

Quiz Questions

(1) Introduction to web services

← Classroom

REST API - GET POST
Introduction To Web Services

?

V

Problem Submissions Doubts

✓ Introduction To Web Services

Easy • Score 20/20

Problem statement

What is the primary purpose of a web service?

Send feedback

Options: Pick one correct answer from below

☐ To allow users to access websites

☒ To allow websites to communicate with other systems over the internet

☐ To allow users to search the internet

☐ To allow a website to store data

Solution description

Web services provide a way to retrieve and send data to different software systems over the internet with the help of standardized web protocols such as HTTP, using XML or JSON format.

(2) Introduction to Web Services

← Classroom

REST API - GET POST
Introduction To Web Services

?

V

Problem Submissions Doubts

✓ Introduction To Web Services

Easy • Score 20/20

Problem statement

What is an example of a web service?

Send feedback

Options: Pick one correct answer from below

☒ Payment Gateway API

☐ Microsoft Word

☐ Adobe Photoshop

☐ Amazon.com

Solution description

Web services are platform-independent, don't have a UI, and are meant to communicate with other applications. Websites have a UI and are accessed by using a web browser.

(3) Architecture of Web Services

← Classroom

REST API - GET POST
Architecture of Web Services

?

V

Problem Submissions Doubts

Architecture of Web Services
Easy • Score 20/20

Problem statement [Send feedback](#)

Given below is the diagram of web services architecture. Choose the correct order of execution of a client request to the server.

```
graph TD; Client((Client)) -- Request --> Server((Server)); Server -- Response --> Client; Server -- Publish --> Broker((Broker)); Broker -- Acknowledgement --> Client; Broker -- Server status --> Server;
```

1. The client requests the server for the information.
2. The server registers itself as ready to publish on the broker server.
3. The client asks the broker if the server is running or not.
4. The server sends the response back.
5. The broker acknowledges the request and sends the status back to the client.

Options: Pick one correct answer from below

☐ 3 → 2 → 5 → 1 → 4

☐ 1 → 2 → 3 → 4 → 5

☒ 2 → 3 → 5 → 1 → 4

☐ 2 → 3 → 1 → 4 → 5

Solution description

As soon as the server is up and running, it registers itself on the broker. The client then sends a request to the broker to check the server status. The broker then acknowledges the request and shares the server status. If the server is up and running, the client goes ahead with the request to the server and sends the response back to the client.

(4) Restful Web Services

← Classroom

REST API - GET POST
RESTful Web Services

?

V

Problem Submissions Doubts

RESTful Web Services
Easy • Score 20/20

Problem statement [Send feedback](#)

A developer is building a movie library that provides reviews for a given movie. He wishes to use the IMDB API for the same. The API requires him to send an authorization token with the request. In which component would he send the authorization token?

Options: Pick one correct answer from below

☐ Request Body

☐ Response Header

☐ Response Body

☒ Request Header

Solution description

The Request Header contains information about the request, so that server can respond accordingly. It contains details about the request which are not relevant to the business logic but provides additional information about the request such as Authorization credentials, Accepted type etc. For example, here is a list of some Request Headers and their usage.

Request Header	Example	Usage
Host	developer.mozilla.org	Host and port number
User-Agent	Mozilla/5.0 (Windows NT 6.1; Win64; x64)	Identify the Application of the requesting user.
Authorization	Basic YWxhZGRpbjpwGVuc2VzYW1l	Used to provide credentials that authenticate a user agent with a server.
Accept	text/html	Indicates which content types the client is able to understand
Connection	keep-alive	Controls whether the network connection stays open after the current transaction finishes.

(5) Restful Web Services

← Classroom

REST API - GET POST
RESTful Web Services

?

V

Problem Submissions Doubts

✓ RESTful Web Services

Easy • Score 20/20

Send feedback

Problem statement

You are building a RESTful API for a to-do list application. The API should allow users to create, read, update, and delete to-do list items.

Match the request with the correct type of the corresponding method.

A. A user wants to update the details of a specific to-do item. Which HTTP method should the API map this request to?

B. A user wants to delete a specific to-do item from the list. Which HTTP method should the API map this request to?

C. A user wants to add a new to-do item to the list. Which HTTP method should the API map this request to?

D. A user wants to retrieve the complete list of to-do items. Which HTTP method should the API map this request to?

1. GET
2. POST
3. PUT
4. DELETE

Options: Pick one correct answer from below

☒ (A-3)(B-4)(C-2)(D-1)

☐ (A-2)(B-4)(C-3)(D-1)

☐ (A-3)(B-4)(C-1)(D-2)

☐ (A-2)(B-3)(C-4)(D-1)

Solution description

We use the GET method to retrieve information from the server. POST method is used to add new information. The PUT method is used to update existing data and the DELETE method is used to delete something specific from the server.

(6) Restful Web Services with Spring

← Classroom

REST API - GET POST
RESTful Web Services with Spring Boot

?

V

Problem Submissions Doubts

✓ RESTful Web Services with Spring Boot

Easy • Score 20/20

Send feedback

Problem statement

Imagine you are building a Spring Boot web application that allows users to view all blog posts. You have created the following controller to handle this operation:

```
@RestController
@RequestMapping("/users")
public class BlogController {

    @GetMapping("/blogs")
    public String getAllBlogs(){
        // Returns all Blog posts
    }
}
```

Now, you want to test if the method is working as intended or not. Which of the following URLs would be used to retrieve a list of all posts?

Options: Pick one correct answer from below

☐ '/blogs'

☐ '/users/getAllBlogs'

☐ '/blogs/users'

☒ '/users/blogs'

Solution description

The @RequestMapping annotation marks the base URI for the whole controller class. Then @GetMapping specifies the URI for the specific method. So, to access the 'getAllBlogs()' method, we first need to specify the @RequestMapping URI followed by the @GetMapping URI.

(7) Restful Web Services with Spring boot

← Classroom

REST API - GET POST
RESTful Web Services with Spring Boot

?

V

Problem Submissions Doubts

✓ RESTful Web Services with Spring Boot

Easy • Score 20/20

Send feedback

Problem statement

Which of the following is true about the @RestController annotation in Spring?

Options: Pick one correct answer from below

☐ It is used to create SOAP web services in Spring.

☒ It is used to create RESTful web services in Spring.


☐ It is used to create web services in Spring that use both REST and SOAP.

☐ It has no connection to web services in Spring.

Solution description

The @RestController annotation is a specialized version of the @Controller annotation in Spring, used to build RESTful web services. It is used to annotate controller classes in Spring and indicates that the class should handle RESTful HTTP requests.

(8) Restful Web Services with Spring boot – 2

← 

REST API - GET POST
RESTful Web Services with Spring Boot

Problem Submissions Doubts

✓ RESTful Web Services with Spring Boot

Score 20/20

Problem statement

Send feedback

The @PathVariable annotation is used to bind a method parameter to a request header

Options:

Pick one correct answer from below


☐ True

☒ False

Solution description

The @PathVariable annotation is used to bind a method parameter to a URI variable, not a request header.

(9) Implementing the POST Method

←  Classroom

REST API - GET POST
Implementing the POST Method

Problem Submissions Doubts

✓ Implementing the POST Method

Easy • Score 20/20

Problem statement

Send feedback

Imagine you are building a Spring Boot web application that allows users to add a blog using the post method of RESTful API in Spring Boot.

You have a Blog Model class as follows:

```
public class Blog {  
    private long blogId;  
    private String content;  
  
    //getter and setter  
}
```

Choose the best way to implement the post method using @PostMapping and @RequestBody annotations in a Spring Boot controller class.

Options: Pick one correct answer from below

☐

```
***  
@PostMapping("/blogs")  
@RequestBody(blog)  
public void createBlog(Blog blog) {  
    // implementation goes here  
}
```

☒

```
***  
@PostMapping("/blogs")  
public void createBlog(@RequestBody Blog blog) {  
    // implementation goes here  
}
```

☐

```
***  
@PostMapping("/blogs")  
public void createBlog(Blog blog) {  
    @RequestBody(blog)  
    // implementation goes here  
}
```

☐ All of the above

Solution description

@RequestBody is an annotation that is used with a method parameter to bind the incoming HTTP request body to the object specified. So in the case of (@RequestBody Blog blog), the incoming JSON body will be converted to the 'blog' object. The JSON body which is being sent from user:

(10) Implementing the POST Method

← Classroom

REST API - GET POST
Implementing the POST Method

Problem Submissions Doubts

✓ Implementing the POST Method

Easy • Score 20/20

Send feedback

Problem statement

Imagine you are building a Spring Boot application to store the details of all students in your class. You have already built this controller class as given below. The Student class is as follows:

```
public class Student {  
    private String name;  
    private Long Id;  
}
```

The controller class is as follows:

```
@RestController  
public class StudentController {  
    @Autowired  
    StudentService studentService;  
    @PostMapping('/student')  
    public void addStudent(Student student){  
        this.studentService.addStudent(student);  
    }  
}
```

Select the best practice to implement the Service class.

```
public class StudentService {  
    public void addStudent(Student student){  
        List<Student> studentList = new ArrayList<>();  
        studentList.add(student);  
    }  
}
```

```
public class StudentService {  
    public void addStudent(Student student){  
        List<Student> studentList = new ArrayList<>();  
        Map<Long, Student> studentMap = new HashMap<>();  
        studentList.add(student);  
        studentMap.put(student.getId(), student);  
    }  
}
```

```
public class StudentService {  
    private List<Student> studentList = new ArrayList<>();  
    private Map<Long, Student> studentMap = new HashMap<>();  
    public void addStudent(Student student){  
        this.studentList.add(student);  
        this.studentMap.put(student.getId(), student);  
    }  
}
```

```
public class StudentService {  
    private List<Student> studentList = new ArrayList<>();  
    public void addStudent(Student student) {  
        this.studentList.add(student);  
    }  
}
```

(11) Implementing the GET Method

← Classroom

REST API - GET POST
Implementing the GET Method

Problem Submissions Doubts

✓ Implementing the GET Method

Easy • Score 20/20

Send feedback

Problem statement

You are building a web application with a RESTful API using the Spring Framework. You have created a Controller class and you want to map a specific HTTP GET request to a method in your controller. You should use the _____ annotation to do this.

Options: Pick one correct answer from below

☐ @PostMapping

☒ @GetMapping

☐ @RequestMapping

☐ @ResponseMapping

Solution description

@GetMapping is used to map a GET request to a method in our controller. @PostMapping is used to map a POST request, and @RequestMapping is used to map a base URL for the controller class. There is no @ResponseMapping annotation in Spring Boot.

(12) Implementing the GET Method

← Classroom

REST API - GET POST
Implementing the GET Method

?

V

Problem Submissions Doubts

✓ Implementing the GET Method

Easy • Score 20/20

Problem statement

Send feedback

You are building a web application with a RESTful API using the Spring Framework. You have created a Controller class with a method that should handle an HTTP GET request to the URL `/blogs/{blogId}`. You want to extract the `blogId` value from the URL and use it in your method. Which annotation should you use to achieve this?

Options: Pick one correct answer from below

☐ @RequestMapping

☐ @ResponseBody

☒ @PathVariable

☐ @RequestParam

Solution description

The `@PathVariable` annotation extracts a value from the URL path and binds it to a method parameter. In this scenario, you would use the `@PathVariable` annotation to bind the `blogId` value from the URL to a method parameter.