

Web Engineering Multiple Choice Questions Final Test 2

You are tasked with finding the right web server platform that adheres to the following requirements: You want to manage the server software on your own, but the platform should provide you with the ability to retrieve computational resources and capacity on the fly when explicitly calling endpoints of the platform. What server concept we heard about in the lecture is the closest related to this description?

Given the following request to your API:
`https://localhost/frame/oak.jpg?quality=large`

An implemented endpoint:
(Please ignore the blatant security issues with this code)

```
app.get('/frame/:style.:ending', (req, res) => {
  if(!req.query.quality) {
    return res.sendStatus(404);
  }
  res.sendFile(path.join(__dirname, "../resources/frame/" + req.query.quality, `${req.params.style}.${req.params.ending}`));
});
```

Which of the following answers captures best what is going on?

What is a session cookie?

What is middleware (as part of backend abstractions/Node.js)?

What are declarative frontends?

In the following component, we see the use of a computed property. What is wrong in this example?

```
<template>
  <label>From <input type="text" v-model="range.min"></label>
  <label>To <input type="text" v-model="range.max"></label>
  {{ randomInRange }}
</template>
<script>
export default {
  name : 'RandomNumber',
  data : function {
    return {
      range : {
        min: 4,
        max: 8
      }
    }
  },
  computed : {
    randomInRange : function() {
      Math.floor(Math.random() * (range.max - range.min + 1)) + range.min
    }
  }
}
</script>
```

Which one of these concepts in frontend abstractions best describes data binding in templates where changes in the input can also influence the model?

You are given a Vue component called "FromToRange" that contains one input field. Below, we want to use that component and bind it to the "range" object in our model (data) in a way that when min or max in the model change, the form input takes the form `\${min}-\${max}` within the FromToRange component (so if we set the object in data to range: { min: 4, max: 8 }, the input in the component should display "4-8").

And, vice-versa, when the form input in the "FromToRange" component changes, the values of the "range" objects also change (so if the user types "9-14" into the form input, the object in data is set to range: { min: 9, max: 14 }).

```
<template>
  <FromToRange />
</template>
<script>
import FromToRange from "@components/FromToRange";
export default {
  components : {
    FromToRange
  }
  data : () => {
    return {
      range : {
        min : 0,
        max: 0
      }
    }
  }
}
</script>
```

Which of the following answers describes a best practice solution to enable reactive and bidirectional binding for this component in Vue?

Given the following code that uses HTML/JavaScript with DOM and Event Handling:

```
<html>
<body>
  <input type="range" id="numSlider" oninput="updateValue(this.value)">
  <input type="text" id="numInput" onchange="updateValue(this.value)">
  <script>
    function updateValue(value) {
      const span = getSpan('mm');
      if(isNaN(value)) {
        document.body.removeChild(span);
        return;
      }
      document.getElementById('numSlider').value = value;
      document.getElementById('numInput').value = value;
      span.innerText = `${value * 10} mm`;
    }
    function getSpan(id) {
      let el = document.getElementById(id);
      if(!el) {
        el = document.createElement('span');
        el.id = id;
        document.body.appendChild(el);
      }
      return el;
    }
  </script>
</body>
</html>
```

Which of the following Vue components is the equivalent to this code above that most closely follows declarative rendering with bidirectional binding?

Here is an endpoint in Node.js:

```
app.post('/users/:id', (res, req) => {  
  switch (req.query.method) {  
    case 'addBlogPost':  
      // add a post to the user's blog  
      ...  
      return res.sendStatus(200);  
  
    case 'deleteBlogPost':  
      // delete a post from the user's blog  
      ...  
      return res.sendStatus(204);  
  
    default:  
      return res.sendStatus(400);  
  }  
});
```

What REST principle is violated here?

A web service API from a webshop offers a "/user" endpoint. When retrieving a specific user, the JSON representation also contains a reference/links to orders from the user. Which REST principle closest fits this description of aiding exploration in web services?

You want to issue a dynamic query from the client that would include all fields from a schema (even if the schema evolves over time to include more or less fields). What concept in GraphQL best describes a technique that would enable you to do that?

Which of the following descriptions best illustrates GraphQL?

What is the advantage of using a state management system, such as Vuex, over simply changing global state?

Your web service depends on a database. For development you have a local instance of the database running on localhost. When deploying your application to production, you connect to an external database server. What backend abstraction would you use to manage the database credentials within your code?

© Jürgen Cito, Michael Schröder (TU Wien)