Sure, here is an example of Dart/Flutter code for uploading an image to Azure Blob Storage using the REST API:

Dart

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

void uploadImageToAzureBlobStorage(File imageFile) async {
  final storageAccountName = 'YOUR_STORAGE_ACCOUNT_NAME';
  final containerName = 'YOUR_CONTAINER_NAME';
  final blobName = imageFile.path.split('/').last;
  final accountKey = 'YOUR_ACCOUNT_KEY';

  final uri = Uri.parse('https://${storageAccountName}.blob.core.windows.net/${containerName}/${blobName}');

  final sasToken = generateSasToken(accountKey, uri.queryParameters);

  final request = http.MultipartRequest('PUT', uri);
  request.headers['Authorization'] = 'Bearer $sasToken';
  request.files.add(http.MultipartFile('image', imageFile.readAsBytes()));

  final response = await request.send();

  if (response.statusCode == 201) {
    print('Image uploaded successfully');
  } else {
    print('Error uploading image: ${response.statusCode}');
  }
}

String generateSasToken(String accountKey, String queryParameters) {
  final encodedHmacSha256 = hmacSha256(accountKey, queryParameters);
  final encodedBase64 = base64Encode(encodedHmacSha256);

  return 'Bearer ${encodedBase64}';
}

String hmacSha256(String key, String data) {
  final hmac = Hmac(sha256, utf8.encode(key));
  final digest = hmac.convert(utf8.encode(data));
  return digest.toString();
}
```

To use this code, you will need to replace YOUR_STORAGE_ACCOUNT_NAME, YOUR_CONTAINER_NAME, YOUR_ACCOUNT_KEY, and imageFile with your own values.

Here is an example of how to call the uploadImageToAzureBlobStorage function:

Dart

```dart
final imageFile = File('/path/to/image.jpg');
uploadImageToAzureBlobStorage(imageFile);
```

This will upload the image image.jpg to the Azure Blob Storage container YOUR_CONTAINER_NAME in the storage account YOUR_STORAGE_ACCOUNT_NAME.

I hope this helps!