

JS Advanced Exam – 13 March 2022

3. Rent Car

Your Task

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **rentCar**, which represents an object. You may use the following code as a template:

```
describe("Tests ...", function() {
  describe("TODO ...", function() {
    it("TODO ...", function() {
      // TODO: ...
    });
  });
});
// TODO: ...
});
```

The object that should have the following functionality:

- **searchCar(shop, model)** - A function that accepts two parameters (one array and one string):
 - The function checks whether the submitted string **model** is present in the **shop** (example: ["Volkswagen", "BMW", "Audi"]), and return number of matching elements and the model of the car in the message: ``There is ${findModel.length} car of model ${model} in the catalog!``;
 - There is a need for validation of the input, a shop and a model may not always be valid. In case of submitted invalid parameters, **throw** an error **"Invalid input!"**;
 - If there are no matching elements, the function **throw** an error: **'There are no such models in the catalog!'**
- **calculatePriceOfCar(model, days)** - A function that accepts two parameters (string and number):
 - There is a need for validation of the input, a **model**, and **days** may not always be valid. In case of submitted invalid parameters, **throw** an error **"Invalid input!"**;
 - The function returns the model and the price it will cost for renting a car for the given days: ``You choose ${model} and it will cost ${cost}!``;
 - Otherwise, if there is no such model, the function **throw** an error: **'No such model in the catalog!'**.
- **checkBudget(costPerDay, days, budget)** - A function that accepts three parameters (numbers):
 - There is a need for validation of the input, a **costPerDay**, **days**, and a **budget** may not always be valid. In case of submitted invalid parameters, **throw** an error **"Invalid input!"**;
 - If the budget is bigger or equal to cost, function return: ``You rent a car!``;
 - If the budget is less than cost, the function returns the message: **'You need a bigger budget!'**.

JS Code

To ease you in the process, you are provided with an implementation that meets all of the specification requirements for the **rentCar** object:

rentCar.js

```
const rentCar = {
```

```

searchCar(shop, model) {
  let findModel = [];
  if (Array.isArray(shop) && typeof model == 'string') {
    for (let i = 0; i < shop.length; i++) {
      if (model == shop[i]) {
        findModel.push(shop[i]);
      }
    }
    if (findModel.length !== 0) {
      return `There is ${findModel.length} car of model ${model} in the
catalog!`;
    } else {
      throw new Error('There are no such models in the catalog!')
    }
  } else {
    throw new Error('Invalid input!')
  }
},
calculatePriceOfCar (model, days) {
  let catalogue = {
    Volkswagen: 20,
    Audi: 36,
    Toyota: 40,
    BMW: 45,
    Mercedes: 50
  };

  if (typeof model == 'string' && Number.isInteger(days)) {
    if (catalogue.hasOwnProperty(model)) {
      let cost = catalogue[model] * days;
      return `You choose ${model} and it will cost ${cost}!`
    } else {
      throw new Error('No such model in the catalog!')
    }
  } else {
    throw new Error('Invalid input!')
  }
},
checkBudget(costPerDay, days, budget) {
  if (!Number.isInteger(costPerDay) || !Number.isInteger(days) ||
!Number.isInteger(budget)) {
    throw new Error('Invalid input!');
  } else {
    let cost = costPerDay * days;
    if (cost <= budget) {
      return `You rent a car!`
    } else {
      return 'You need a bigger budget!'
    }
  }
}
}

```

Submission

Submit your tests inside a **describe()** statement, as shown above.