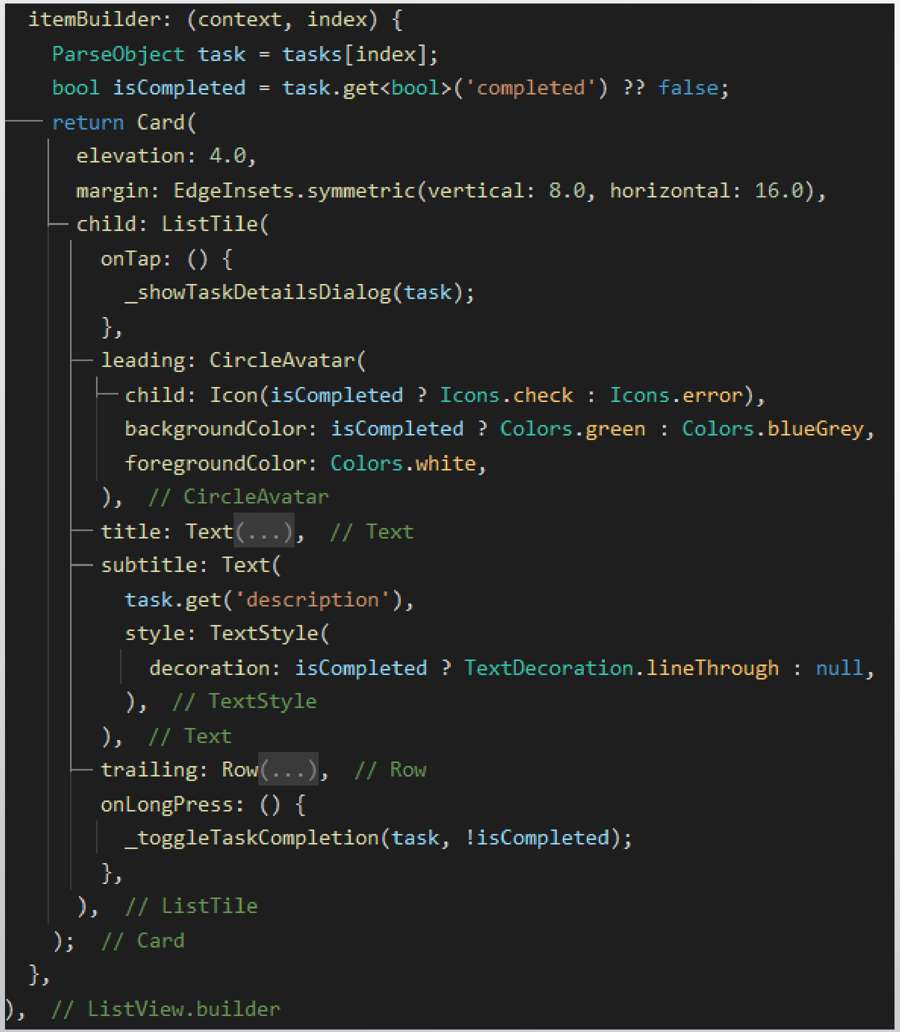
- Output in Web and Android and the Back4App API Logs simultaneously.



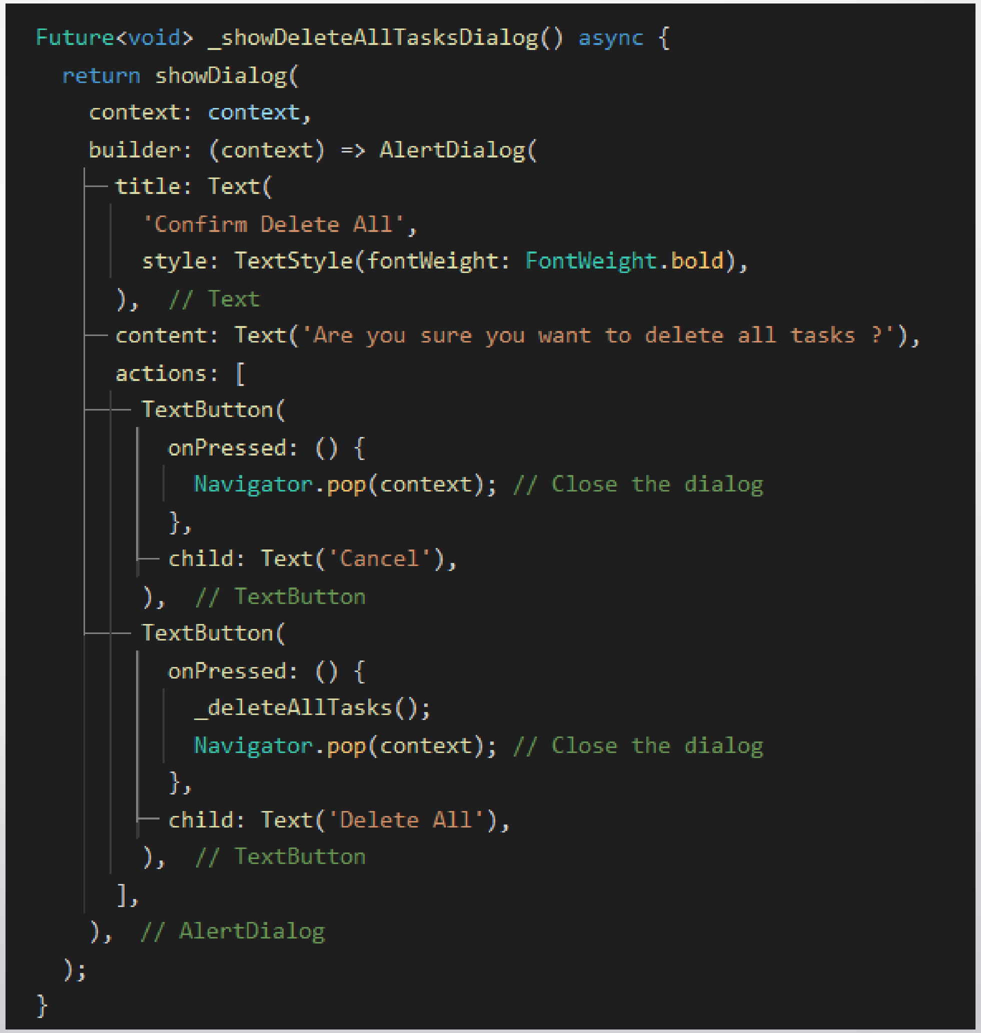
```
╭-- Parse Request
curl -X GET -H 'X-Parse-Application-Id: Co1KGyjPAB8gqwlw5mgxSo41xER9hSp1XAbLXodr' -H 'X-Parse-Client-Key: Jnir1HVMYvvseAPDHGspesRrlrDKuOCcD7BErriP' https://parseapi.back4app
 .com/classes/Task?where=%7B%7D&order=-createdAt

 https://parseapi.back4app.com/classes/Task?where={}&order=-createdAt
╰--
╭-- Parse Response
Class: Task
Function: ParseApiRQ.query
Status Code: 200
Payload: [{"className":"Task","objectId":"rOtPKyiEPe","createdAt":"2023-11-19T20:58:44.464Z","updatedAt":"2023-11-19T21:14:58.116Z","title":"Drink Tea","description":"Create Tea First. Then
 collect it in Cup."}]
╰--
```

# Bonus Features

Add additional features or enhancement which are :
- On Long Press Task will be completed or convert to pending vice-versa.
- Delete all the Task from Top Bar and show the snack bar for all the events performed.



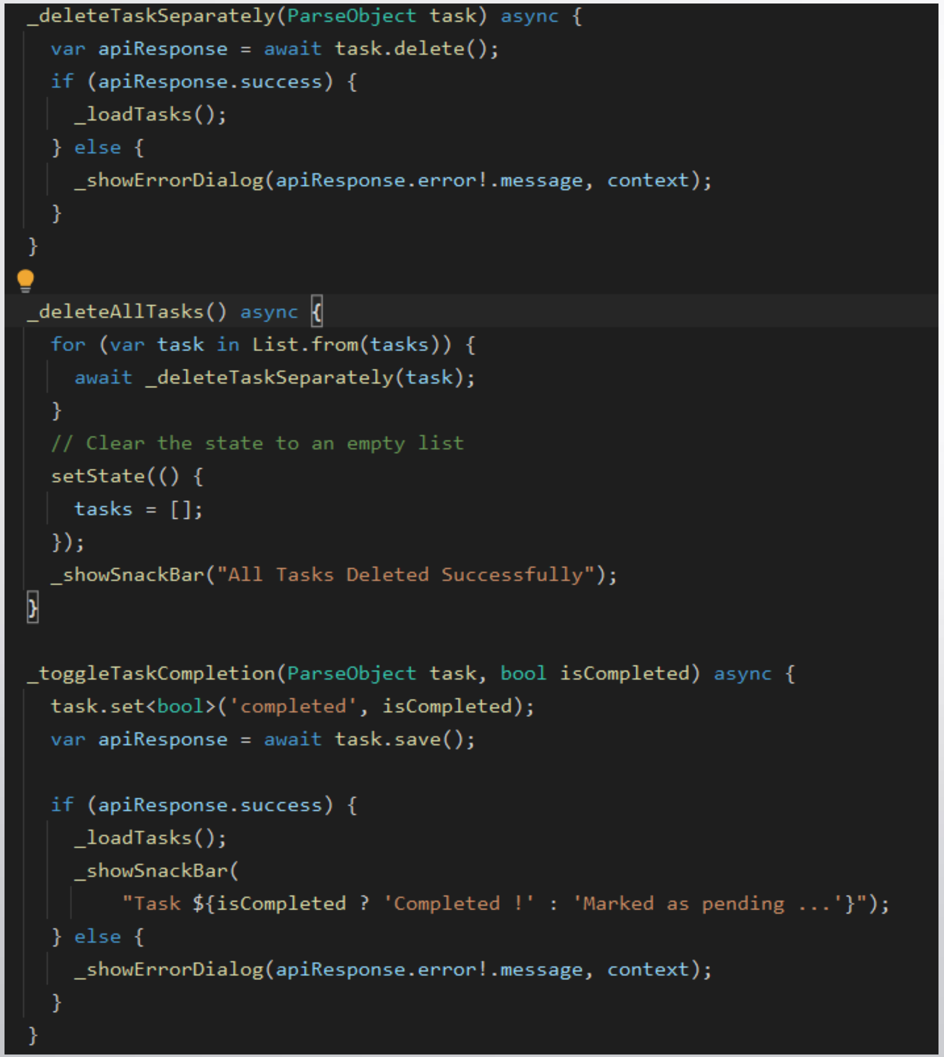
```dart
 itemBuilder: (context, index) {
   ParseObject task = tasks[index];
   bool isCompleted = task.get<bool>('completed') ?? false;
   return Card(
     elevation: 4.0,
     margin: EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),
     child: ListTile(
       onTap: () {
         _showTaskDetailsDialog(task);
       },
       leading: CircleAvatar(
         child: Icon(isCompleted ? Icons.check : Icons.error),
         backgroundColor: isCompleted ? Colors.green : Colors.blueGrey,
         foregroundColor: Colors.white,
       ), // CircleAvatar
       title: Text(...), // Text
       subtitle: Text(
         task.get('description'),
         style: TextStyle(
           decoration: isCompleted ? TextDecoration.lineThrough : null,
         ), // TextStyle
       ), // Text
       trailing: Row(...), // Row
       onLongPress: () {
         _toggleTaskCompletion(task, !isCompleted);
       },
     ), // ListTile
   ); // Card
 },
), // ListView.builder
```



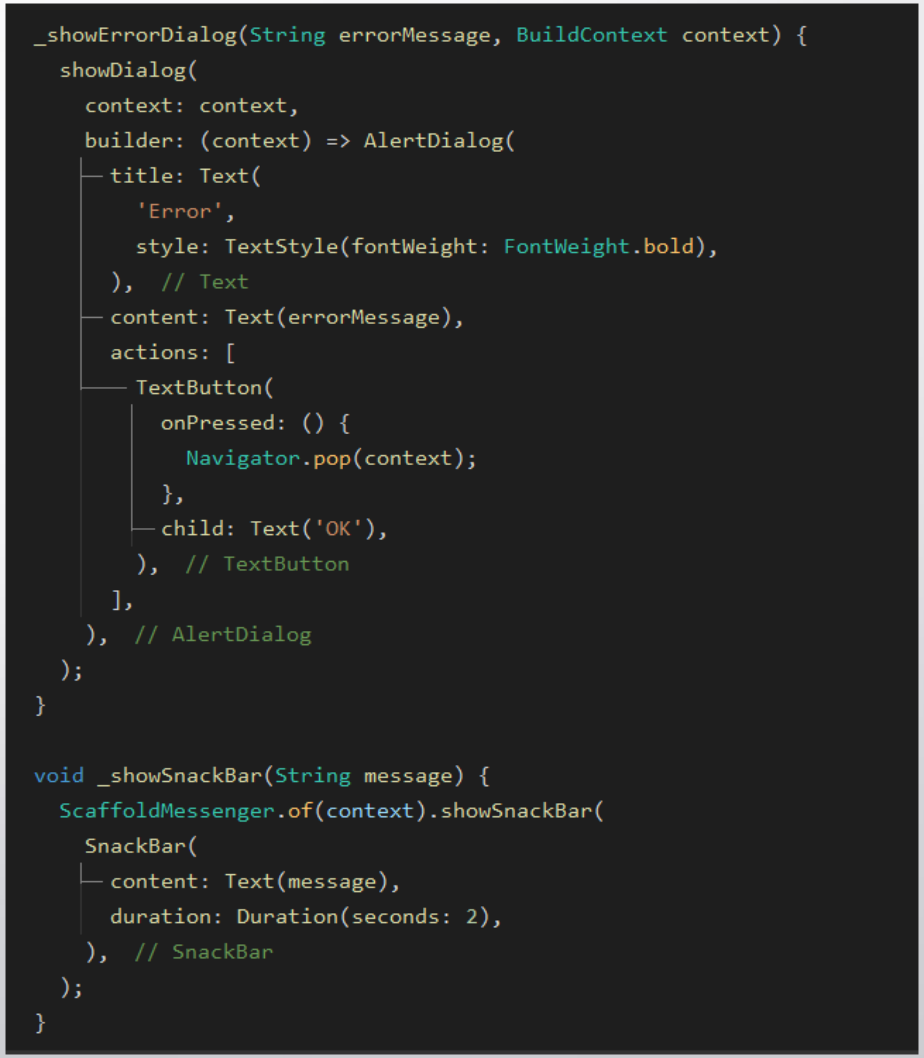
```dart
Future<void> _showDeleteAllTasksDialog() async {
  return showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text(
        'Confirm Delete All',
        style: TextStyle(fontWeight: FontWeight.bold),
      ), // Text
      content: Text('Are you sure you want to delete all tasks ?'),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context); // Close the dialog
          },
          child: Text('Cancel'),
        ), // TextButton
        TextButton(
          onPressed: () {
            _deleteAllTasks();
            Navigator.pop(context); // Close the dialog
          },
          child: Text('Delete All'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
}
```

# Bonus Features

Add additional features or enhancement which are :
- On Long Press Task will be completed or convert to pending vice-versa.
- Delete all the Task from Top Bar and show the snack bar for all the events performed.



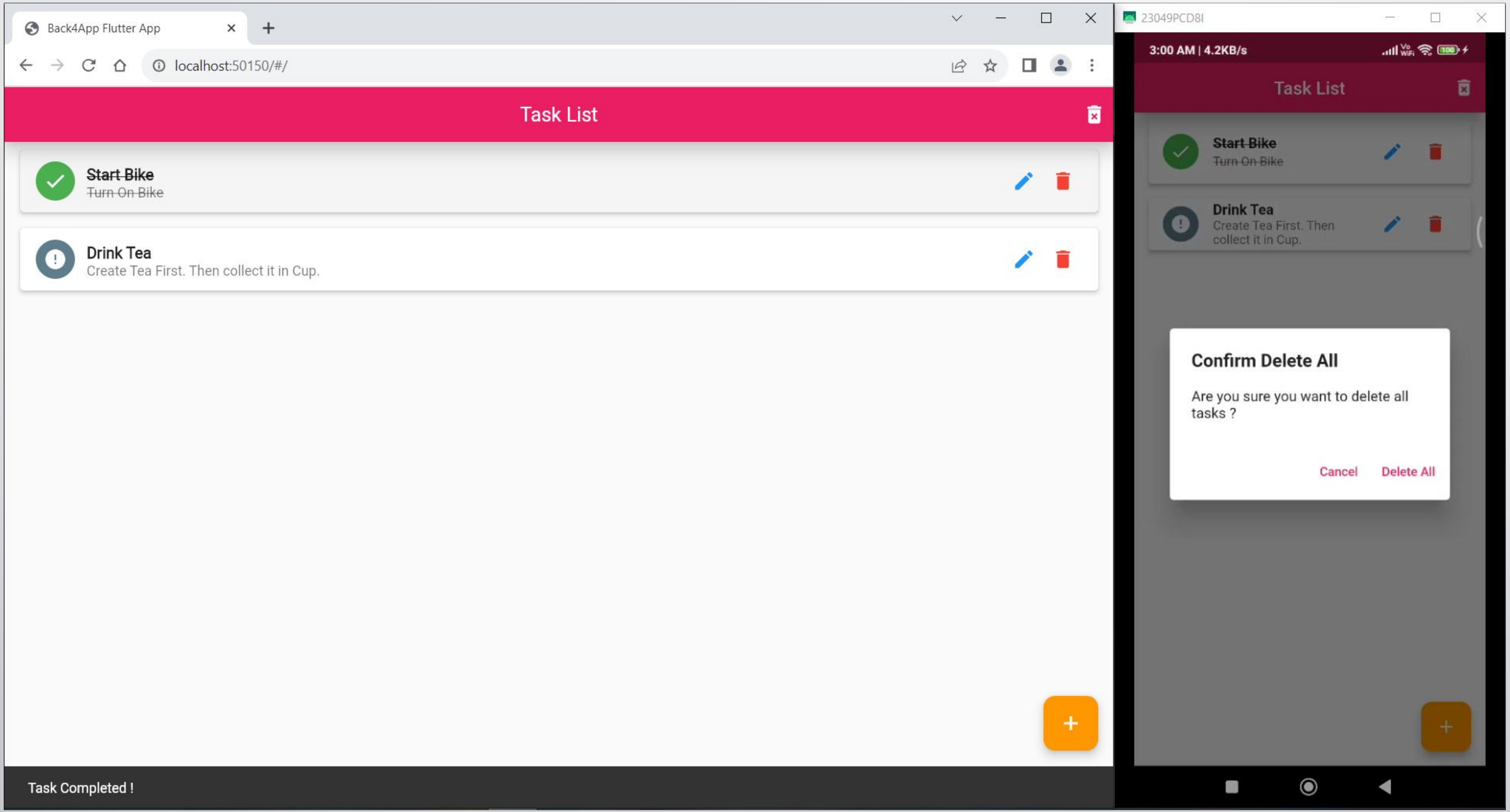
```
_deleteTaskSeparately(ParseObject task) async {
  var apiResponse = await task.delete();
  if (apiResponse.success) {
    _loadTasks();
  } else {
    _showErrorDialog(apiResponse.error!.message, context);
  }
}

_deleteAllTasks() async {
  for (var task in List.from(tasks)) {
    await _deleteTaskSeparately(task);
  }
  // Clear the state to an empty list
  setState(() {
    tasks = [];
  });
  _showSnackBar("All Tasks Deleted Successfully");
}

_toggleTaskCompletion(ParseObject task, bool isCompleted) async {
  task.set<bool>('completed', isCompleted);
  var apiResponse = await task.save();

  if (apiResponse.success) {
    _loadTasks();
    _showSnackBar(
        "Task ${isCompleted ? 'Completed !' : 'Marked as pending ...'}");
  } else {
    _showErrorDialog(apiResponse.error!.message, context);
  }
}
```



```
_showErrorDialog(String errorMessage, BuildContext context) {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text(
        'Error',
        style: TextStyle(fontWeight: FontWeight.bold),
      ), // Text
      content: Text(errorMessage),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('OK'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
}

void _showSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(message),
      duration: Duration(seconds: 2),
    ), // SnackBar
  );
}
```

# Bonus Features

- Output in Web and Android simultaneously.



Task List

~~Start Bike~~
~~Turn On Bike~~

Drink Tea
Create Tea First. Then collect it in Cup.

Task Completed !

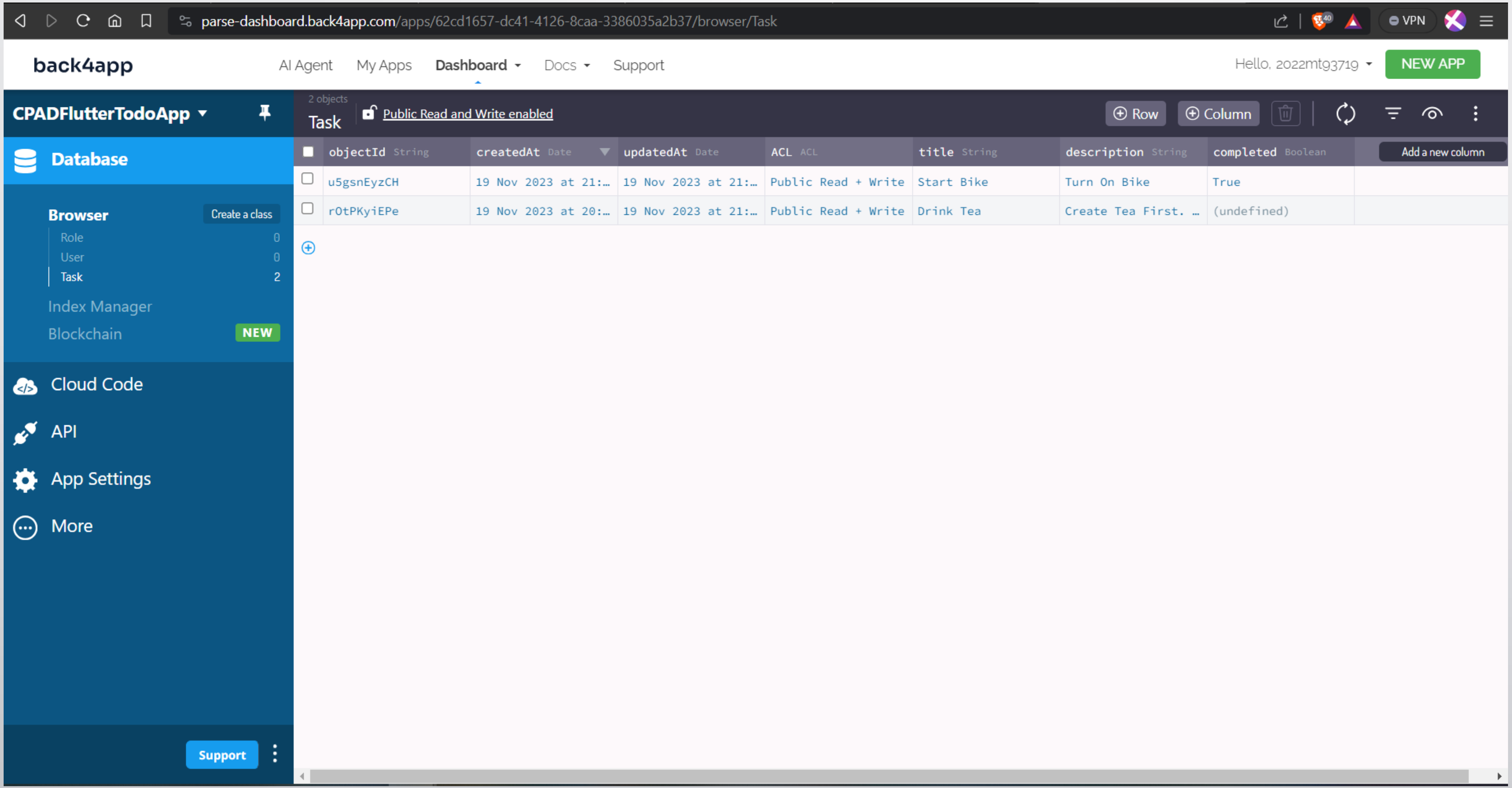
**Confirm Delete All**

Are you sure you want to delete all tasks ?

Cancel Delete All

# Back4App Server Console

- Live Data in columns objectId, createdAt, updatedAt, title, description, completed in Back4App Server console.



| objectId String | createdAt Date | updatedAt Date | ACL ACL | title String | description String | completed Boolean |
|---|---|---|---|---|---|---|
| u5gsnEyzCH | 19 Nov 2023 at 21:… | 19 Nov 2023 at 21:… | Public Read + Write | Start Bike | Turn On Bike | True |
| rOtPKyiEPe | 19 Nov 2023 at 20:… | 19 Nov 2023 at 21:… | Public Read + Write | Drink Tea | Create Tea First. … | (undefined) |

## HELPER LINKS

### REFERENCES

- https://docs.flutter.dev/get-started/install/windows
- https://parse-dashboard.back4app.com/
- https://www.back4app.com/docs/flutter/parse-sdk/parse-flutter-sdk
- https://docs.flutter.dev/ui/layout

**GITHUB REPOSITORY LINK:**
https://github.com/TheKenilKDoshi/Cross_Platform_App_Development_Flutter_TODO_Back4App

**ANDROID APK LINK:** 2022MT93719_KenilDoshi_FlutterTODOBack4App.apk
**or**
https://drive.google.com/file/d/14nfQlNaozsjl5qPJ1LIZZuor2ncxx8J7/view?usp=drive_link

THANK YOU!!