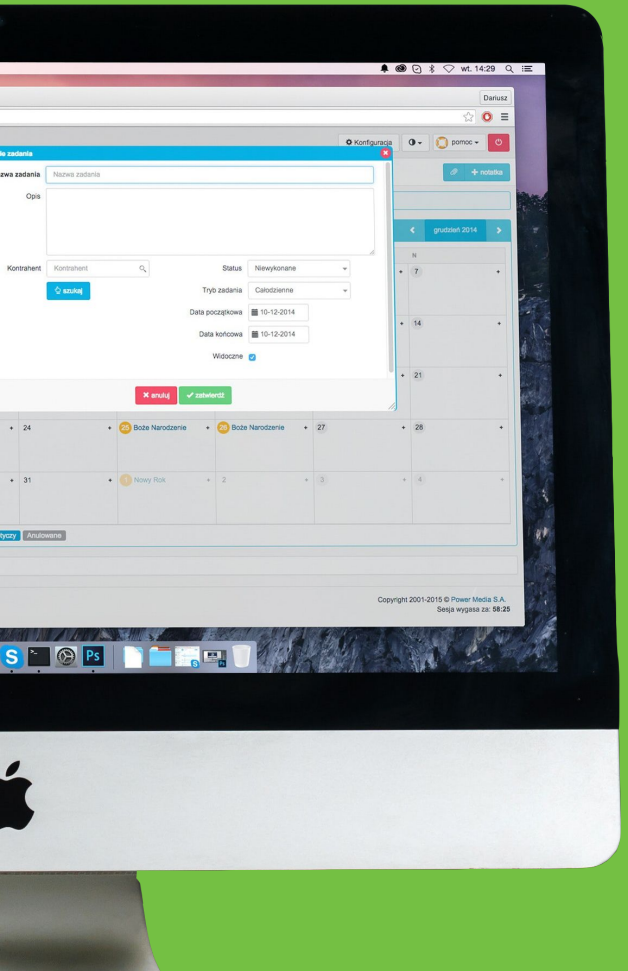

4CS017

Internet Software Architecture

ISA

Lecture Week 5

API, JSON

This week's agenda

- Understanding the API
- JSON
- JSON Data
- Accessing JSON

Getting Started

When you use an application, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your system. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API.

Let us take a familiar example.

Imagine you're sitting at a table in a restaurant with a menu of choices to order from. The kitchen is the part of the "system" that will prepare your order. What is missing is the critical link to communicate your order to the kitchen and deliver your food back to your table. That's where the waiter or API comes in. The waiter is the messenger – or API – that takes your request or order and tells the kitchen – the system – what to do. Then the waiter delivers the response back to you; in this case, it is the food.

1. Web API

1.1. What is Web API?

- Browser APIs
- Third party APIs

1.2. Working with APIs in JavaScript

2.JSON

2.1 What is JSON?

2.2. JSON structure

2.3. JSON syntax

2.4. Why use JSON?

2.5. JSON Data

2.6. JSON Object

2.7. JSON Array

2.8. Accessing JSON Data

3. JavaScript Objects VS JSON

4. Learn More

A Web API is a developer's dream.

- It can extend the functionality of the browser
- It can greatly simplify complex functions
- It can provide easy syntax to complex code

What is Web API?

- API stands for Application Programming Interface.
- A Web API is an application programming interface for the Web.
- A Browser API can extend the functionality of a web browser.
- A Server API can extend the functionality of a web server.

Browser APIs

All browsers have a set of built-in Web APIs to support complex operations, and to help accessing data.

For example, the Geolocation API can return the coordinates of where the browser is located.

```
const myElement = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    myElement.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  myElement.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
```

Third Party APIs

Third party APIs are not built into your browser.

To use these APIs, you will have to download the code from the Web.

Examples:

- YouTube API - Allows you to display videos on a web site.
- Twitter API - Allows you to display Tweets on a web site.
- Facebook API - Allows you to display Facebook info on a web site.

Working with APIs in JavaScript

An API is simply a medium to fetch or send data between interfaces. Let's say you want to make an application that provides the user with some real-time data fetched from the server or maybe even allows you to modify or add data to some other endpoint. This is made possible by the API or the Application Programming Interface.


Any
Questions

1. Web API
1.1. What is Web API?
- Browser APIs
- Third party APIs

1.2. Working with APIs in JavaScript

2.JSON

2.1 What is JSON?
2.2. JSON structure
2.3. JSON syntax
2.4. Why use JSON?
2.5. JSON Data
2.6. JSON Object
2.7. JSON Array
2.8. Accessing JSON Data

3. JavaScript Objects VS JSON
4. Learn More

What is JSON?

- JSON stands for JavaScript Object Notation
- JSON is a text format for storing and transporting data
- JSON is "self-describing" and easy to understand
- JSON is often used when data is sent from a server to a web page
- The JSON format is syntactically identical to the code for creating JavaScript objects. Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
- The JSON syntax is derived from JavaScript object notation syntax, but the JSON format is text only. Code for reading and generating JSON data can be written in any programming language.

JSON structure

literal format

ı in a standard

```
{
  "string": "Hi",
  "number": 2.5,
  "boolean": true,
  "null": null,
  "object": { "name": "Kyle", "age": 24 },
  "array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],
  "arrayOfObjects": [
    { "name": "Jerry", "age": 28 },
    { "name": "Sally", "age": 26 }
  ]
}

```

JSON syntax

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

Why Use JSON?

- The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.
- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.
- JavaScript has a built in function for converting JSON strings into JavaScript objects:

```
JSON.parse()
```

- JavaScript also has a built in function for converting an object into a JSON string:

```
JSON.stringify()
```

JSON Data

JSON data consists of key/value pairs similar to JavaScript object properties. The key and values are written in double quotes separated by a colon :. For example,

```
// JSON data
  "name": "John"
```

Note: JSON data requires double quotes for the key.

JSON Object

The JSON object is written inside curly braces { }. JSON objects can contain multiple key/value pairs. For example,

```
// JSON object
  { "name": "John", "age": 22 }
```

JSON Array

JSON array is written inside square brackets [ ]. For example,

```
// JSON array

[ "apple", "mango", "banana"]


// JSON array containing objects

[
    { "name": "John", "age": 22 },
    { "name": "Peter", "age": 20 },
    { "name": "Mark", "age": 23 }
]
```

Accessing JSON Data

You can access JSON data using the dot notation. For example,

```
// JSON object
const data = {
    "name": "John",
    "age": 22,
    "hobby": {
  "reading" : true,
  "gaming" : false,
  "sport" : "football"
    },
    "class" : ["JavaScript", "HTML", "CSS"]
}
// accessing JSON object
console.log(data.name); // John
console.log(data.hobby); // { gaming: false, reading: true, sport: "football"}

console.log(data.hobby.sport); // football
console.log(data.class[1]); // HTML
```

JavaScript Objects VS JSON

Though the syntax of JSON is similar to the JavaScript object, JSON is different from JavaScript objects.

JSON	JavaScript Object
The key in key/value pair should be in double quotes.	The key in key/value pair can be without double quotes.
JSON cannot contain functions.	JavaScript objects can contain functions.
JSON can be created and used by other programming languages.	JavaScript objects can only be used in JavaScript.

Before you come for Lab, Research!!

- Working with APIs in JavaScript - GeeksforGeeks
- Web APIs (w3schools.com)
- JSON Introduction (w3schools.com)
- JSON
- Working with JSON - Learn web development | MDN (mozilla.org)


THANK
YOU