Documentation: Getting Public IP, Encoding Data, and Calling Student Validation API

This documentation will guide you through the process of obtaining the current device's public IP, converting it to MD5, manipulating the user ID, and calling an API to validate the student and receive a token.

1. Prerequisites

- Dart
- Flutter
- `http` package
- `crypto` package

Ensure you have the required packages in your `pubspec.yaml` file:

```
dependencies:
  flutter:
    sdk: flutter
  http: ^0.13.3
  crypto: ^3.0.1
```

2. Implementation Steps

Step 1: Get Public IP

Create a function to fetch the current device's public IP:

```
import 'dart:convert';
import 'package:http/http.dart' as http;

Future<String> getPublicIP() async {
  final response = await http.get(Uri.parse('https://api.ipify.org?format=json'));
  if (response.statusCode == 200) {
    final ip = jsonDecode(response.body)['ip'];
    return ip;
  } else {
    throw Exception('Failed to get public IP');
  }
}
```

Step 2: Convert IP to MD5

Create a function to convert the IP address to an MD5 hash:

```
import 'package:crypto/crypto.dart';
import 'dart:convert';

String convertToMD5(String input) {
  return md5.convert(utf8.encode(input)).toString();
}
```

Step 3: Encode User ID

Create a function to encode the user ID by multiplying it by 345:

```
String encodeUserID(int userID) {
  return (userID * 345).toString();
}
```

Step 4: Call the Student Validation API

Create a function to call the API with the encoded IP and user ID:

```
Future<String> validateStudent(String encodedIP, String encodedUserID) async {
  final response = await http.post(
    Uri.parse('http://jobs.skillssprint.com/api/student_validation'),
    headers: {
      'Content-Type': 'application/json',
    },
    body: jsonEncode({
      'encoded_ip': encodedIP,
      'encoded_userid': encodedUserID,
    }),
  );

  if (response.statusCode == 200) {
    final responseBody = jsonDecode(response.body);
    return responseBody['token'];
  } else {
    throw Exception('Failed to validate student');
  }
}
```

Step 5: Integrate All Functions

Create a function to integrate all the steps and get the token:

```
Future<String> getToken(int userID) async {
  String ip = await getPublicIP();
  String encodedIP = convertToMD5(ip);
  String encodedUserID = encodeUserID(userID);
  String token = await validateStudent(encodedIP, encodedUserID);
  return token;
}
```

Example Usage

Use the following example to get the token by providing a user ID:

```
dart
Copy code
void main() async {
  int userID = 123; // Replace with actual user ID
  try {
    String token = await getToken(userID);
    print('Token: $token');
  } catch (e) {
    print('Error: $e');
  }
}
```

Summary

- **Get Public IP:** Fetches the current device's public IP.
- **Convert to MD5:** Converts the IP to an MD5 hash.
- **Encode User ID:** Multiplies the user ID by 345.
- **Call Validation API:** Sends a POST request with the encoded IP and user ID.
- **Get Token:** Integrates all steps to get the token.

By following these steps, you will be able to fetch the public IP, encode the required data, and call the student validation API to receive a token.