



Client Developer  
Challenge

# quick facts



---

## Objective

- Develop functionality for a simple webapp using Javascript/Typescript (No backend is required)
- The functions should be executable from the developer console
- A small project template is provided that needs to be extended
- Only develop what is explicitly requested - when in doubt ask!

---

## Conditions

- Write code that works
- Write pretty, readable and well refactored code

---

## Preparations

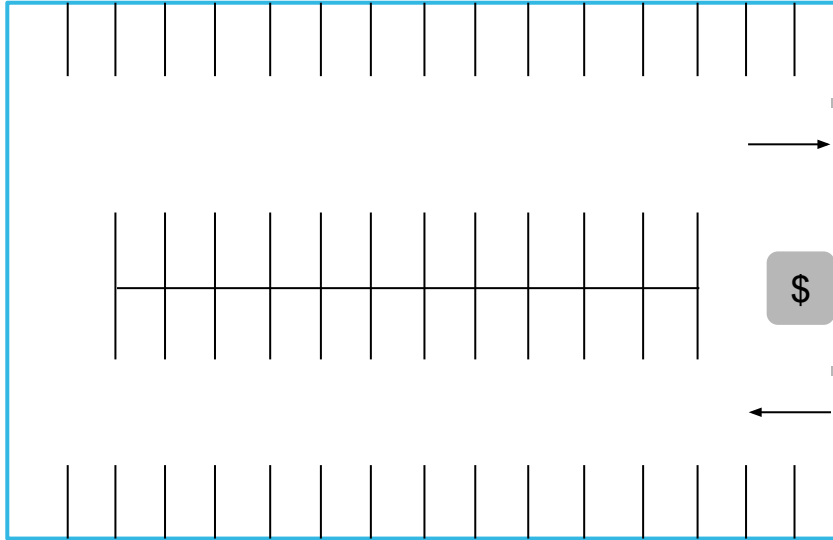
- Create a new project and init it as a private Git repo
- Add the provided project code from the zip file and push it
- Create a feature branch before implementing the tasks

---

## Rules

- Make a Git commit at least after every finished task
- Be “fair” - only read the next task after you have committed the results of the previous task.
- We will talk about your refactoring steps that you have done between tasks!

# parking lot



The goal is to create an application for a parking lot company with 54 spaces.

You can use any frameworks and packages.

When a car enters it gets a ticket and when the car leaves the ticket will be returned. The browser is a simulated ticket machine



**good luck!**

# task #1



- When a car enters the parking lot a new ticket with a unique barcode is given out
- Create a function that calculates and returns a new barcode (16 digits). Save the ticket and time when the ticket is given out for later reference.
- Add persistence so that tickets are also available after reloading the web page

```
getTicket();
```

# task #2



- Before exiting the parking lot the car owner must pay the fee at the cash-machine. Tickets are identified by the barcode. First we need to display the price to pay.
- Create a function to calculate the price for the parking time. Every started hour costs €2.

```
calculatePrice (barcode) ;
```

# task #3



- When the machine displays the correct price the person will make the payment with the option of her/his choice (credit card, debit card, cash). This payment process is not part of this app.
- Create a function that marks the ticket as paid and saves the time and payment option used.
- When paid, the function from task #2 should return price 0 and a payment receipt.

```
payTicket (barcode, paymentMethod) ;
```

# task #4



- To exit the parking lot the ticket must be returned at the gate.
- The gate opens only
  - if the ticket has been paid (payment is always for a full hour)
  - if not more than 15min have passed since the payment otherwise the person needs to make another payment.
- Create a function that returns the state (paid or unpaid) of the ticket depending on these conditions.

```
getTicketState (barcode) ;
```

# task #5



- Next to the entry of the parking lot you want to create a sign displaying the amount of free parking spaces (Total parking spaces minus given out tickets).
- Create a function that calculates the number of currently available spaces. Make sure that there can't be more cars in the parking lot than available parking spaces.
- `getFreeSpaces()` ;



# submission



- Create a PR for the branch
- Give us access to your repo  
([dominik@platogo.com](mailto:dominik@platogo.com) &  
[lukas@platogo.com](mailto:lukas@platogo.com) & [vanja@platogo.com](mailto:vanja@platogo.com))
- We will ask you at the interview about your endpoints and the decisions you took.

See you soon!

