

1. REST is built upon the following existing technologies (select all that apply):

Show Results

20/20 Students Answered

- A HTML ✗
- B JSON ←
- C HTTP ←
- D Java ✗

—

3. REST is designed to act upon resources. URLs specify:

Show Results

20/20 Students Answered

- A the resource — ✓
- B the action ⚡ Http Request methods - GET, POST, PUT, and DELETE -- not part of the URL
- C both ✗
- D neither ✗

4. Which HTTP method and URL should be used to retrieve post 313?

Show Results

20/20 Students Answered

A GET /blog/posts?id=313

B PULL /blog/posts/313

C GET /blog/posts/313

D GET /blog/posts/{313}

← query string parameters are typically used for filter (not usually used for resource info)

6. Which HTTP method and URL should be used to add a comment to post 87?

Show Results

20/20 Students Answered

A POST /blog/posts/87/comments/1

B POST /blog/posts/87/comments

C PUT /blog/posts/87/comments

D POST /blog/posts/comments/87

← convention is that the post id follows the /posts/ and then append comments on end for this POST request

8. Validation annotations are specified on the:

Show Results

20/20 Students Answered

A controller

B view

C model

D DAO

Yes we put @VALID in the controller, but it references (is placed in front of) the @RequestBody which is a MODEL!

9. Assuming the Product DAO is in place and there are no validation errors, what does the following code do?

```
@RequestMapping( path = "/products", method =  
RequestMethod.PUT)  
public Product update(@RequestBody Product product,  
@PathVariable int id) {  
    productDao.update(product, id);  
    return product;  
}
```

Show Results

20/20 Students Answered

- A compile and run correctly ☒
- B fail to compile ☒
- C fail at runtime ☒
- D too ambiguous to tell ☒

path = "/products/{id}"