



جامعة عجمان
AJMAN UNIVERSITY

APIs & Web Technologies

Mohamed Al Solh

alsolh@alsolh.com



REST API

Agile - Scrum & indeed trends

- [youtube.com/watch?v=502ILHjX9EE](https://www.youtube.com/watch?v=502ILHjX9EE)
- <https://www.hiringlab.org/2019/11/19/todays-top-tech-skills/#>

Introduction

- Application talk to each other, there needs to be a standard way of integration
- Applications have evolved in many ways on how they talk to each other
 - Socket programming
 - Early days there was corba
 - SOAP and many other technologies like Java Message Queues, EJBs
 - REST
 - Web Sockets, MQTT

REST

- Data and functionality are considered resources (URI)
- Resources are manipulated with a set of operations (VERBS)
- Resources can be represented in multiple formats (JSON/XML/etc...)
- Communication between client and endpoint is stateless
 - Server will not store any state from the client
- Optional – Code on demand (JSON)
- <https://restfulapi.net/rest-architectural-constraints/#code-on-demand>

JSON vs XML

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language",
            "GlossSeeAlso": [
              "GML",
              "XML"
            ]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<glossary>
  <title>example glossary</title>
  <GlossDiv>
    <title>S</title>
    <GlossList>
      <GlossEntry ID="SGML" SortAs="SGML">
        <GlossTerm>Standard Generalized Markup Language</GlossTerm>
        <Acronym>SGML</Acronym>
        <Abbrev>ISO 8879:1986</Abbrev>
        <GlossDef>
          <para>A meta-markup language, used to create markup
languages such as DocBook.</para>
          <GlossSeeAlso OtherTerm="GML">
            <GlossSeeAlso OtherTerm="XML">
          </GlossDef>
          <GlossSee OtherTerm="markup">
        </GlossEntry>
      </GlossList>
    </GlossDiv>
  </glossary>
```

BSON

```
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D 0A 43 61 63 68 HTTP/1.1 200 OK..Cach
65 2D 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A e-Control: no-cache..
50 72 61 67 6D 61 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A 43 6F 6E Pragma: no-cache..Con
74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F tent-Type: applicatio
6E 2F 62 73 6F 6E 3B 20 63 68 61 72 73 65 74 3D 75 74 66 2D 38 n/bson; charset=utf-8
0D 0A 45 78 70 69 72 65 73 3A 20 2D 31 0D 0A 53 65 72 76 65 72 ..Expires: -1..Server
3A 20 4D 69 63 72 6F 73 6F 66 74 2D 49 49 53 2F 38 2E 30 0D 0A : Microsoft-IIS/8.0..
58 2D 41 73 70 4E 65 74 2D 56 65 72 73 69 6F 6E 3A 20 34 2E 30 X-AspNet-Version: 4.0
2E 33 30 33 31 39 0D 0A 58 2D 53 6F 75 72 63 65 46 69 6C 65 73 .30319..X-SourceFiles
3A 20 3D 3F 55 54 46 2D 38 3F 42 3F 51 7A 70 63 56 58 4E 6C 63 : =?UTF-8?B?QzpcVXNlc
6E 4E 63 62 58 64 68 63 33 4E 76 62 6C 78 45 62 32 4E 31 62 57 nNcbXdhc3Nvb1xEb2N1bW
56 75 64 48 4E 63 56 6D 6C 7A 64 57 46 73 49 46 4E 30 64 57 52 VudHNcVmlzdWFsIFN0dWR
70 62 79 41 79 4D 44 45 7A 58 46 42 79 62 32 70 6C 59 33 52 7A pbyAyMDEzXFBYb2plY3Rz
58 45 4A 7A 62 32 35 46 65 47 46 74 63 47 78 6C 58 45 4A 7A 62 XEJzb25FeGFtcGxlXEJzb
32 35 46 65 47 46 74 63 47 78 6C 58 47 46 77 61 56 78 69 62 32 25FeGFtcGxlXGFwaVxib2
39 72 63 31 77 78 3F 3D 0D 0A 58 2D 50 6F 77 65 72 65 64 2D 42 9rclwx?=.X-Powered-B
79 3A 20 41 53 50 2E 4E 45 54 0D 0A 44 61 74 65 3A 20 46 72 69 y: ASP.NET..Date: Fri
2C 20 31 37 20 4A 61 6E 20 32 30 31 34 20 30 31 3A 30 35 3A 34 , 17 Jan 2014 01:05:4
30 20 47 4D 54 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 0 GMT..Content-Length
3A 20 31 31 31 0D 0A 0D 0A 6F 00 00 00 10 49 64 00 01 00 00 00 : 111.....o....Id.....
02 54 69 74 6C 65 00 13 00 00 00 47 72 65 61 74 20 45 78 70 65 .Title.....Great Expe
63 74 61 74 69 6F 6E 73 00 02 41 75 74 68 6F 72 00 10 00 00 00 ctations..Author.....
43 68 61 72 6C 65 73 20 44 69 63 6B 65 6E 73 00 01 50 72 69 63 Charles Dickens..Pric
65 00 66 66 66 66 66 66 E6 23 40 09 50 75 62 6C 69 63 61 74 69 6F e.ffffff#@.Publicatio
6E 44 61 74 65 00 00 CC A9 AE 43 01 00 00 00 nDate..i@C....
```

Working with Teams on Integration Testing

- Usually new APIs are being developed or changes are introduced to them
- These changes are not available for other teams to develop directly, each team need to provide a documentation on how to call the APIs
- That is why WSDLs are there, and why Swagger is introduced

SOAP UI

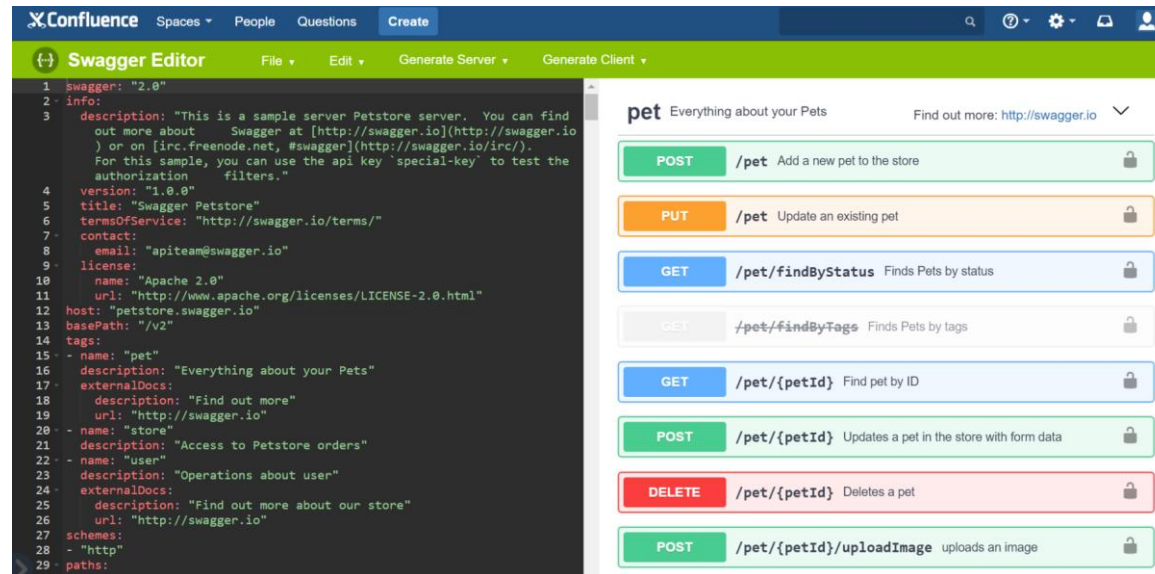
- Soap ui supports both restful and soap apis
- Soap can generate clients for you
- <http://www.dneonline.com/calculator.asmx?WSDL>

Postman & curl

- You can generate curl from postman
- Postman supports restful web apis
- Postman has some simple test functionality
- <http://dummy.restapiexample.com/>

Swagger

- Used to overcome the limitation of documenting in JSON as alternative to WSDL
- Can generate Mock API
- You can test directly using their portal



<https://swagger.io/tools/swaggerhub/>

HTTP Verbs

- HTTP GET
- HTTP POST
- HTTP PUT
- HTTP DELETE
- HTTP PATCH

HTTP Verbs

HTTP METHOD	CRUD	ENTIRE COLLECTION (E.G. /USERS)	SPECIFIC ITEM (E.G. /USERS/123)
POST	Create	201 (Created), 'Location' header with link to /users/{id} containing new ID.	Avoid using POST on single resource
GET	Read	200 (OK), list of users. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single user. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	405 (Method not allowed), unless you want to update every resource in the entire collection of resource.	200 (OK) or 204 (No Content). Use 404 (Not Found), if ID not found or invalid.
PATCH	Partial Update/Modify	405 (Method not allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). Use 404 (Not Found), if ID not found or invalid.
DELETE	Delete	405 (Method not allowed), unless you want to delete the whole collection — use with caution.	200 (OK). 404 (Not Found), if ID not found or invalid.

HTTP Status Codes

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
- <https://blog.alexa.com/http-status-codes-cheat-sheet/>
- 1xx Status Codes: Information Request
- 2xx Status Codes: Success
- 3xx Status Codes: Redirection
- 4xx Status Codes: Client Error
- 5xx Status Codes: Server Error

- Always return body with the status

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Websockets

- <https://www.websocket.org/echo.html>

Web containers

- https://en.wikipedia.org/wiki/Web_container#:~:text=A%20web%20container%20is%20responsible,that%20include%20server%2Dside%20code.
- you need a server that processes http requests and binds them to your code according to the location being accessed, all this requires lots of code but the web application server handles it for you.
 - Jsp
 - Jsf
 - Thymeleaf
<https://en.wikipedia.org/wiki/Thymeleaf#:~:text=In%20web%20applications%20Thymeleaf%20aims,under%20the%20Apache%20License%202.0.>
 - <https://www.quora.com/Is-JSP-a-deprecated-technology-now-What-are-the-other-options-for-viewing-technology-in-Spring-Is-it-good-to-use-Thymeleaf-or-JSF>
 - All these are server side technologies
- Alternatively you can use nginx as a web server and host your frontend there
 - http://nginx.org/en/docs/beginners_guide.html

Spring Boot

- <https://www.pluralsight.com/courses/spring-boot-fundamentals>

Spring Main Topics

- <https://www.baeldung.com/spring-boot-annotations>
- <https://www.baeldung.com/spring-boot-starters>
- <https://www.baeldung.com/spring-boot-actuators>
- <https://www.baeldung.com/spring-boot-application-configuration>
- Other topics
 - <https://www.baeldung.com/spring-boot>

Complete Example

- Spring Boot, Mysql, React docker
- <https://www.callicoder.com/spring-boot-mysql-react-docker-compose-example/>
- <https://github.com/callicoder/spring-security-react-ant-design-polls-app>

Complete Example

```
Windows PowerShell
PS C:\Users\alsoh\code\spring-boot\spring-security-react-ant-design-polls-app> docker-compose up
Creating network "spring-security-react-ant-design-polls-app_backend" with the default driver
Creating network "spring-security-react-ant-design-polls-app_frontend" with the default driver
Creating volume "spring-security-react-ant-design-polls-app_db-data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
8559a31e96f4: Already exists
d51ce1c2e575: Already exists
c2344adc4858: Already exists
fcf3ceff18fc: Already exists
16da0c38dc5b: Already exists
b905d1797e97: Already exists
4b50d1c6b05c: Already exists
d85174a87144: Pull complete
a4ad33703fa8: Pull complete
f7a5433ce20d: Pull complete
3dcd2a278b4a: Pull complete
Digest: sha256:32f9d9a069f7a735e28fd44ea944d53c61f990ba71460c5c183e610854ca4854
Status: Downloaded newer image for mysql:5.7
Building app-server
```

Extra Reading - Certificates

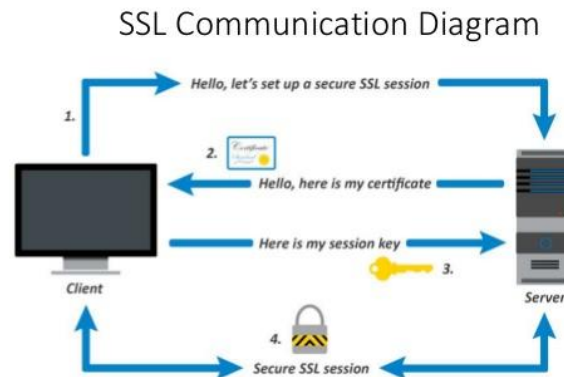
Theory: https://en.wikipedia.org/wiki/Self-signed_certificate / <https://en.wikipedia.org/wiki/HTTPS> / Public and Private Key - <https://www.cloudflare.com/learning/ssl/how-does-public-key-encryption-work/>

Practical: <https://www.baeldung.com/spring-boot-https-self-signed-certificate>

Free Verified Certificates - <https://letsencrypt.org/>

Extra Reading - Certificates

- The Client wants to establish a connection - ClientHello
- The server answers with a ServerHello. This includes a randomly generated number and a server certificate
- The client creates a "pre master secret" and sends it encrypted to the server using its public key.
- Both parties create a session key from random number and pre master secret.



Spring Security & OATH

- <https://spring.io/guides/tutorials/spring-boot-oauth2/>
- <https://www.baeldung.com/spring-security-login>

JHipster

- <https://www.jhipster.tech/installation/>
- <https://www.jhipster.tech/tech-stack/>