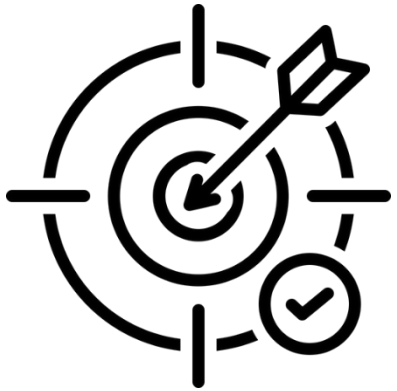
REST-Based Services

Web Programming and Testing

Mario Simaremare, S.Kom., M.Sc.
Program Studi Sarjana Sistem Informasi
Institut Teknologi Del

Objectives

- The objective of this session is the following:
  - The students are able to elaborate the concept behind the REST-based services.



Outlines

1. Flashback.
2. REST-based service.
3. REST services around us.

Flashback

Service and API Contract

- A service is a software program that makes its functionality available via a published **API** that is part of a service contract.

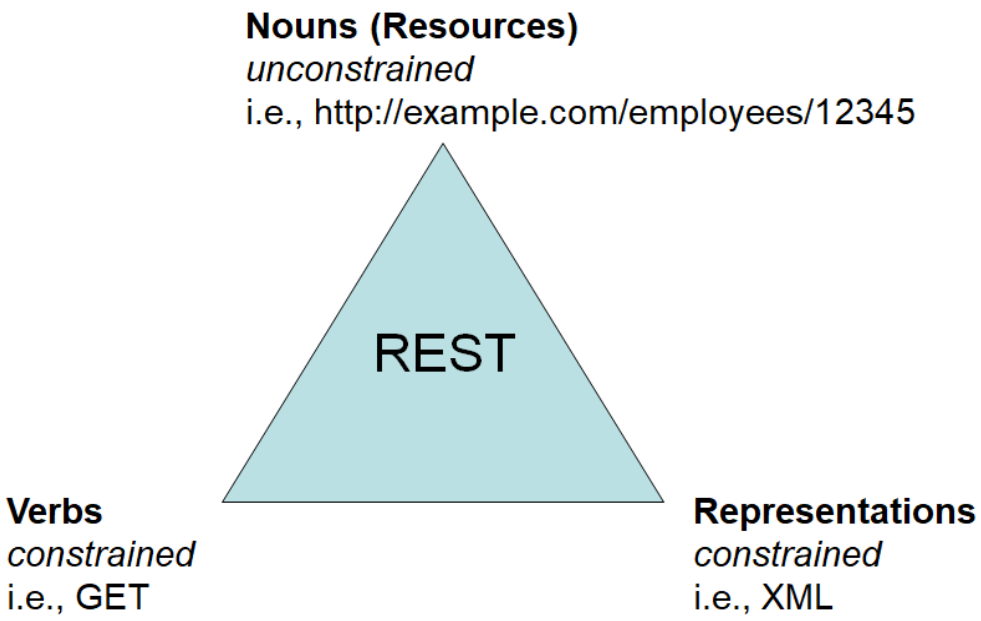
- The API specification is defined in a contract.
  It contains:
  - How to consume the API,
  - The parameters specifications, and
  - What kind of output is expected.

REST-based Services

REST

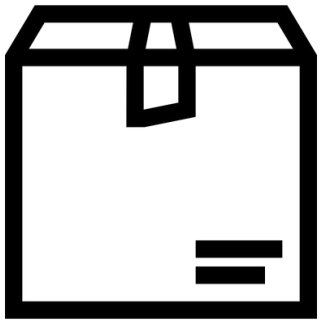
- REST: REpresentational State Transfer.
  - Is an architectural pattern.
  - It runs on top of the HTTP infrastructure.

- REST can be perceived as:
  - A mechanism to exchange data.
  - The state of the data being exchanged is represented in a standard format (e.g. JSON, XML, etc.)



REST: Semantics

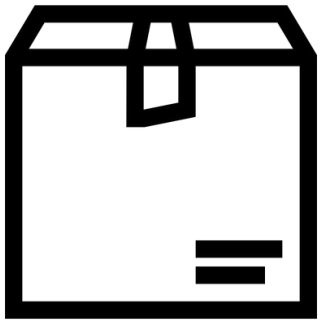
- Compared to the traditional approach (e.g. SOAP service), REST promotes semantics. The semantics is reached by:
  - utilizing HTTP verbs and HTTP response code to ‘self-explain’ the request and the response.



- HTTP verbs are methods available in the protocol.
  - `GET`, `POST`, `PUT`, `PATCH`, `DELETE`, and `OPTION`, etc.
- HTTP response code to represent the outcome.
  - 200 – OK, 403 – Unauthorized, etc.

REST: Stateless

- Interaction between the two communicating parties is stateless.
  - Because it runs on top of HTTP which is stateless.
  - No session at all.
    The client is the sole party to keep track of its own session.
  - Due to this, in the case of sensitive data request, the client must embed its authentication token along with the request.



- In what case statelessness is a good thing?
  - Servers do not need to keep a tab for its user.
  - It promotes scalability since the client is responsible for its own session.

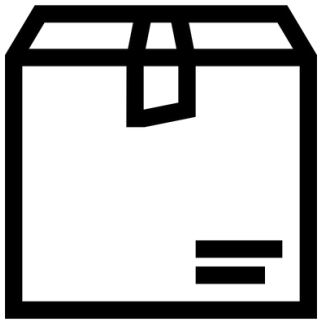
Where Is The Contract?

- Contract is essential and a service must obey the contract.
- In the traditional services, SOAP-based services, the contract is written in a well-formatted WSDL document.
- What about REST?
  - Unlike the traditional services, the specification of a REST service is defined in the API documentation.
  - Not in a specification from which the service is derived.
  - It simplify things, but may raise an inconsistency.

REST Services Around Us

REST Services Around Us?

- In most cases, mobile apps do not store its entire content locally. It is very likely that most of the data is stored in the server and only the most recent lives locally in cache.
  - The communication between the app and the server is very likely implemented in REST.



- Today's social media websites, like Facebook, LinkedIn, etc. retrieves data from the servers seamlessly though REST services.

GitHub REST API

You can use the GitHub REST API to create calls to get the data you need to integrate with GitHub.

REST API overview

Learn about resources, libraries, previews and troubleshooting for GitHub's REST API.

Reference

View reference documentation to learn about the resources available in the GitHub REST API.

Guides

Learn about getting started with the REST API, authentication, and how to use the REST API for a variety of tasks.

APIs provided by GitHub

Get a user

Provides publicly available information about someone with a GitHub account.

GitHub Apps with the `Plan` user permission can use this endpoint to retrieve information about a user's GitHub plan. The GitHub App must be authenticated as a user. See "Identifying and authorizing users for GitHub Apps" for details about authentication. For an example response, see 'Response with GitHub plan information' below"

The `email` key in the following response is the publicly visible email address from your GitHub profile page. When setting up your profile, you can select a primary email address to be "public" which provides an email entry for this endpoint. If you do not set a public email address for `email`, then it will have a value of `null`. You only see publicly visible email addresses when authenticated with GitHub. For more information, see Authentication.

The Emails API enables you to list all of your email addresses, and toggle a primary email to be visible publicly. For more information, see "Emails API".

verb → `GET` `/users/{username}` ← URL format

Parameters

Name	Type	In	Description
accept	string	header	Setting to `application/vnd.github.v3+json` is recommended.
username	string	path	

Code samples

```
curl \
  -H "Accept: application/vnd.github.v3+json" \
  https://api.github.com/users/USERNAME
```

JavaScript (@octokit/core.js)

In this article

API to retrieve a user information

https://docs.github.com/en/free-pro-team@latest/rest/reference/users

Command Prompt

```
C:\Users\MSS>curl -H "Accept: application/vnd.github.v3+json" https://api.github.com/users/sigurita
{
  "login": "sigurita",
  "id": 35382893,
  "node_id": "MDQ6VXNlcjM1MzgyODkz",
  "avatar_url": "https://avatars3.githubusercontent.com/u/35382893?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/sigurita",
  "html_url": "https://github.com/sigurita",
  "followers_url": "https://api.github.com/users/sigurita/followers",
  "following_url": "https://api.github.com/users/sigurita/following{/other_user}",
  "gists_url": "https://api.github.com/users/sigurita/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/sigurita/starred{/owner}{/repo}",
  "subscriptions_url": "https://api.github.com/users/sigurita/subscriptions",
  "organizations_url": "https://api.github.com/users/sigurita/orgs",
  "repos_url": "https://api.github.com/users/sigurita/repos",
  "events_url": "https://api.github.com/users/sigurita/events{/privacy}",
  "received_events_url": "https://api.github.com/users/sigurita/received_events",
  "type": "User",
  "site_admin": false,
  "name": "sigurita",
  "company": null,
  "blog": "https://sigurita.com",
  "location": null,
  "email": null,
  "hireable": null,
  "bio": null,
  "twitter_username": null,
  "public_repos": 2,
  "public_gists": 0,
  "followers": 0,
  "following": 0,
  "created_at": "2018-01-12T20:52:23Z",
  "updated_at": "2020-11-11T05:12:39Z"
}

C:\Users\MSS>
```

Consuming the API via cURL

https://api.github.com/users/sigu

api.github.com/users/sigurita

Raw Parsed


```
{
    "login": "sigurita",
    "id": 35382893,
    "node_id": "MDQ6VXNlcjM1MzgyODkz",
    "avatar_url": "https://avatars3.githubusercontent.com/u/35382893?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/sigurita",
    "html_url": "https://github.com/sigurita",
    "followers_url": "https://api.github.com/users/sigurita/followers",
    "following_url": "https://api.github.com/users/sigurita/following{/other_user}",
    "gists_url": "https://api.github.com/users/sigurita/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/sigurita/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/sigurita/subscriptions",
    "organizations_url": "https://api.github.com/users/sigurita/orgs",
    "repos_url": "https://api.github.com/users/sigurita/repos",
    "events_url": "https://api.github.com/users/sigurita/events{/privacy}",
    "received_events_url": "https://api.github.com/users/sigurita/received_events",
    "type": "User",
    "site_admin": false,
    "name": "sigurita",
    "company": null,
    "blog": "https://sigurita.com",
    "location": null,
    "email": null,
    "hireable": null,
    "bio": null,
    "twitter_username": null,
    "public_repos": 2,
    "public_gists": 0,
    "followers": 0,
    "following": 0,
    "created_at": "2018-01-12T20:52:23Z",
    "updated_at": "2020-11-11T05:12:39Z"
}
```


Since the API requires a GET, we could use browser to consume it

SoapUI
Supported by SMARTBEAR

Tools Learn Docs Resources

Try ReadyAPI Free

Download the Most Advanced API Testing Tool on the Market

With an improved interface and feature set, you can immediately switch to ReadyAPI and pick up right where you left off in SoapUI. It's as seamless as it can get.

Use API testing tool, e.g. SoapUI or Postman

ReadyAPI

Get the most advanced functional testing tool for REST, SOAP and GraphQL APIs.

Download ReadyAPI

Learn More

- ✓ SOAP API Testing
- ✓ REST API Testing
- ✓ WSDL Coverage
- ✓ Scripted Assertions
- ✓ Largest Online API Testing Community
- ✓ GraphQL API Testing

SoapUI Open Source

Get the open source version of the most widely used API testing tool in the world.

Download SoapUI Open Source

Learn More

- ✓ SOAP API Testing
- ✓ REST API Testing
- ✓ WSDL Coverage
- ✓ Scripted Assertions
- ✓ Largest Online API Testing Community
- ✗ GraphQL API Testing

POSTMAN Product Use Cases Pricing Enterprise Explore Learning Center Sign In Sign Up

Download Postman

Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman.

Use API testing tool, e.g. SoapUI or Postman

The Postman app

The ever-improving Postman app (a new release every two weeks) gives you a full-featured Postman experience.

Download the App

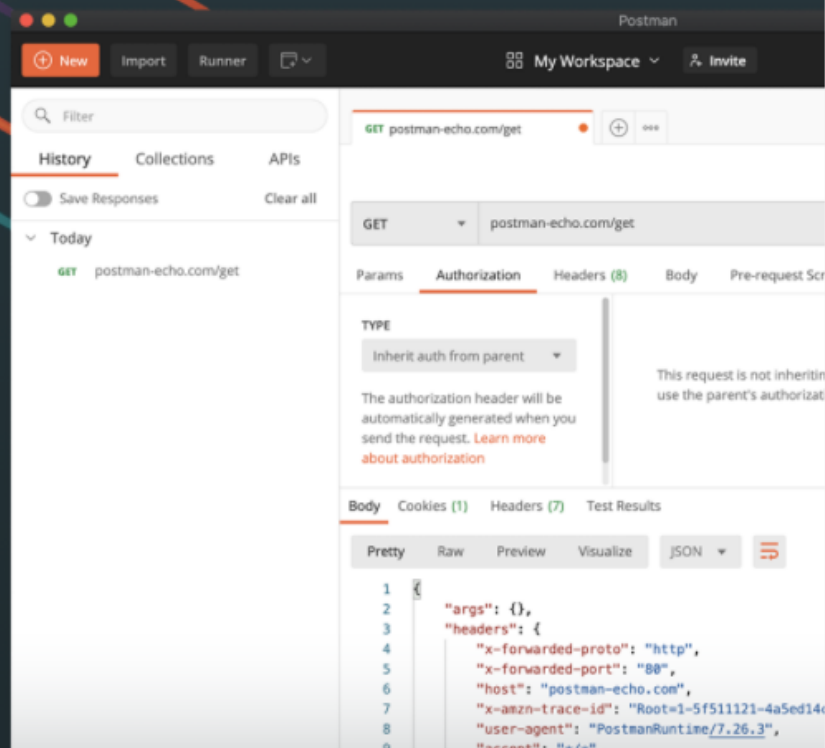
By downloading and using Postman, I agree to the Privacy Policy and Terms.

Version 7.34.0 | Release Notes | Product Roadmap

Not your OS? *Download for Mac (macOS) or Linux (x64)*

Postman on the web

You can now access Postman through your web browser. Simply create a free Postman account, and you're in.



Endpoint Explorer

Choose a method and enter a REST endpoint URL

Method: GET

Endpoint: https://api.github.com/users/sigurita

Send | Save Request

Request | Authentication & Headers

Header	Value
Accept	application/vnd.github.v3+json

+ Add header

Response (Raw)

```
HTTP/1.1 200 OK
date=Thu, 12 Nov 2020 02:40:50 GMT
content-type=application/json; charset=utf-8
server=GitHub.com
status=200 OK
cache-control=public, max-age=60, s-maxage=60
vary=Accept, Accept-Encoding, Accept, X-Requested-W
etag=W/"8712bbf30b853c4cc6a60e1a0930a0500a0ad3c684e
last-modified=Wed, 11 Nov 2020 05:12:39 GMT
x-github-media-type=github.v3; format=json
access-control-expose-headers=ETag, Link, Location,
access-control-allow-origin=*
```

Don't show this window on launch

Close

Authorized Access

- There is a situation where the API requires the consumer to be authorized first.

- Authorization is manifested in the form of tokens.
  - The token should be attached to the request.

Do NOT share the token!

Settings > Developer Settings > Personal access tokens

☑ user	Update all user data
☑ read:user	Read all user profile data
☑ user:email	Access user email addresses (read-only)
	Follow and unfollow users
	Delete repositories

✓ 3c468c385118511063329938 74ce0d97c7b[redacted]

Update the authenticated user

Note: If your email is set to private and you send an `email` parameter as part of this request to update your profile, your privacy settings are still enforced: the email address will not be displayed on your public profile or via the API.

PATCH /user

Parameters

Name	Type	In	Description
`accept`	string	header	Setting to `application/vnd.github.v3+json` is recommended.
`name`	string	body	The new name of the user.
`email`	string	body	The publicly visible email address of the user.
`blog`	string	body	The new blog URL of the user.
`twitter_username`	string or null	body	The new Twitter username of the user.
`company`	string	body	The new company of the user.
`location`	string	body	The new location of the user.
`hireable`	boolean	body	The new hiring availability of the user.
`bio`	string	body	The new short biography of the user.

Code samples

Shell

```
curl \
  -X PATCH \
  -H "Accept: application/vnd.github.v3+json" \
```

In this article

Updating user information

Updating user information

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum

Navigator

Projects
- Project 1
  - https://api.github.com
    - Sigurita [/users/sigurita]
      - Sigurita 1
        - user-detail
  - https://api.github.com
    - User [/user]
      - User 1
        - Request 1

Request 1

Method: PATCH
Endpoint: https://api.github.com
Resource: /user
Parameters:

Request | Raw

Name	Value	Style	Level

Required: Sets if parameter is required
Type:
Options: Add.. Edit..

Media Type: application/ison
Post QueryString

Authorization: sigurita

Access Token: 3c468c385118511061332993874ce0d97c[redacted]

Enter existing access token, or use "Get Token" below.

Get Token

Advanced...

Learn about OAuth 2

Auth (sigurita) Headers (1) Attachments (... Representations (1) JMS Headers JMS Property (0)

response time: 1132ms (1402 bytes)

Request Properties | Request Params

Property	Value
Name	Request 1
Description	
Encoding	
Endpoint	https://api.github....
Timeout	
Bind Address	
Follow Redirects	true
Username	
Password	
Domain	

Properties

XML JSON HTML Raw


```
1  {
2    "login": "sigurita",
3    "id": 35382893,
4    "node_id": "MDQ6VXNlcjMlMzgyODkz",
5    "avatar_url": "https://avatars3.githubusercontent.com/u/35382893?v=4",
6    "gravatar_id": "",
7    "url": "https://api.github.com/users/sigurita",
8    "html_url": "https://github.com/sigurita",
9    "followers_url": "https://api.github.com/users/sigurita/followers",
10   "following_url": "https://api.github.com/users/sigurita/following{/other_user}",
11   "gis
12   "sta
13   "sub
14   "org
15   "repos_url": "https://api.github.com/users/sigurita/repos",
16   "events_url": "https://api.github.com/users/sigurita/events{/privacy}",
17   "received_events_url": "https://api.github.com/users/sigurita/received_events",
18   "type": "User",
19   "site_admin": false,
20   "name": "sigurita",
21   "company": null,
22   "blog": "https://sigurita.com",
23   "location": null,
24   "email": null,
25   "hireable": null,
26   "bio": "Aloha",
27   "twitter_username": null,
28   "public_repos": 2,
29   "public_gists": 0,
30   "followers": 0,
31   "following": 0,
32   "created_at": "2018-01-12T20:52:23Z",
33   "updated_at": "2020-11-12T03:47:23Z",
34   "private_gists": 0,
35   "total_private_repos": 0,
36   "owned_private_repos": 0,
37   "disk_usage": 6,
38   "collaborators": 0,
39   "two_factor_authentication": false,
40   "plan":    {
41     "name": "free",
42     "space": 976562499,
43     "collaborators": 0,
44     "private_repos": 10000
45   }
46 }
```


Headers (28) Attachments (... SSL Info (2 certs) Representations (2) Schema (conflic... JMS (0)

SoapUI log http log jetty log error log wsrm l... memory l...

sigurita

sigurita

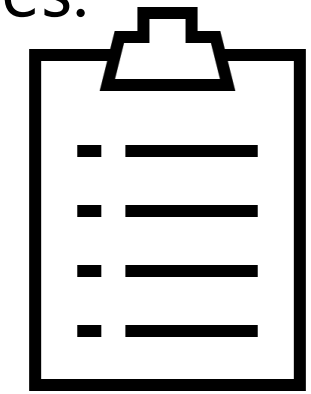
Aloha

Edit profile

https://sigurita.com

To-dos

1. Next time we will discuss more on the semantic aspects, both the HTTP request verbs and HTTP response codes.
2. Explore the GitHub APIs.



References

Srinivasan, M. (2012). Web Technology: Theory and Practice. Pearson.

Erl T. (2016). Service-Oriented Architecture: Analysis and Design for Services and Microservices. Pearson

GitHub REST API Documentation. https://docs.github.com/

Thank you