INSTINCT INSTITUTE

ICT Department

ទាញយកទិន្នន័យពី Internet ជាមួយនឹង http, APIs, និង JSON

Instructor: **Oum Saokosal**, *Master of Engineering in Information Systems, South Korea '2010*

Email: oumsaokosal@gmail.com

Phone: 012 252 752 (Telegram)

ទាញយកទិន្នន័យពី Internet ជាមួយនឹង http, APIs, និង JSON

- ស្ទើរតែគ្រប់ app ទាំងអស់នាពេលបច្ចុប្បន្ន គឺតែងតែទាញយកទិន្នន័យចេញពី Internet។ Flutter បានផ្តល់នូវ library មួយយ៉ាងល្អសំរាប់ទាញយកទិន្នន័យចេញពី website ណាមួយ ដែល library នោះគឺឈ្មោះថា http។
- តាមរយៈ http នេះ, យើងក៏អាចទាញយក Web API ចេញពី website ផងដែរ។

របៀបប្រើ http

1. ដាក់ http នៅក្នុង file pubspect.yaml៖

```
dependencies:
  http: ^0.12.0+2
```

បញ្ជាក់៖សូមមើលពី version ចុងក្រោយរបស់ http នៅក្នុង៖

https://pub.dev/packages/http

2. run cmd:

```
flutter packages get
```

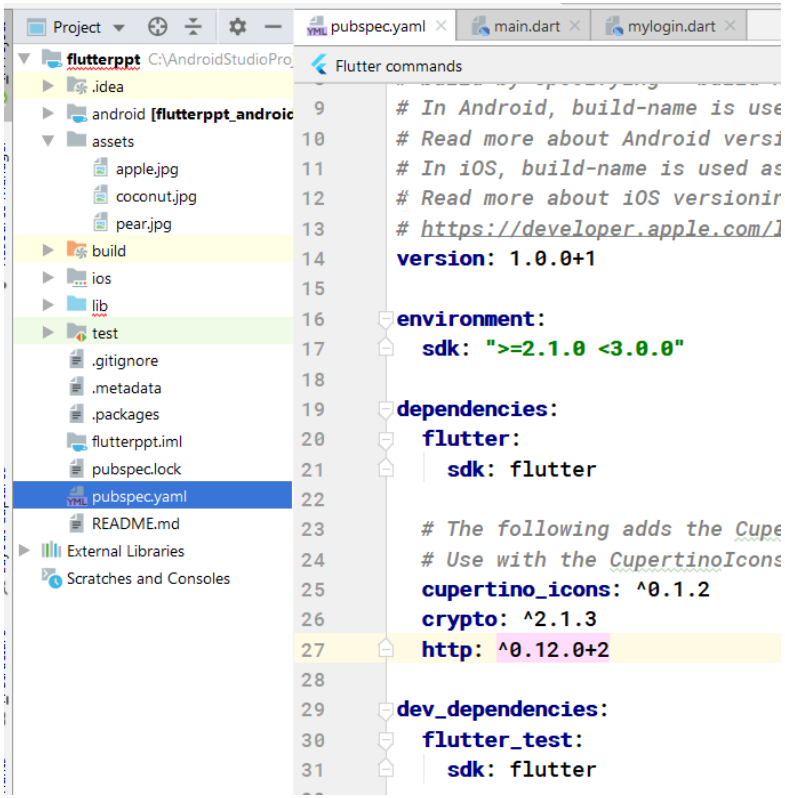
3. បន្ទាប់មក យើងអាច Import http:

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

4. សាកល្បងទាញទិន្នន័យពី website ណាមួយ:

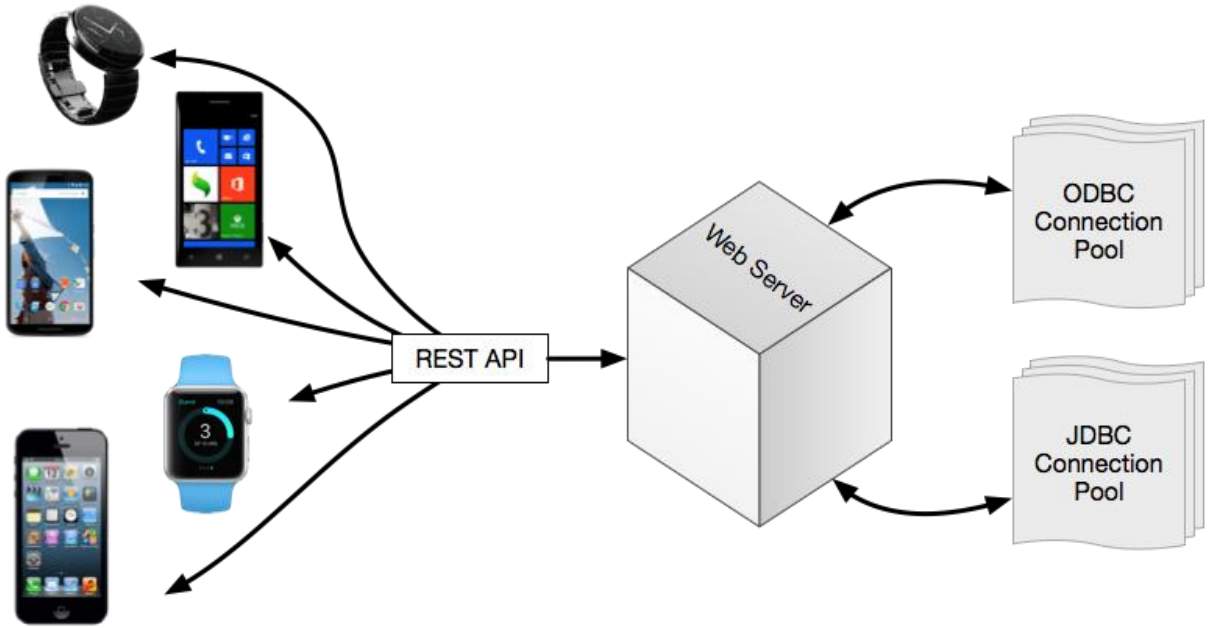
```
Future readData() async {
  http.Response data = await http.get('https://google.com');
  print(data.body);
}
```

```
readData();
```



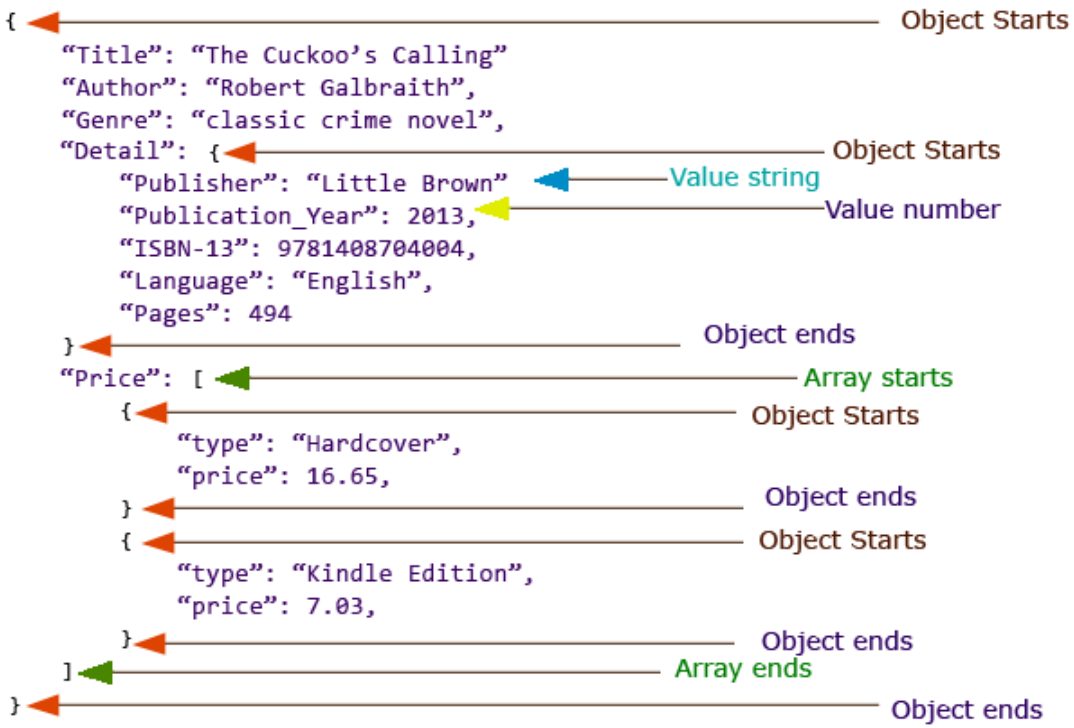
ទាញយកទិន្នន័យ API ទំរង់ជា JSON

- Smartphone មិនអាចទាញយក database ដែលមានប្រភេទជា relational database **ដោយផ្ទាល់ចេញ** ពី database server ដូចជា SQL Server, MySQL ឬ Oracle បានទេ។ ព្រោះវាមិនអាចប្រើ ADO .NET, JDBC ឬ ODBC ទៅភ្ជាប់នឹង database server ទាំងនោះបានឡើយ។
- ដើម្បីដោះស្រាយបញ្ហានេះ គេបានបង្កើតជា Web API ឬហៅកាត់ថា API។
- API វាដើរតួរជាអ្នកកណ្តាលរវាងទូរសព្ទ និង Web Server។ យើងចង់ read/write គឺយើងត្រូវឆ្លងកាត់ API សិន។ បន្ទាប់មក API បោះទិន្នន័យបន្តទៅ Web Server ហើយបោះបន្តទៅ Database ណាមួយ។



ទាញយកទិន្នន័យ API ទំរង់ជា JSON

- សំរាប់ការធ្វើ API វិញគឺអាស្រ័យទៅតាមសមត្ថភាពរបស់យើងម្នាក់ៗ។ យើងប្រើភាសាអ្វីក៏អាចធ្វើ API បានដែរ ដូចជា Java, C#, Python, PHP។
- API មានច្រើនប្រភេទ ដែលពីមុនគេប្រើភាសា XML។ តែឥឡូវនេះគេនិយមប្រើ Restful API តែប៉ុណ្ណោះ ហើយភាសាដែលប្រើគឺ JSON (JavaScript Object Notation)។



ឧទាហរណ៍នៃការទាញ API (1)

ខាងក្រោមនេះគឺជាឧទាហរណ៍នៃការទាញ API ចេញពី url មួយ៖

https://jsonplaceholder.typicode.com/posts?userId=5

1. ដាក់ http នៅក្នុង file pubspect.yaml៖ (*https://pub.dev/packages/http*)

```
dependencies:
  http: ^0.12.0+2
```

2. run cmd:

```
flutter packages get
```

3. បន្ទាប់មក យើងអាច Import http:

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

```
import 'dart:convert';
```

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

ឧទាហរណ៍នៃការទាញ API (2)

4. យើងត្រូវពិនិត្យមើល structure របស់ API សិន

https://jsonplaceholder.typicode.com/posts?userId=5


```
[
  {
    "userId": 5,
    "id": 41,
    "title": "non est facere",
    "body": "molestias id nostrum\nexcepturi repellendus..."
  },
  {
    "userId": 5,
    "id": 42,
    "title": "commodi ullam sint error explicabo praesentium voluptas",
    "body": "odio fugit voluptatum ducimus earum autem est..."
  },
  ...
]
```

ឧទាហរណ៍នៃការទាញ API (3)

5. យើងសង្កេតឃើញថា៖

5.1. JSON នេះមានទំរង់ចាប់ផ្តើមចេញ [ ] គឺជា List នៅក្នុង Flutter

5.2. បន្ទាប់មកបាន {} ដែលជា Map នៅក្នុង Flutter

```
{
    "userId": 5,
    "id": 41,
    "title": "non est facere",
    "body": "molestias id nostrum\nexcepturi repellendus..."
  }
```

5.3. នៅក្នុង map នេះគឺមាន attribute ៤៖

- userId គឺជា integer
- id គឺជា integer
- title គឺជា String
- body គឺជា String

ឧទាហរណ៍នៃការទាញ API (4)

6. បង្កើត class សំរាប់តំណាងអោយទិន្នន័យនៃ Map នោះ៖

```dart
class User {
  int userId;
  int id;
  String title;
  String body;

  User({this.userId, this.id, this.title, this.body});

  factory User.fromMap(Map<String, dynamic> json){
    return User(
      userId: json["userId"],
      id: json["id"],
      title: json["title"],
      body: json["body"],
    );
  }

  Map<String, dynamic> toMap(){
    return {
      "userId": userId,
      "id": id,
      "title": title,
      "body": body,
    };
  }
}
```

បញ្ជាក់៖ factory គឺជា keyword មួយ ដែលជំនួយដល់កាត់បន្ថយការបង្កើត object ដែលស្មុន

ឧទាហរណ៍នៃការទាញ API (5) - compute function

7. ប្រើ http សំរាប់ទាញយក API ព្រម convert ទិន្នន័យពីទំរង់ JSON មក User៖

```
List<User> parseUser(String jsonString) {
  List list = json.decode(jsonString);
  List<User> userList = list.map((x) => User.fromMap(x)).toList();
  return userList;
}

Future<List<User>> fetchUserList(http.Client client) async {
  http.Response response =
      await http.get("https://jsonplaceholder.typicode.com/posts?userId=5");
  return compute(parseUser, response.body);
}
```

ក្នុងនេះ យើងប្រើ function compute() ដែលមានតួរសំខាន់ក្នុងការធ្វើអោយការទាញ ទិន្នន័យដ៏ធំពី internet មិនមានការគាំង។ ព្រោះ compute វាបានបំបែកជា thread ថ្មីមួយទៀត កុំអោយប៉ះពាល់ដល់ main thread។ បើយើងមិនប្រើ compute ទេ នោះទិន្នន័យវាប្រើពេលយូរ ហើយវានាំអោយគាំង App។

ឧទាហរណ៍នៃការទាញ API (6)

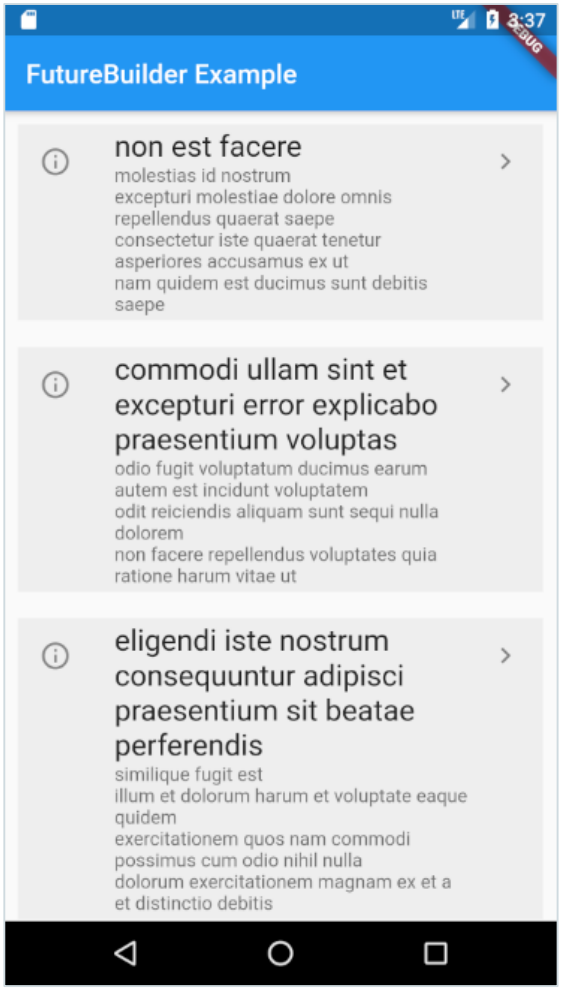
8. កូដក្នុង body ដែលទាញទិន្នន័យដោយប្រើ FutureBuilder៖

```
return FutureBuilder<List<User>>(
  future: fetchUserList(http.Client()),
  builder: (context, snapshot) {
    if (snapshot.connectionState == ConnectionState.done) {
      if (snapshot.hasData) {
        return _buildListView(snapshot.data);
      } else {
        return Center(
          child: Text("No Data"),
        );
      }
    } else {
      return Center(
        child: CircularProgressIndicator(),
      );
    }
  },
);
```

ឧទាហរណ៍នៃការទាញ API (7)

9. ចុងក្រោយបន្ទាប់ពីយើងបានទិន្នន័យ យើងបោះបន្តទៅអោយ _buildListView៖

```
_buildListView(List<User> users) {
  return ListView.builder(
      itemCount: users.length,
      itemBuilder: (context, index) {
        return Container(
          color: Colors.grey[200],
          margin: EdgeInsets.all(10.0),
          child: ListTile(
            leading: Icon(Icons.info_outline),
            title: Text(
              users[index].title,
               style: TextStyle(fontSize: 22.0),
            ),
            subtitle: Text(users[index].body),
            trailing: Icon(Icons.keyboard_arrow_right),
          ),
        );
      });
}
```



ការបំលែងពី JSON ទៅ Dart Class ដោយស្វ័យប្រវត្តិ

- Quicktype:

  https://app.quicktype.io/

- Javier Lecuona:

  https://javiercbk.github.io/json_to_dart/