Why soap failed?

Slow: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

WSDL dependent: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

Unit 2

The REST Architectural style

syllabus

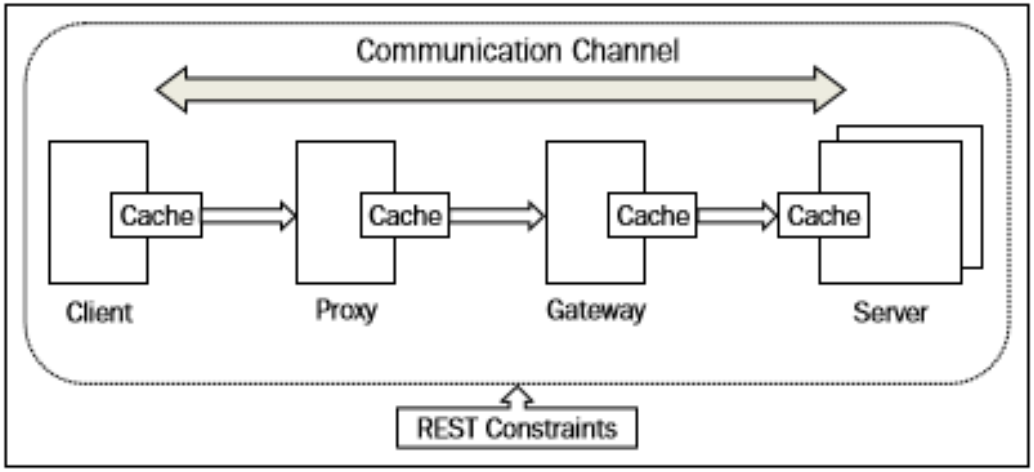
- Introducing HTTP
- The core architectural elements of a RESTful system
- Java tools and frameworks for building RESTful web services
- JSON message format and tools and frameworks around JSON
- Build RESTful web services with JAX-RS APIs
- The Description and Discovery of RESTful Web Services
- Design guidelines for building RESTful web services
- Secure RESTful web services

The REST architectural style

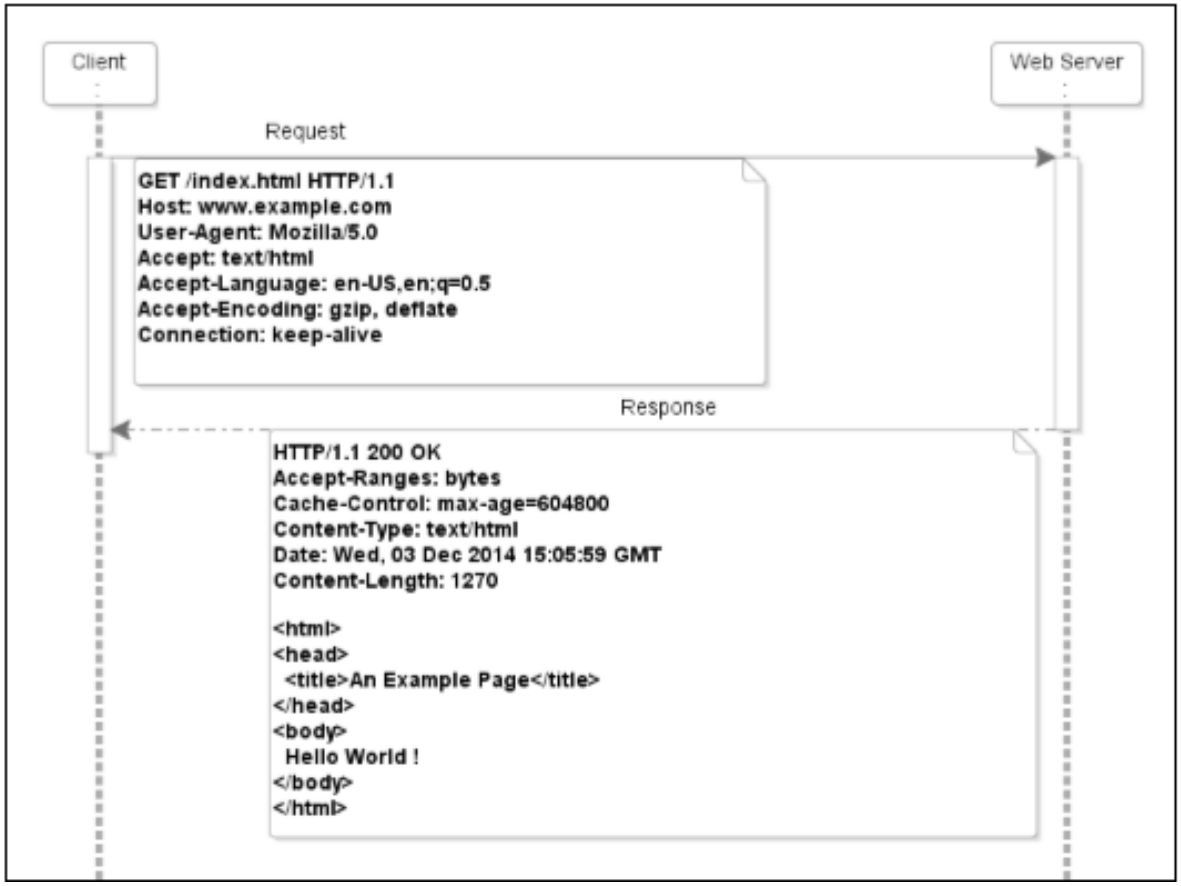
- REST is not an architecture; rather, it is a set of constraints that creates a software architectural style, which can be used for building distributed applications.

Constraints

- Client-server
- Stateless
- Cacheable
- Uniform interface
- Layered system
- Code on demand



HTTP request-response model



HTTP request-response model

```
GET /index.html HTTP/1.1 Host:
www.example.com User-Agent: Mozilla/5.0
Accept: text/htmlAccept-Language: en-
US,en;q=0.5 Accept-Encoding: gzip, deflate
Connection: keep-alive
```

```
HTTP/1.1 200 OK
ACCEPT-RANGES: BYTES
CACHE-CONTROL: MAX-AGE=604800
 CONTENT-TYPE: TEXT/HTML
DATE: WED, 03 DEC 2014 15:05:59 GMT
CONTENT-LENGTH: 1270
<HTML> <HEAD>  <TITLE>AN EXAMPLE PAGE</TITLE>
</HEAD> <BODY>  HELLO WORLD ! </BODY> </HTML>.
```

Understanding the HTTP request method

Method	Description
GET	This method is used for retrieving resources from the server by using the given URI.
HEAD	This method is the same as the GET request, but it only transfers the status line and the header section without the response body.
POST	This method is used for posting data to the server. The server stores the data (entity) as a new subordinate of the resource identified by the URI. If you execute POST multiple times on a resource, it may yield different results.
PUT	This method is used for updating the resource pointed at by the URI. If the URI does not point to an existing resource, the server can create the resource with that URI.
DELETE	This method deletes the resource pointed at by the URI.
TRACE	This method is used for echoing the contents of the received request. This is useful for the debugging purpose with which the client can see what changes (if any) have been made by the intermediate servers.
OPTIONS	This method returns the HTTP methods that the server supports for the specified URI.
CONNECT	This method is used for establishing a connection to the target server over HTTP.
PATCH	This method is used for applying partial modifications to a resource identified by the URI.

HTTP status codes

- 1xx Informational

100 –continue
101 Switching Protocols
102 Processing

- 2xx Success

200 OK
201 Created
204 No Content

- 3xx Redirection

304 Not modified

- 4xx Client Error

400 Bad Request
401 Unauthorized
403 Forbidden
404 Not Found
405 Method Not Allowe
408 Request Timeout
409 conflict
5xx Server Error501
500 Internal Server Erro

Representing content types using HTTP header fields

- `Content-Type: text/html`
- `Text`
- `Multipart`
- `Message`
- `Image`
- `Audio`
- `Video`
- `application`

RESTful web services

- RESTful web services use web protocol i.e. HTTP protocol method.

• have scalability, maintainability, help multiple application communication built on various programming languages etc. •

- RESTful web service implementation defines the method of accessing various resources which are required by the client and he has sent the request to the server through the web browser.

include:

- Resources
- Request Headers
- Request Body
- Response Body
- Status codes

The core architectural elements of a RESTful system

- Resources and their identifiers
- Representations of resources
- Generic interaction semantics for the REST resources
- Self-descriptive messages
- Hypermedia as the engine of an application state

the HATEOAS principle.

- http://www.packtpub.com/resources/departments/IT URI returns the following response to the client:
- {"departmentId":10,
- "departmentName":"IT",
- "manager":"John Chen,
- "links": [{
- "rel": "employees",
- "href": "http://packtpub.com/resources/departments/IT/employees"
- }]"}

- The application state now changes into the following form (as represented by the response content):
- [{"employeeId":100,
- "firstName":"Steven",
- "lastName":"King",
- "links": [{
- "rel": "self",
- "href": "http://www.packtpub.com/resources/employees/100"
- }]
- },
- {"employeeId":101

- "firstName":"Neena",
- "lastName":"Kochhar",
- "links": [{
- "rel": "self",
- "href": "http://www.packtpub.com/resources/employees/101"
- }]
- }]

Java tools and frameworks for building RESTful web services

- Jersey,
- Apache CXF
- RESTEasy
- Restlet

JSON message format and tools and frameworks around JSON

Processing json data