Övning 10: JavaScript och Webb API:er

فراخوانی API و ارائه داده در DOM را تمرین کنید.

API مخفف Application Programming Interface است. یک web api از یک یا چند نقطه پایانی عمومی تشکیل شده است. اغلب، از پروتکل http استفاده می شود و نقطه پایانی به سادگی یک uri (آدرس وب) است که در آن داده هایی که می خواهیم بازیابی کنیم در سرور قرار دارد.

کل سیستم بر اساس درخواست و پاسخ است. اگر درخواستی را به نقطه پایانی https://api.github.com/users/NET22B/repos ارسال کنیم، پاسخی با تمام مخازن تنظیم شده برای کلاس دریافت می کنیم (لطفاً آن را در مرورگر خود قرار دهید و ببینید آنچه را پس می گیرید). اگر می‌خواهیم یک وب‌سایت برای پیش‌بینی آب‌وهوا بسازیم، می‌توانیم به جای جمع‌آوری داده‌ها، به یک نقطه پایانی در API MetaWeather درخواست کنیم. Api های مختلف از فرمت های داده متفاوتی پشتیبانی می کنند اما json و xml رایج ترین آنها هستند.

این ما را قادر می‌سازد تا از منابع (داده‌های) موجود در api بدون در نظر گرفتن اینکه با کدام زبان برنامه‌نویسی توسعه می‌دهیم و فارغ از اینکه api با چه زبانی ساخته شده است، استفاده کنیم. URI مخفف Uniform Resource Identifier است و برای درخواست به کتابخانه های API استفاده می شود.



روش‌های درخواست HTTP که افعال HTTP نیز نامیده می‌شوند، نشان می‌دهند که کدام عمل را می‌خواهید با درخواست خود انجام دهید. پنج مورد از رایج ترین آنها در جدول زیر نشان داده شده است. برای مثال اگر بخواهیم داده ها را بازیابی کنیم، درخواست GET می کنیم. تنها چیزی که باید در این تمرین استفاده کنید GET است!

|  |  |
| --- | --- |
| HTTP-VERB | CRUD |
| POST | Create |
| GET | Read |
| PUT | Update/Replace (full update) |
| PATCH | Update/Modify (partiell update) |
| DELETE | Delete |

بدنه درخواست/پاسخ داده‌ای است که معمولاً در قالب JSON ارسال می‌شود.

{  
 "object": {  
 "a": "b",  
 "c": "d",  
 "e": "f"  
 },  
 "array": [  
 1,  
 2  
 ],  
 "string": "Hello World"  
}

Postman برنامه ای است که می توان از آن برای فراخوانی و آزمایش API ها در حین توسعه استفاده کرد.

https://www.postman.com/downloads/

وظیفه 1: بیومتریک جنگ ستارگان

Star Wars API حاوی داده های زیادی در مورد شخصیت ها، گونه ها، سفینه های فضایی، سیارات و غیره از فیلم های جنگ ستارگان است. مستندات: https://www.swapi.tech/documentation، برای مثال، می توانید با درخواست GET، همه کاراکترها را فهرست کنید:

https://www.swapi.tech/api/people

همچنین می توانید یک درخواست GET با یک رشته کوئری برای دریافت یک کاراکتر خاص ایجاد کنید:

https://www.swapi.tech/api/people/?name=chewbacca

وظیفه شما این است که صفحه ای بسازید که در آن کاربر نام یکی از شخصیت های جنگ ستارگان را وارد کرده و داده های بیومتریک را پس بگیرد. برخی از نام‌های نمونه عبارتند از Luke Skywalker، R2-D2، Darth Vader، Leia Organa، Chewbacca.

1. یک کادر متنی برای دریافت نام ایجاد کنید. از <input></input> استفاده کنید و به آن ویژگی type=”text” بدهید.
2. یک دکمه با <button></button> ایجاد کنید.
3. یک جعبه متن برای خروجی با <textarea></textarea> ایجاد کنید و حداقل 5 خط با ویژگی rows به آن بدهید.
4. یک تابع در فایل js بنویسید:
   1. مقدار جعبه متن ورودی را می خواند و آن را به رشته "https://www.swapi.tech/api/people/?name=" اضافه می کند تا یک URI کامل دریافت کنیم،
   2. از URI به عنوان پارامتری برای واکشی درخواست استفاده می کند.
   3. و اطلاعات را در جعبه متن خروجی چاپ می کند.

function getApi(){  
 /\*Skriv din kod här\*/  
 fetch(fullUri)  
 .then(res => res.json())  
 .then(data =>  
 {

/\*Och här\*/  
 })  
 .catch(err => console.log(err))  
}

داده یک شی JSON است که شامل تمام اطلاعات بازیابی شده در مورد کاراکتر است. اسناد را بررسی کنید یا با console.log پرینت بگیرید تا ببینید شیء چگونه ساخته شده است، ارتفاع، جرم، جنسیت و رنگ\_مو را استخراج کنید و از آنها برای ساختن رشته ای با استفاده از حروف الفبای الگو (اسلش معکوس) استفاده کنید. مثال:

let s = `Value 1: ${value1}, Value 2: ${value2}`;

1. یک eventListener اضافه کنید که به دکمه گوش می دهد و عملکرد بالا را با کلیک اجرا می کند.

**وظیفه 2: یک کارت بکشید**

Deck of Cards API به ما امکان می دهد از یک یا چند عرشه کارت بکشیم. در بدنه پاسخ، هم مقداری که کارت دارد (کت و شلوار و ارزش) و هم آدرس اینترنتی تصویر کارت کشیده شده را دریافت می کنید. مستندات:

http://deckofcardsapi.com/

شما باید صفحه ای بنویسید که هر بار یک کارت را نشان دهد که کاربر روی دکمه کلیک می کند.

1. یک <button></button> ایجاد کنید.
2. یک <div></div> ایجاد کنید که در آن تصویر روی کارت قرار است برش داده شود.
3. یک فایل js مشابه مانند کار قبلی با چند تفاوت بنویسید:
   1. از این URI برای واکشی استفاده کنید:

https://deckofcardsapi.com/api/deck/new/draw/?count=1

* 1. اسناد را بررسی کنید تا نحوه استخراج نشانی اینترنتی تصویر روی کارت را از داده ها مشاهده کنید.
  2. از createElement، setAttribute، innerHTML و appendChild برای ایجاد یک عنصر تصویر، تنظیم ویژگی src آن، تنظیم مجدد div و افزودن عنصر تصویر به آن استفاده کنید.

1. یک eventListener اضافه کنید که به دکمه گوش می دهد و عملکرد بالا را با کلیک اجرا می کند.

اضافی

یک بازی با ورق بسازید :) از اسناد میمون کمک بگیرید. شما انتخاب می کنید که چه چیزی بسازید. شاید یک بازی بلک جک ساده؟

Övning 10: JavaScript och Webb API:er  
*Öva på att kalla på API:er och presentera data i DOM:en.***Teori**

***API*** står för Application Programming Interface. Ett webb-api består av en eller flera publika endpoints. Oftast används http-protokollet och endpointen är helt enkelt en uri (webbadress) till var på serven data vi vill hämta ligger.  
Hela systemet bygger på request och response. Om vi skickar ett request till endpointen https://api.github.com/users/NET22B/repos får vi tillbaka en respons med alla repositories som är upplagda för klassen (testa gärna att klistra in detta i er browser och se vad ni får tillbaka). Om vi vill bygga en hemsida för väderprognoser kan vi göra förfrågan (requests) till en endpoint i MetaWeathers API istället för att samla data själva. Olika api:er stödjer olika dataformat men json och xml är de vanligaste.  
Det här möjliggör att vi kan använda oss av de resurser (data) som finns på api:et oberoende av vilket programmeringsspråk vi utvecklar med och oberoende av vilket språk api:et är byggt med. ***URI*** står för Uniform Resource Identifier och används för att göra förfrågan till API-bibliotek.



***HTTP request methods,*** även kallat *HTTP-verb,* anger vilken action man vill utföra med sin förfrågan. De fem vanligaste visas i tabellen nedan. Om vi till exempel vill hämta data så gör man ett GET request. Det enda ni behöver använda er av i den här övningen är GET!

|  |  |
| --- | --- |
| HTTP-VERB | CRUD |
| POST | Create |
| GET | Read |
| PUT | Update/Replace (full update) |
| PATCH | Update/Modify (partiell update) |
| DELETE | Delete |

***Request/response body*** är datat som skickas, oftast i JSON format.

{  
 "object": {  
 "a": "b",  
 "c": "d",  
 "e": "f"  
 },  
 "array": [  
 1,  
 2  
 ],  
 "string": "Hello World"  
}

***Postman*** är ett program som kan användas till att anropa och testa api:er när man utvecklar.  
https://www.postman.com/downloads/  
**Uppgift 1: Star Wars Biometrics**The Star Wars API innehåller mängder av data gällande karaktärer, arter, rymdskepp, planeter osv från Star Wars-filmerna. Dokumentation: <https://www.swapi.tech/documentation> Man kan till exempel lista alla karaktärer genom att göra en GET-request:   
https://www.swapi.tech/api/people  
Man kan också göra en GET-request med en query string för att få en specifik karaktär:  
https://www.swapi.tech/api/people/?name=chewbacca  
Er uppgift är att bygga en sida där användaren skriver in namnet på en Star Wars-karaktär och får tillbaka biometrisk data. Några exempelnamn är Luke Skywalker, R2-D2, Darth Vader, Leia Organa, Chewbacca.

1. Skapa en textruta för att ta mot namnet. Använd <input></input> och ge den attributen type=”text”.
2. Skapa en knapp med <button></button>.
3. Skapa en textruta för output med <textarea></textarea> och ge den minst 5 fem rader med rows-attributet.
4. Skriv en funktion i js-filen som:
   1. läser av värdet av input-textrutan och lägger till den till strängen ”https://www.swapi.tech/api/people/?name=” så vi får en fullständig URI,
   2. använder URI:n som parameter till en fetch-request,
   3. och skriver ut informationen till output-textrutan.  
      function getApi(){  
       /\*Skriv din kod här\*/  
       fetch(fullUri)  
       .then(res => res.json())  
       .then(data =>  
       {

/\*Och här\*/  
 })  
 .catch(err => console.log(err))  
}

data är ett JSON-objekt som innehåller all hämtad information om karaktären. Kolla i dokumentationen eller skriv ut med console.log för att se hur objektet är uppbyggt, Plocka ut height, mass, gender och hair\_color och använd dem för att bygga en sträng med hjälp av template literals (bakåtvända fnuttar). Exempel:

let s = `Value 1: ${value1}, Value 2: ${value2}`;

1. Lägg till en eventListener som lyssnar på knappen och kör ovanstående funktion vid klick.

**Uppgift 2: Dra ett kort**

Deck of Cards API låter oss dra kort från en eller flera kortlekar. I response body så får man både vilket värde kortet har (suit och value) samt en url till en bild på det dragna kortet. Dokumentation:  
http://deckofcardsapi.com/  
Ni ska skriva en sida som visar ett kort i taget när användaren klickar på en knapp.

1. Skapa en <button></button>.
2. Skapa en <div></div> där bilden på kortet ska skivas ut.
3. Skriv en liknande js-fil som i föregående uppgift med vissa skillnader:
   1. Använd denna URI för fetchen:  
      <https://deckofcardsapi.com/api/deck/new/draw/?count=1>
   2. Kolla i dokumentationen för att se hur du ska plocka ut url:en till bilden på kortet från data.
   3. Använd createElement, setAttribute, innerHTML och appendChild för att skapa ett image-element, sätta dess src-attribut, nollställa div-en och lägga till image-elementet till den.
4. Lägg till en eventListener som lyssnar på knappen och kör ovanstående funktion vid klick.

**Extra**

Bygg ett kortspel :) Tag hjälp av dokumentationen för apiet. Ni väljer själv vad ni bygger. Ett enkelt Blackjack spel kanske?