

Spring Professional Exam Tutorial v5.0

Question 01

Question 01 - What does REST stand for?

REST stands for **RE**presentational State Transfer.

It is an architectural style of designing distributed applications, in which requesting system accesses and manipulates textual representations of Web resources by using a uniform and predefined set of stateless operations, given a set of constraints.

Web resources are made available through URIs (Uniform Resource Identifiers) and are accessed or modified, usually through HTTP operations.

For example:

- ▶ `GET /customers`
- ▶ `GET /customers/1`
- ▶ `PUT /customers/6`
- ▶ `DELETE /customers/3`

Question 01 - What does REST stand for?

Most implementations of REST services are using HTTP as the application protocol, and JSON as format that moves data between caller and callee. However REST is not necessarily tied to HTTP or JSON.

REST defines constraints to be used for creating Web services, aligning with those constraints results in gaining non-functional properties of system, such as performance, scalability, simplicity, modifiability, visibility, portability, and reliability:

- ▶ Client-server architecture and Separation of Client-Server concerns
- ▶ Statelessness
- ▶ Cacheability
- ▶ Uniform interface
 - ▶ Resource identification in requests
 - ▶ Resource manipulation through representations
 - ▶ Self-descriptive messages
 - ▶ Hypermedia as the engine of application state
- ▶ Layered system
- ▶ Code on demand (optional)

Question 01 - What does REST stand for?

Requesting system interacts with the target system, usually by HTTP requests which should follow those semantics:

HTTP Method	+	URI
verb	+	noun (plural form)

GET	/customers
GET	/customers/1
PUT	/customers/6
PATCH	/customers/7
DELETE	/customers/3

REST services often allow you to perform all CRUD operations for specified resources:

- ▶ C - create
- ▶ R - read
- ▶ U - update
- ▶ D - delete

Question 01 - What does REST stand for?

Following HTTP Methods are usually used with REST services:

- ▶ GET - fetch resource or collection of resources at the specified URI
- ▶ PUT - create new resource, or update existing resource at the specified URI
- ▶ PATCH - partially updates existing resource at the specified URI, for example update only `firstName` of `Customer`
- ▶ DELETE - delete resource at the specified URI
- ▶ POST - triggers operation at server or creates a new resource at the specified URI

Other HTTP Methods that REST service can support are: HEAD, CONNECT, OPTIONS, TRACE.

Question 01 - What does REST stand for?

In case of request processing error, REST service should respond in a way for it to be easily to understand cause of issue. Error return to requester usually includes:

- ▶ HTTP Status Code
- ▶ Code ID
- ▶ Human readable error message that explains error

HTTP Status Code should fall within one of the groups:

- ▶ 1xx - Informational
- ▶ 2xx - Success
 - ▶ 200 OK
 - ▶ 201 Created
 - ▶ 204 No Content
- ▶ 3xx - Redirection
- ▶ 4xx - Client Error
 - ▶ 400 Bad Request
 - ▶ 401 Unauthorized
 - ▶ 403 Forbidden
 - ▶ 404 Not Found
- ▶ 5xx - Server Error
 - ▶ 500 Internal Server Error
 - ▶ 501 Not Implemented

Question 01 - What does REST stand for?

REST API should be backward compatible as long as possible, in case of introducing breaking change, one of versioning methods should be used:

- ▶ **URI versioning**
 - ▶ `/api/v1/products`
- ▶ **Query string versioning**
 - ▶ `/api/products?version=1`
- ▶ **Header versioning**
 - ▶ `/api/products`
 - ▶ **Header** - `Accepts-version: 1.0`
- ▶ **Media type content negotiation versioning**
 - ▶ `/api/products`
 - ▶ **Header** - `Accept: application/vnd.my-app.v1+json`

