




JSON

KOM IHÅG DOKUMENT

```
{  
  "id": 1,  
  "name": "John Doe",  
  "age": 23,  
  "city": "Philadelphia"  
}
```



JSON

ÄR ETT OBJEKT MEN EJ ETT
RENT JS OBJEKT.
ALLA NYCKLAR HAR CITAT
TECKEN

KONVERTERA

Använd `stringify()` när ni konverterar från ett JS objekt till JSON

Använd `parse()` för att konvertera JSON till JS Objekt



FETCH

KOM IHÅG DOKUMENT

```
fetch('din-api-endpoint', {
  method: 'GET',
  headers: {
    'Authorization': 'Bearer din-autentiserings-token' // Om autentisering krävs
  }
})
.then(response => {
  if (response.ok) {
    return response.json(); // Parsa och använd svaret här
  } else {
    throw new Error('Något gick fel');
  }
})
.then(data => {
  // Hantera data från GET-förfrågan här
})
.catch(error => {
  console.error('Fel:', error);
});
```

Fetch

GET

```
const dataToCreate = { key1: 'värde1', key2: 'värde2' };

fetch('din-api-endpoint', {
  method: 'POST',
  headers: {
    'Authorization': 'Bearer din-autentiserings-token', //Ofta har webbservar inloggning
    'Content-Type': 'application/json' // Ange rätt content-type för POST
  },
  body: JSON.stringify(dataToCreate) // Konvertera data till JSON och skicka som body (kroppen)
})
.then(response => {
  if (response.ok) {
    return response.json();
  } else {
    throw new Error('Något gick fel');
  }
})
.then(data => {
  // Hantera svaret från POST-förfrågan här
})
.catch(error => {
  console.error('Fel:', error);
});
```

Fetch

POST (SKAPA NYTT, ID
BRUKAR I REGEL SKAPAS
AUTOMATISKT I
DATABASEN)


```
889 const resourceId = 123; // Till exempel ID för den resurs du vill uppdatera
890 const updatedData = { key1: 'nytt-värde1', key2: 'nytt-värde2' };
891
892 //Glöm ej id
893 fetch(`din-api-endpoint/${resourceId}`, {
894   method: 'PUT',
895   headers: {
896     'Authorization': 'Bearer din-autentiserings-token',
897     'Content-Type': 'application/json'
898   },
899   body: JSON.stringify(updatedData)
900 })
901   .then(response => {
902     if (response.ok) {
903       return response.json();
904     } else {
905       throw new Error('Något gick fel');
906     }
907   })
908   .then(data => {
909     // Hantera svaret från PUT-förfrågan här
910   })
911   .catch(error => {
912     console.error('Fel:', error);
913   });
```

Fetch

PUT (ÄNDRA, GLÖM EJ ATT
HA MED ID I URL
ADRESSEN, VILKEN VILL VI
ÄNDRA PÅ?)

```
const resourceId = 123; // Resursens ID

fetch(`din-api-endpoint/${resourceId}`, {
  method: 'DELETE',
  headers: {
    'Authorization': 'Bearer din-autentiserings-token' // Om autentisering krävs
  }
})
  .then(response => {
    if (response.ok) {
      // Rätt borttagning
      console.log('Resurs borttagen');
    } else {
      // Felhantering om något går fel
      throw new Error('Något gick fel');
    }
  })
  .catch(error => {
    console.error('Fel:', error);
  });
```

Fetch

DELETE, TA BORT (GLÖM EJ ID I ADRESSEN)

OBS: SE DET SOM EN PRAXIS ATT HA MED HEADERS I FETCH DÅ MAN IBLAND BEHÖVER HA MED INLOGGNINGSUPPGIFTER MEN I EN FETCH MED DELETE ÄR DET EJ NÖDVÄNDIGT OM SERVERN EJ HAR INLOGGNINGSKRAV (DOCK VANLIGT)
CONTENT TYPE ÄR INTE ALLS NÖDVÄNDIGT DÅ VI EJ SKICKAR NY INFORMATION