WEB SERVER

SHELL SCRIPTS | CGI | APACHE HTTPD | 80

ORACLE DB SERVER | 1521

CUSTOM SERVER APPLICATION | 3456

HOST OPERATING SYSTEM

HTTP RESPONSE (HTML CONTENT)

PORT 80

BROWSER CLIENT

WEB SERVER

JVM + SERVER APP RUNNING OUR WEB AP | APACHE HTTPD | 80

ORACLE DB SERVER | 1521

CUSTOM SERVER APPLICATION | 3456

HOST OPERATING SYSTEM

HTTP RESPONSE (HTML CONTENT)

PORT 80

BROWSER CLIENT

CLIENT

CLIENT
(HUMAN)

CLIENT

MAKE
MY
TOURS.COM

WEB APP
USING JAVA

SOAP
Web service client
using Java

INDIGO
(WEB APP
USING .NET)

SOAP
web service
using C#

SPICEJET
(WEB APP
USING PHP)

SOAP
web service
using PHP

PYTHON PROGRAM
SOAP
WEB SERVICE
CLIENT

Diagram 1 (top):

JAVA
APPLICATION
AS A
CLIENT TO
THE WEB
SERVICE

REF

JAVA VIRTUAL
MACHINE

APPLICATION
WRITTEN
IN .NET

C# OBJECT
WITH
FUNCTIONS

CLR IN .NET

Diagram 2 (bottom):

JAVA
APPLICATION
AS A
CLIENT TO
THE WEB
SERVICE

JAVA METHOD CALL

REF

JAVA OBJECT

JAVA VIRTUAL
MACHINE

SOAP
ENGINE
IN
JAVA

CREATED
USING
WSDL

SOAP REQUEST
IN XML
CONTAINS METHOD
AND PARAMS INFO

SOAP RESPONS
IN XML
CONTAINS
ACTUAL RESPONSE
DATA

SOAP
ENGINE
IN
.NET

exposes
WSDL

APPLICATION
WRITTEN
IN .NET

C# OBJECT
WITH
FUNCTIONS

CLR IN .NET

REST

Representational State Transfer

Transferring State (information) in different representations (XML/**JSON**/CSV/HTML/TEXT/...)
There is standards like WSDL (for SOAP) in REST
Client and server exchange data in the form of (mostly) JSON
Clients are typically JavaScript application (or mobile apps), which run directly on the end-user's browser (or mobile)

State (or data) is represented on the server --> resource
A Resource is identified using a URI (~URL) --> **http://example.com/api/products**
A Resource can be manipulated using HTTP request methods **(GET/POST/PUT/PATCH/DELETE)** <-- VERBS
Client and Server can negotiate for the kind (representation) of data (CONNEG)
CONNEG is done using HTTP request/response headers called '**Accept**' and '**Content-Type**'

The concept of RESTful services is based on 6 constraints laid out by Roy Fielding, creator of HTTP protocol:
1. Client/server application
2. Stateless
3. Uniform interface
4. Layered system
5. Cacheable
6. Code on demand (OPTIONAL)

EMBEDDED APACHE TOMCAT SERVER

HOST OS

handler
mapping

Controller

functions mapped to
url path/patterns

/hello

PORT 8080

/api/hello

DispatcherServlet

mapped to URL
/

model

CLIENT

JSON or XML

Jackson

view
resolver

Content
Negotiating
ViewResolver

JAXB
(part of jdk)

SPRING CONTAINER
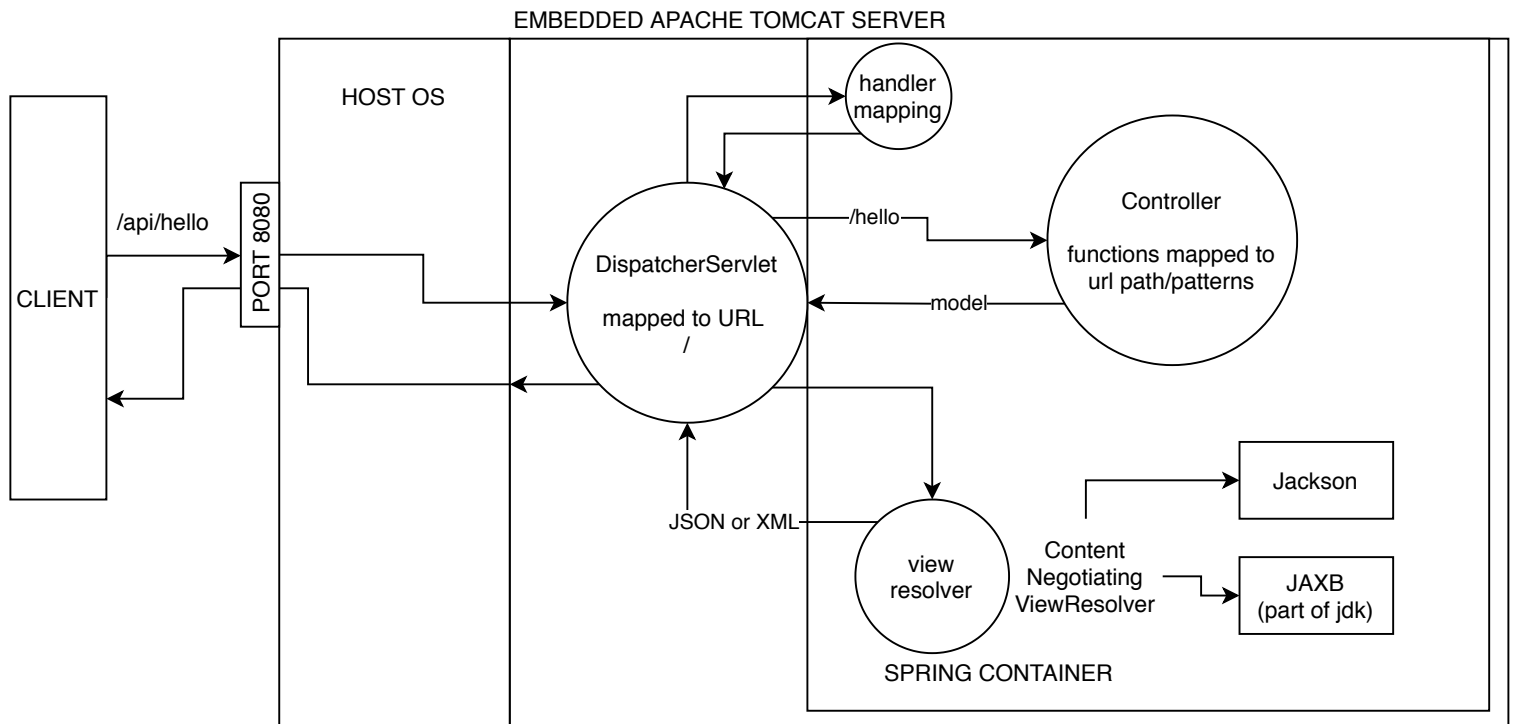
ContentNegotiatingViewResolver

is the default view resolver used by handler function's return value for all
@RestControllers. This resolver checks the header "Accept" and based on the value, it
will make use of either Jackson API (to generate JSON output) or JAXB API (to generate
XML output)