REST API Security
Introduction to REST APIs

Guadalupe Ortiz Bellot

Computer Science and Engineering Department

Universidad de Cádiz

Contents

1. Web Service Introduction
2. Rest APIs

Contents

1. **Web Service Introduction**
    - **Terms**
    - **REST Overview**
    - **REST versus SOAP**
2. Rest APIs

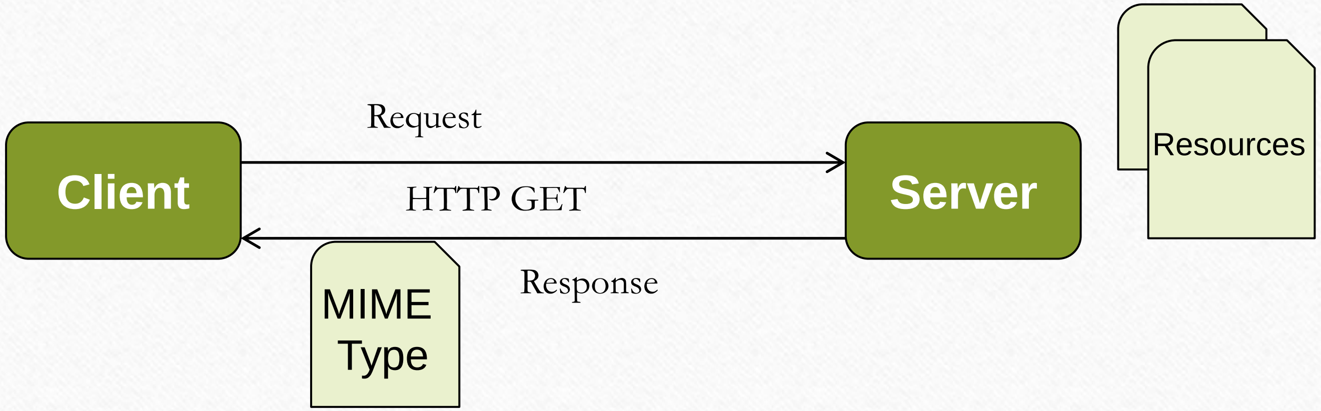
1. Web Service Introduction
Search for the following terms in the Web

- Web Services
- URI
- Endpoint (for web services)
- Media type
- JSON

1. Web Service Introduction
REST Overview

REST: Representational State Transfer.

1. A user makes a request (for instance GET) to an application html address, for instance through the web browser.
2. The browser sends a request to the HTTP server.
3. The server responses with an HTML document with a MIME type.



1. Web Service Introduction
REST versus SOAP (i)

- The services built with REST architectural style (named RESTful services) **encapsulate data in a simple XML format** and transport them through HTTP as a request of a web site to a web server.

- RESTful web services are particularly useful when it is only necessary to **submit and receive simple messages**.

- SOAP is mainly used for Enterprise applications to integrate **more complex data types and applications**, as well as **legacy systems**.

1. Web Service Introduction
REST versus SOAP (ii)

- REST:
  - Light
  - Legible by human beings
  - Easy to build

YAHOO! Google

amazon twitter

- SOAP
  - Easy to consume - sometimes
  - Strongly typed – data type checking
  - Development tools
  - More security

amazon

Google

Contents

1. Web Service Introduction: Rest versus SOAP
2. **Rest APIs**
    1. **Rest Basic Concepts**
    2. **Data Transfer Formats**

2. REST APIs
REST Basic Concepts. Principles (i)

REST is an architectonical style for services that is based on web standards. Its main principles are:

- Everything can be identified as a resource and every resource can be identified by an URI.
- A resource can be represented in multiple formats, defined by a *media type*.
- HTTP standard methods are used to interact with the resources: mainly GET, POST, PUT and DELETE.
- The communications between the client and the endpoint are *without state*.

2. REST APIs
REST Basic Concepts. Principles (ii)

- The World Wide Web is a classic example of REST architectural style: URIs identify the resources and HTTP the protocol used to access the URIs.

- HTTP provides a uniform interface and a set of methods to manipulate the resource.

- A client program, as a web browser, can access, update, add, and delete a web resource through the URI using several HTTP methods.

2. REST APIs
REST Basic Concepts. JAX-RS (i)

- Standard API is based on annotations to create a Java RESTful web service and a client for its invocation.

- JAX -RS specification follows the following objectives:
  - **POJO-centered**: The JAX-RS API provides a set of annotations and related classes/interfaces that can be used in the POJOs with the aim of expose them as RESTful resources.
  - **HTTP-centered**: Since REST resources are exposed through HTTP, the specification provides a clear mapping from the HTTP protocol and the corresponding classes and methods of the JAX-RS API.

2. REST APIs
REST Basic Concepts. JAX-RS (ii)

- Through the use of this API: a POJO can be marked through annotations that permit identifying:
  - A resource as a URI
  - A set of methods well defined to access the resources (GET, POST, et cetera)
  - Multiple representation formats of resources

```
@GET
@Path(“/hello")
@Produces(MediaType.TEXT_PLAIN)
   public String sayHello( ) {//...}
```

2. REST APIs
REST Basic Concepts. JAX-RS (iii)

- At runtime, the environment that implements JAX - RS specification is responsible of the Java application invocation through the **HTTP request mapping** with the **Java method** that satisfies the request.
- Java class and method that represent the resource have to be determined, as well as the content type and the invocated HTTP method.

http://applicationName/hello

GET
Plane Text
public String sayHello( )

2. REST APIs
REST Basic Concepts. JAX-RS (iv)

- **Format independence**: the API provides a mechanism that allows adding the HTTP content type in a standard way.

- **Container independence**: the application developed using JAX-RS must be able to be executed in any container.

- **Java Enterprise Edition Inclusion**: JAX-RS is a Java EE 6 component.

2. REST APIs
REST Basic Concepts. JAX-RS (v)

- It offers support for the use of the HTTP standard methods GET, POST, PUT, DELETE, HEAD y OPTIONS
  - **GET**: Retrieve a resource
  - **POST**: Create a resource
  - **PUT**: Update a resource
  - **DELETE**: Delete a resource
  - **HEAD**: Same function as GET, but it does not return the body. It is used to obtain meta-information about the resource. If there is no method marked as HEAD, it can be done through a GET and the body is discarded. [See examples in https://www.logicbig.com/tutorials/java-ee-tutorial/jax-rs/head-example.html]
  - **OPTIONS**: It provides the available communications options. If there is no method marked as @OPTIONS, an automatic response is generated. [See examples in https://www.logicbig.com/tutorials/java-ee-tutorial/jax-rs/options-example.html]

2. REST APIs
Data Transfer Formats. Basic Concepts

- The client checks and updates the resources in the URI through the exchange of resource representations.
- Such representations contain information in formats like HTML, XML or JavaScript Object Notation (JSON).
- The client must know the type returned by the service.
- In general, the client specifies the representation desired to receive (Accept), and the server returns the resources desired in such format.
- All the information needed to process a request of a resource is contained in the request, therefore the interaction is without state.

2. REST APIs
Data Transfer Formats. Specification

- In the service:
  - In a predetermined way, a REST resource is published or consumed with the MIME type * / *.
  - A REST resource can restrict the media type admitted by the request and the response with annotations @Consumes and @ Produces, respectively.
  - These annotations can be specified in the methods and the classes. If the annotation is specified in the method it cancels the class annotation.

- In the client:
  - Content-Type: it indicates the submitted type (for example,

    "text/plain", "text/xml","text/html", "application/json“)
  - Accept: it indicates the resource types expected to be received.

Support Bibliography and References

- Java EE 7 Essentials. By: Arun Gupta. Publisher: O'Reilly Media, Inc. Pub. Date: August 23, 2013. Print ISBN-13: 978-1-4493-7017-6
- RESTful Java Web Services. By: Jose Sandoval. Publisher: Packt Publishing Pub. Date: November 11, 2009. Print ISBN-13: 978-1-847196-46-0
- Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON. By: Masoud Kalali; Bhakti Mehta. Publisher: Packt Publishing Pub. Date: October 15, 2013. Print ISBN-13: 978-1-78217-812-5