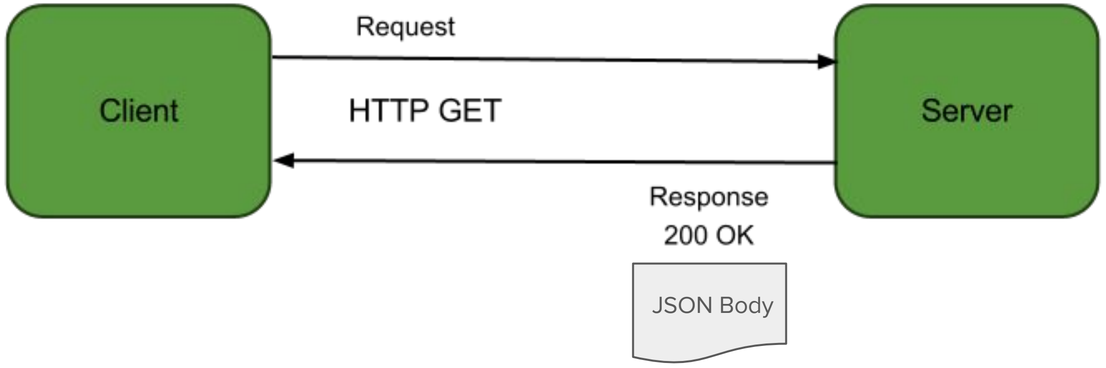
WEB SERVICES
With Axios & Vue

Can You?

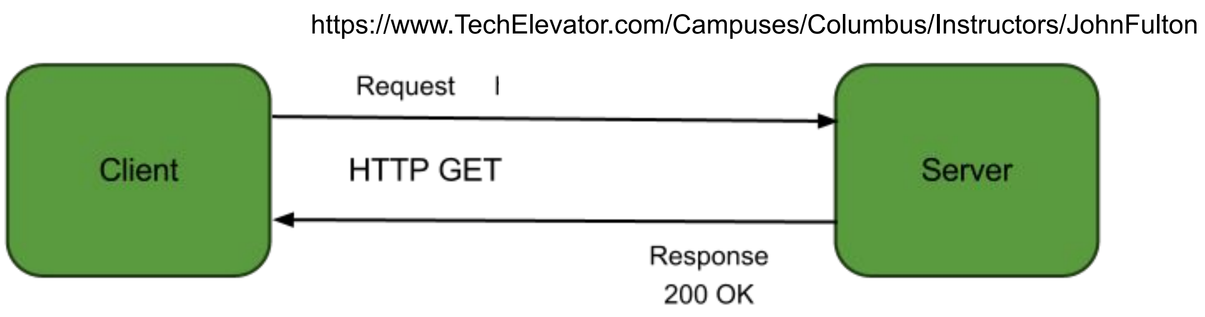
- Explain the process of a typical HTTP request between a web browser and a server
- Explain what a GET request is used for
- Explain that a 2xx Status Code indicates "success"
- Make an HTTP GET request using Postman and inspect the result
- Explain what JSON is and use it in a JavaScript program
- Make an HTTP GET request to a RESTful web service using the Axios library and process the response
- Build a service object for interacting with a RESTful web service
- Use the Vue lifecycle hook `created()` to call a web service to retrieve data when a view is rendered
- Explain the difference between synchronous and asynchronous code
- Explain what a promise is and how it works
- Explain why asynchronous coding techniques are frequently used in JavaScript for interacting with server-side components

REST

REST



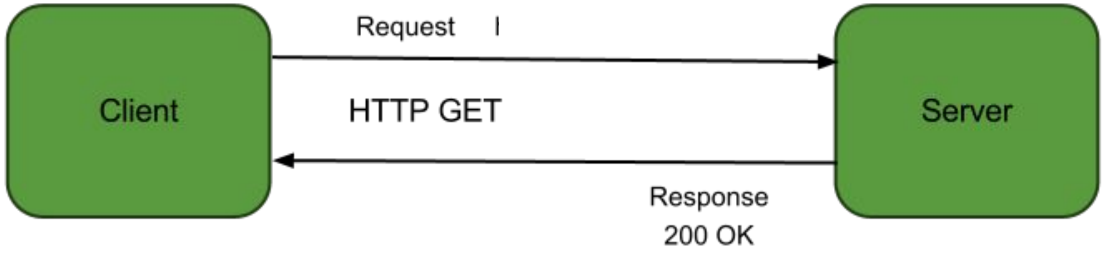
REST


https://www.TechElevator.com/Campuses/Columbus/Instructors/JohnFulton
Request
Client
HTTP GET
Server
Response
200 OK

{ firstName: “John”, lastName: “Fulton”, classroom: “.NET”, isAwesome: true}

REST

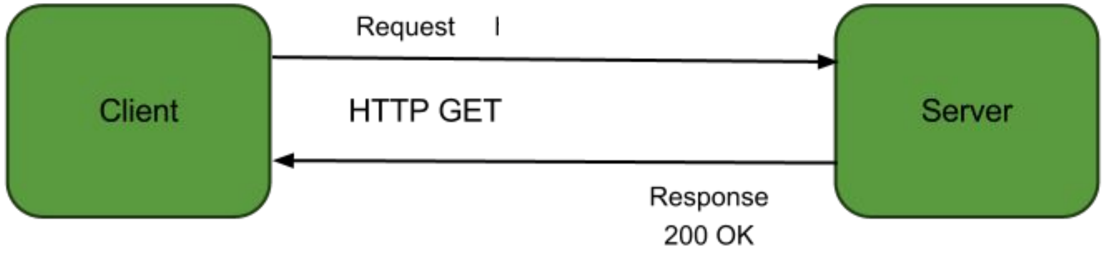
https://www.TechElevator.com/Campuses/Columbus/Instructors/MattEland



{ firstName: “Matt”, lastName: “Eland”, classroom: “Wherever”, isAwesome: undefined}

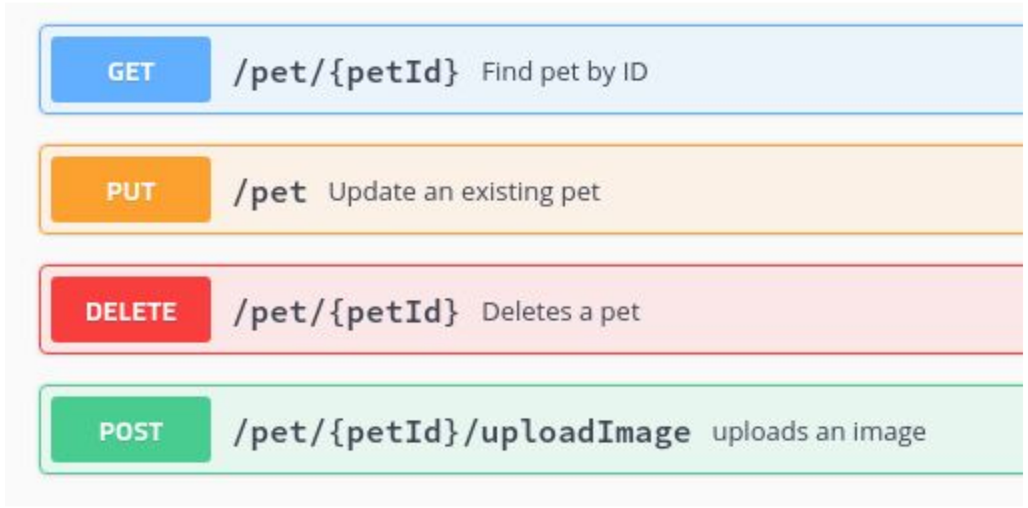
REST

https://www.TechElevator.com/Campuses/Columbus/Instructors



```
[
 { firstName: “John”, lastName: “Fulton”, classroom: “.NET”, isAwesome: true},
 { firstName: “Matt”, lastName: “Eland”, classroom: “Wherever”, isAwesome: undefined},
 // … more instructors here
]
```

VERBS & RESOURCES

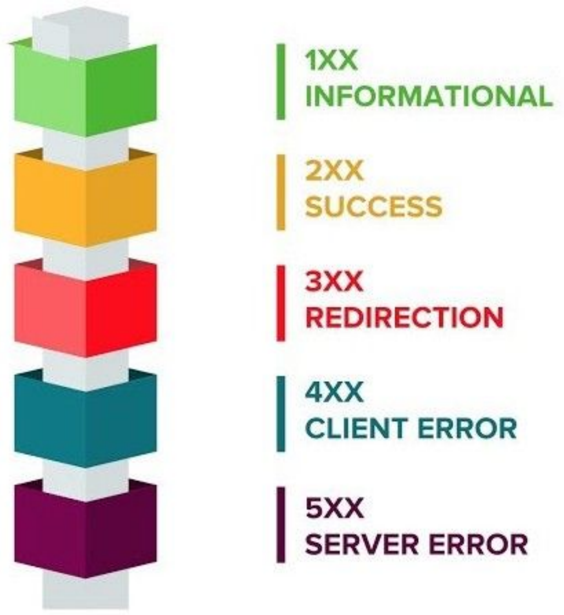

GET /pet/{petId} Find pet by ID

PUT /pet Update an existing pet

DELETE /pet/{petId} Deletes a pet

POST /pet/{petId}/uploadImage uploads an image

STATUS CODES



HTTP STATUS CODES

2xx Success

200	Success / OK

3xx Redirection

301	Permanent Redirect
302	Temporary Redirect
304	Not Modified

4xx Client Error

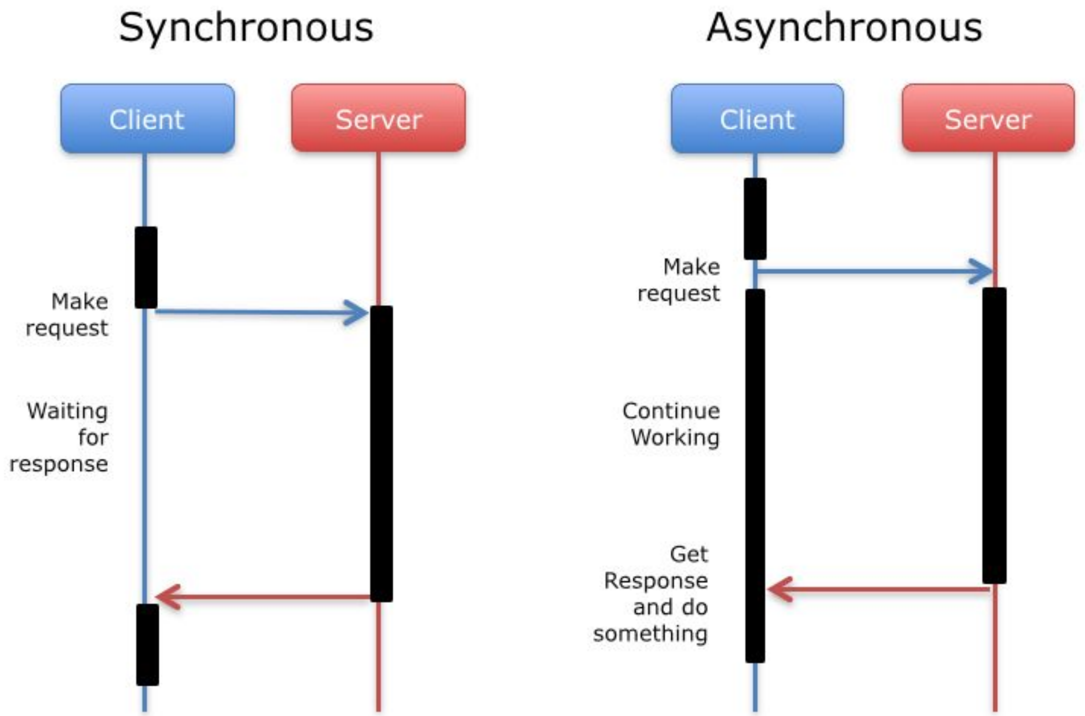
401	Unauthorized Error
403	Forbidden
404	Not Found
405	Method Not Allowed

5xx Server Error

501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

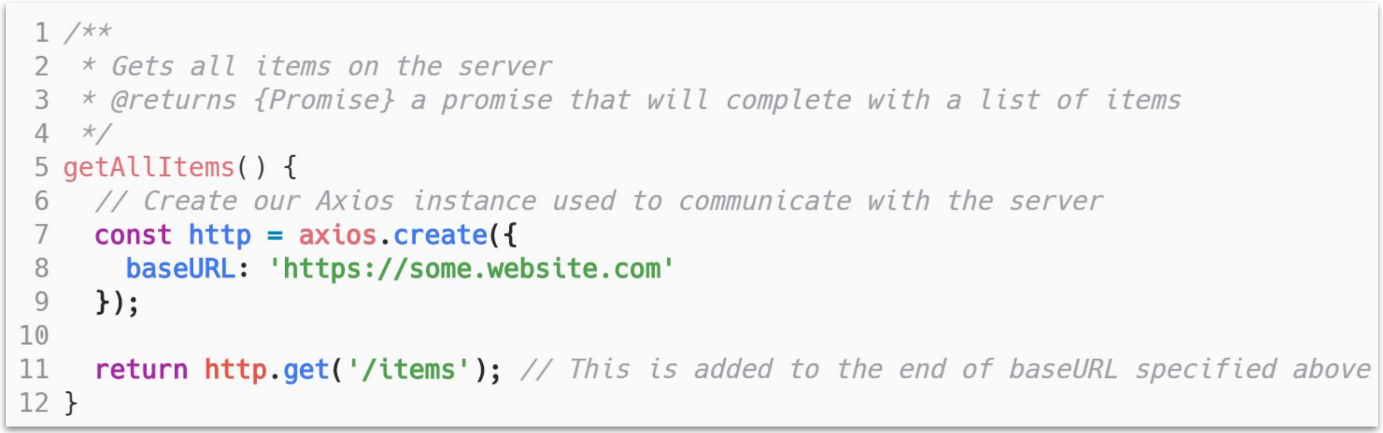
SYNCHRONOUS & ASYNCHRONOUS

SYNC VS ASYNC



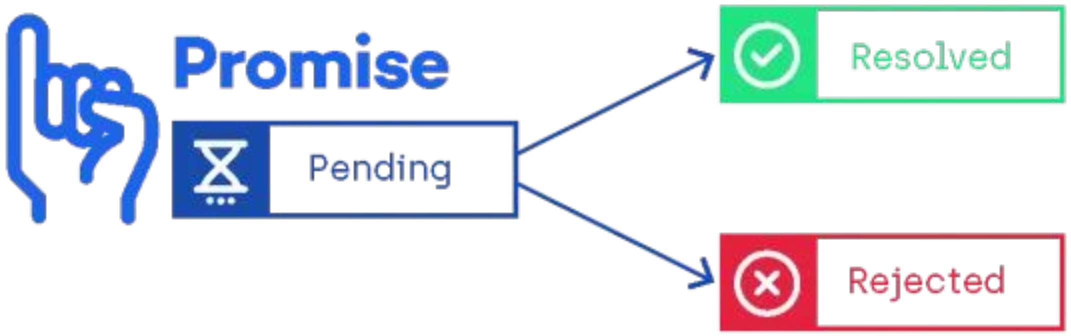
AXIOS

AXIOS GET

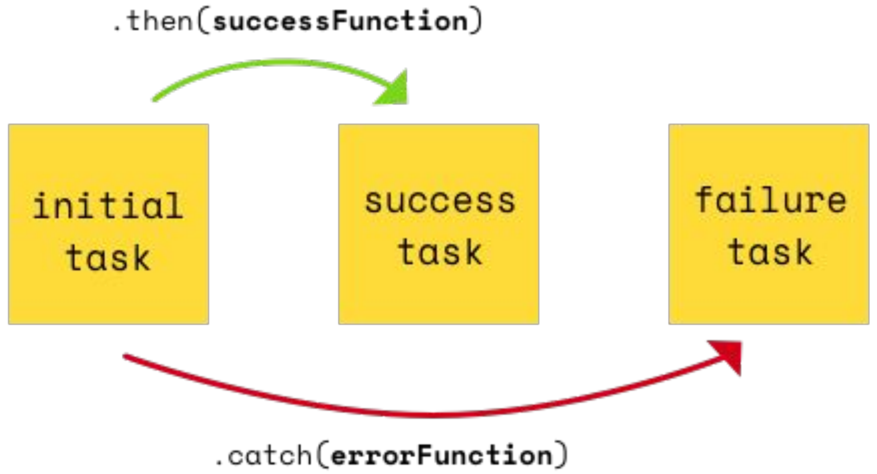


```
/**
 * Gets all items on the server
 * @returns {Promise} a promise that will complete with a list of items
 */
getAllItems() {
  // Create our Axios instance used to communicate with the server
  const http = axios.create({
    baseURL: 'https://some.website.com'
  });

  return http.get('/items'); // This is added to the end of baseURL specified above
}
```

WHAT IS A PROMISE?



USING A PROMISE

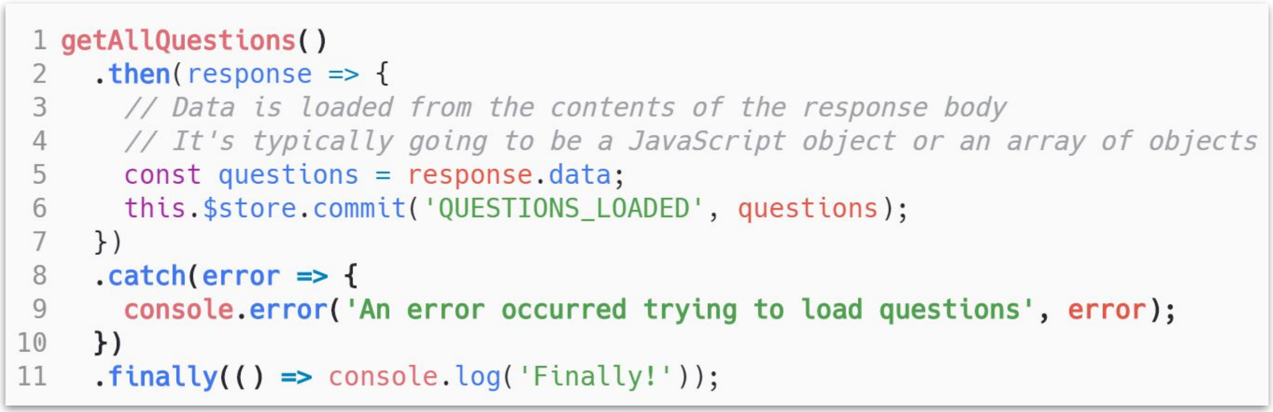


AXIOS GET

```
/**
 * Gets all items on the server
 * @returns {Promise} a promise that will complete with a list of items
 */
getAllItems() {
  // Create our Axios instance used to communicate with the server
  const http = axios.create({
    baseURL: 'https://some.website.com'
  });

  return http.get('/items'); // This is added to the end of baseURL specified above
}
```

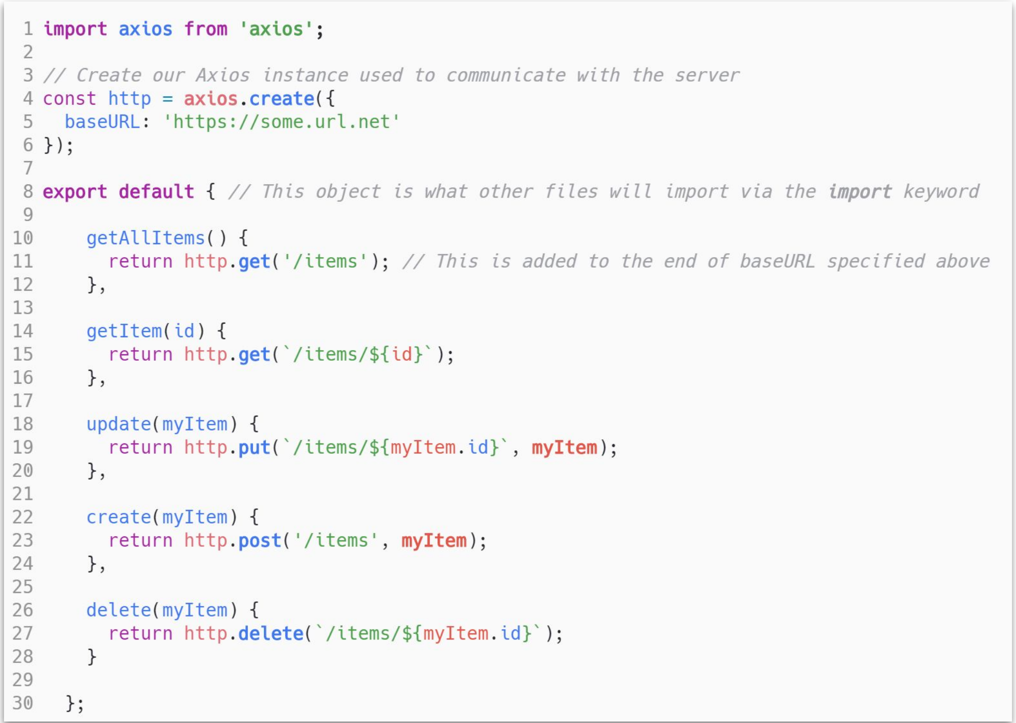
```
getAllItems().then(response => {
    // response.data is loaded from the contents of the response body
    // It's typically going to be a JavaScript object or an array of objects
    const items = response.data;
    this.$store.commit('ITEMS_LOADED', items);
  });
```

PROMISE ERROR HANDLING



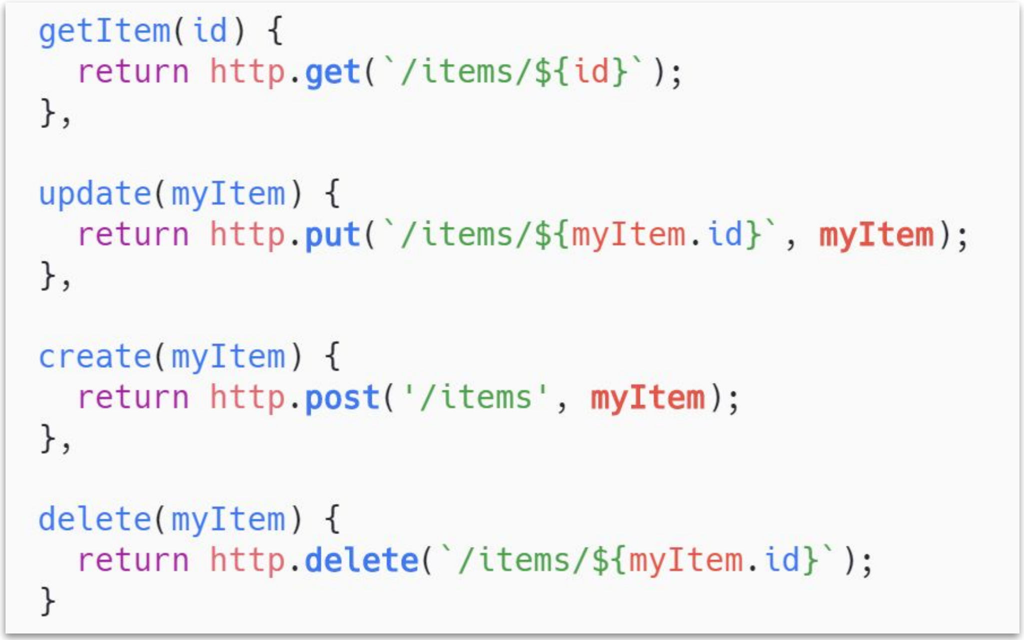
```javascript
getAllQuestions()
  .then(response => {
    // Data is loaded from the contents of the response body
    // It's typically going to be a JavaScript object or an array of objects
    const questions = response.data;
    this.$store.commit('QUESTIONS_LOADED', questions);
  })
  .catch(error => {
    console.error('An error occurred trying to load questions', error);
  })
  .finally(() => console.log('Finally!'));
```

SERVICES



```javascript
import axios from 'axios';

// Create our Axios instance used to communicate with the server
const http = axios.create({
  baseURL: 'https://some.url.net'
});

export default { // This object is what other files will import via the import keyword

    getAllItems() {
      return http.get('/items'); // This is added to the end of baseURL specified above
    },

    getItem(id) {
      return http.get(`/items/${id}`);
    },

    update(myItem) {
      return http.put(`/items/${myItem.id}`, myItem);
    },

    create(myItem) {
      return http.post('/items', myItem);
    },

    delete(myItem) {
      return http.delete(`/items/${myItem.id}`);
    }

  };
```

AXIOS VERBS



```
getItem(id) {
  return http.get(`/items/${id}`);
},

update(myItem) {
  return http.put(`/items/${myItem.id}`, myItem);
},

create(myItem) {
  return http.post('/items', myItem);
},

delete(myItem) {
  return http.delete(`/items/${myItem.id}`);
}
```

HAPPY
CODING!