

**Webbutveckling med PHP**

# **API**

**Utbildare: Mikael Olsson**

[mikael.olsson@emmio.se](mailto:mikael.olsson@emmio.se)

076-174 90 43

# **NACKADEMIN**

# API

- När vi gör ett anrop till en webbsida, vad innehåller svaret?
- Kan man få andra resultat?
  - <https://randomuser.me/api/?results=5>
- Application Programming Interface
- Låter oss hämta det data vi behöver i ett format som vi kan bearbeta, vanligen JSON eller XML.

# JSON

- JSON (JavaScript Object Notation), är ett kompakt, textbaserat format som används för att utbyta data.
- Ni minns detta sedan tidigare.

```
{  
    "firstName": "Jason",  
    "lastName": "Smith",  
    "age": 25,  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": "10021"  
    },  
    "phoneNumber": [  
        { "type": "home", "number": "212 555-1234" },  
        { "type": "fax", "number": "646 555-4567" }  
    ],  
    "newSubscription": false,  
    "companyName": null  
}  
  
• var obj = JSON.parse(text);
```

# API Consumer

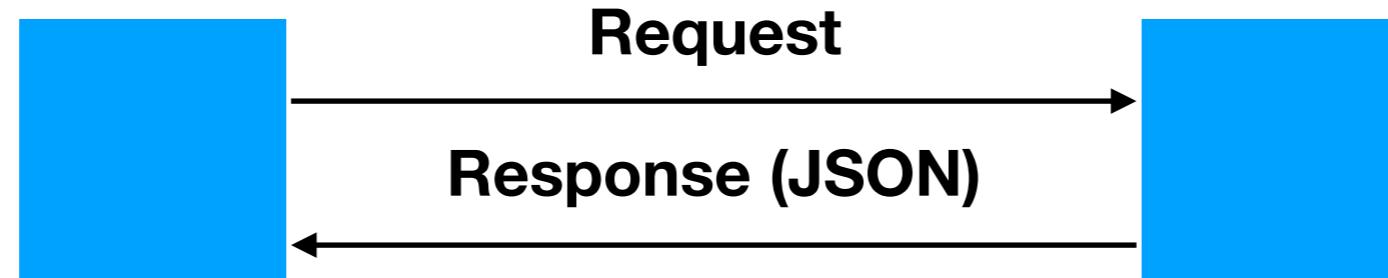
- En applikation som hämtar data från ett API brukar kallas *consumer*.
- Tidigare gjorde vi en consumer som hämtade data från <https://randomuser.me/api/>

```
1  $("button").click(function() {  
2    $.ajax({  
3      url: 'https://randomuser.me/api/?results=5',  
4      success: function(data) {  
5        console.log(data["results"][2]["name"]["first"]);  
6      }  
7    });  
8  });  
9
```

Vår applikation

```
{"results": [ {"gender": "male", "name": {"title": "93a9-",  
"latitude": "65.3817", "longitude": "-169.5081"},  
"c872a8aff727", "username": "redmeercat784", "pass-  
d8a6aa564e2bbf020c8835a0a50c87"}, {"dob": {"date":  
"name": "", "value": null}, "picture": {"large": "https://randomuser.me/api/portraits/  
"seed": "ae5ecc018f20af63", "results": 1, "page": 1}
```

<https://randomuser.me/api/>



# API Consumer

- Vi kan hämta data från API:er med PHP också.
- Olika metoder fungerar på olika servrar, beroende på vad som är tillåtet i PHP-inställningarna och vad som finns installerat på servern.
  - Enklast är kanske `file_get_contents`.
  - Andra alternativ: `curl`, `fopen`
  - <https://oscarliang.com/six-ways-retrieving-webpage-content-php/>

# API Consumer

- Uppgift: Skriv en applikation som hämtar fem personer och skriver ut utvalda data i en tabell.

```
$url = 'https://randomuser.me/api/' ;
$file_content = file_get_contents($url) ;
var_dump($file_content) ;
```

# API Consumer

- Vi skulle även kunna hämta data från ett API och spara i vår egen databas.
  - Ibland är datat vi får strukturerat på ett sätt som gör det olämpligt för våra syften. Då kan vi strukturera om det och spara det.
  - Vi kan *mellanlägra* data för att inte överbelasta API-servern.
  - Vi får bättre performance om vi kan hämta data från vår egen databas.

# API Provider

- En server som tillhandahåller ett API kallas API Provider.
- Vi kan börja med att göra en applikation som listar alla produkter i en html-tabell, precis som tidigare.
- Vi utgår från classic models. Få filen att funka:

14-api/products/list-html.php

# API Provider

- Nu har vi gjort en fil som skapar html. Låt oss se om vi kan ta nästan samma fil och få den att producera ett lämpligt JSON-svar.

14-api/products/read.php

# Testa

- Hur testar vi att anropet fungerar som det ska?
- Kan vi modifiera filen vi skapade förut? (consumern)

# HTTP-verb

- Hittills har vi alltid använt GET i våra API-anrop. Vilka andra finns?
- Dessa är exempel, det finns ännu fler.

| HTTP verb | CRUD               |
|-----------|--------------------|
| POST      | Create             |
| GET       | Read               |
| PUT       | Update/<br>Replace |
| DELETE    | Delete             |

# POST

- Vi borde kunna fixa en sida som tar hand om POST-requests och skapar en ny produkt.

14-api/products/write.php

# Testa

- Hur testar vi att posta data?
- Vår förra fil gör ett GET-request. Vi behöver göra ett POST-request.
- Det finns olika sätt, vi kan t ex använda Postman, men för programmeringssyften kan vi använda Curl.

15-curl/02-post.php

- <https://www.madebymagnitude.com/blog/sending-post-data-from-php/>

# Endpoints

- De URL:er vi har skapat nu skulle kunna utgöra *endpoints*.
- En endpoint är i princip en punkt man kan ansluta till i ett API.

# Hantera data med JS

- Nu när ni har börjat förstå hur API:er fungerar förstår ni kanske också lite mer om hur man kan använda JS för att hantera data?
- Istället för att spara data i en cookie, textarea eller liknande kan man ansluta till ett API och hämta/spara data den vägen.

# Uppgift

- Skapa endpoints/script för att uppdatera data och ta bort data.
- I båda fallen måste man specificera vilken produkt man vill ta bort/uppdatera. Det kan vi t ex göra genom querystring: update.php?id=123\_1234

# Uppdatera

- Extra uppgift:
  - Skapa en sida med ett formulär som innehåller produktdata.
  - Låt användaren uppdatera data med JS/Ajax istället för att ladda om sidan med PHP.

# Autentisera

- Anledningar:
  - Skydda data
  - Skydda servern från överbelastning
  - Statistik

# Basic authentication

- Metod för att låta en HTTP-användare skicka namn och lösenord i ett request.
- Klienten autentiseras genom en header
  - Authorization: Basic <credentials>
  - Credentials är en base64-kodad sträng bestående av användare:lösenord.
- En användare kan autentisera sig genom URL:en:  
<https://Aladdin:OpenSesame@www.example.com/>

# Basic authentication - ex

- 16-authentication/01-basic/

# API-key

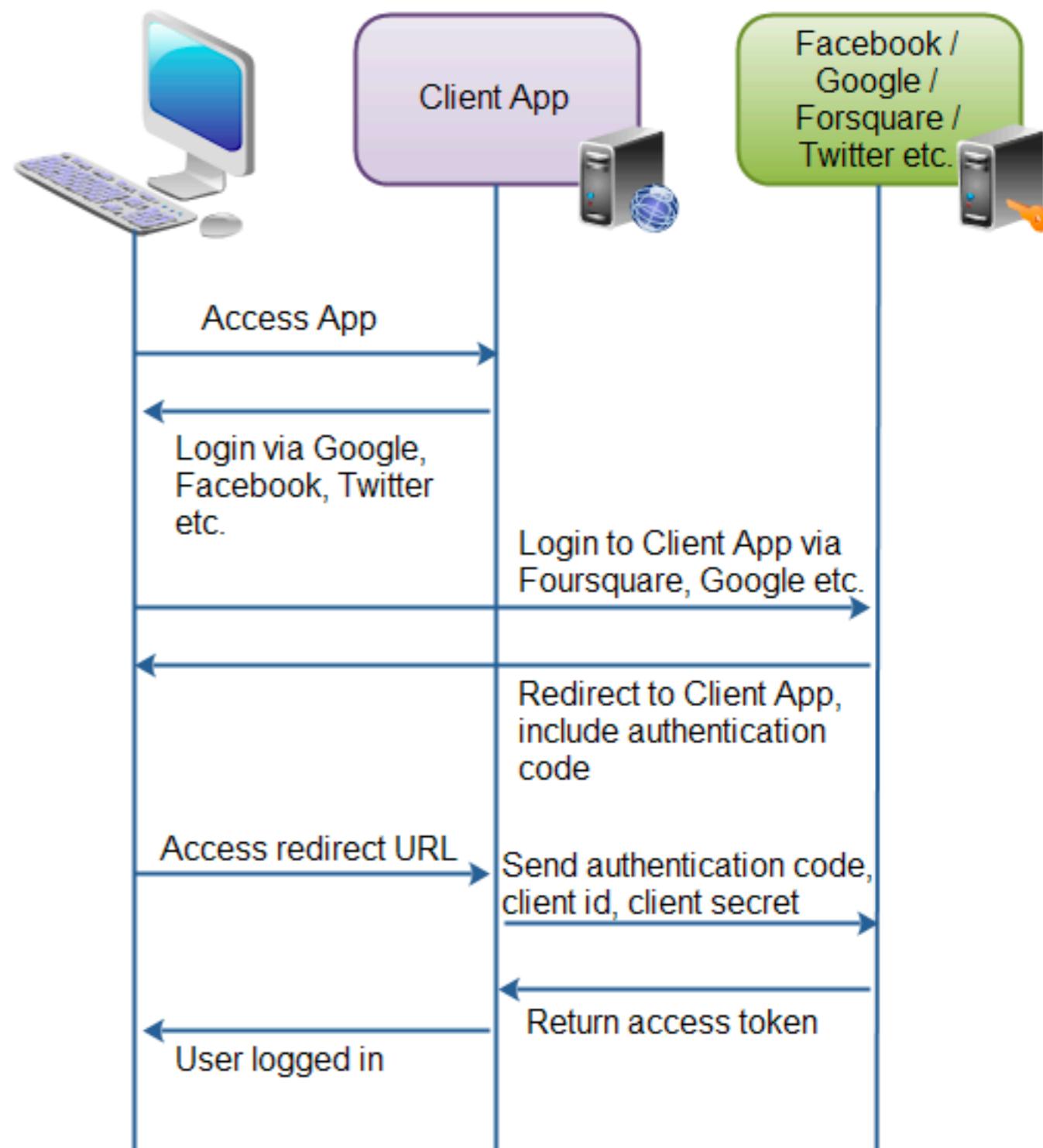
- Ganska simpelt och vanligt sätt att hantera åtkomst.
- Varje användare / konto får en nyckel som används i varje request.

# API-key - ex

- 16-authentication/02-api-key
- Validate phonenumber with <https://numvalidate.com/>

# Oauth2

- Ett protokoll för att låta applikationer komma åt varandras data.



# Oauth2

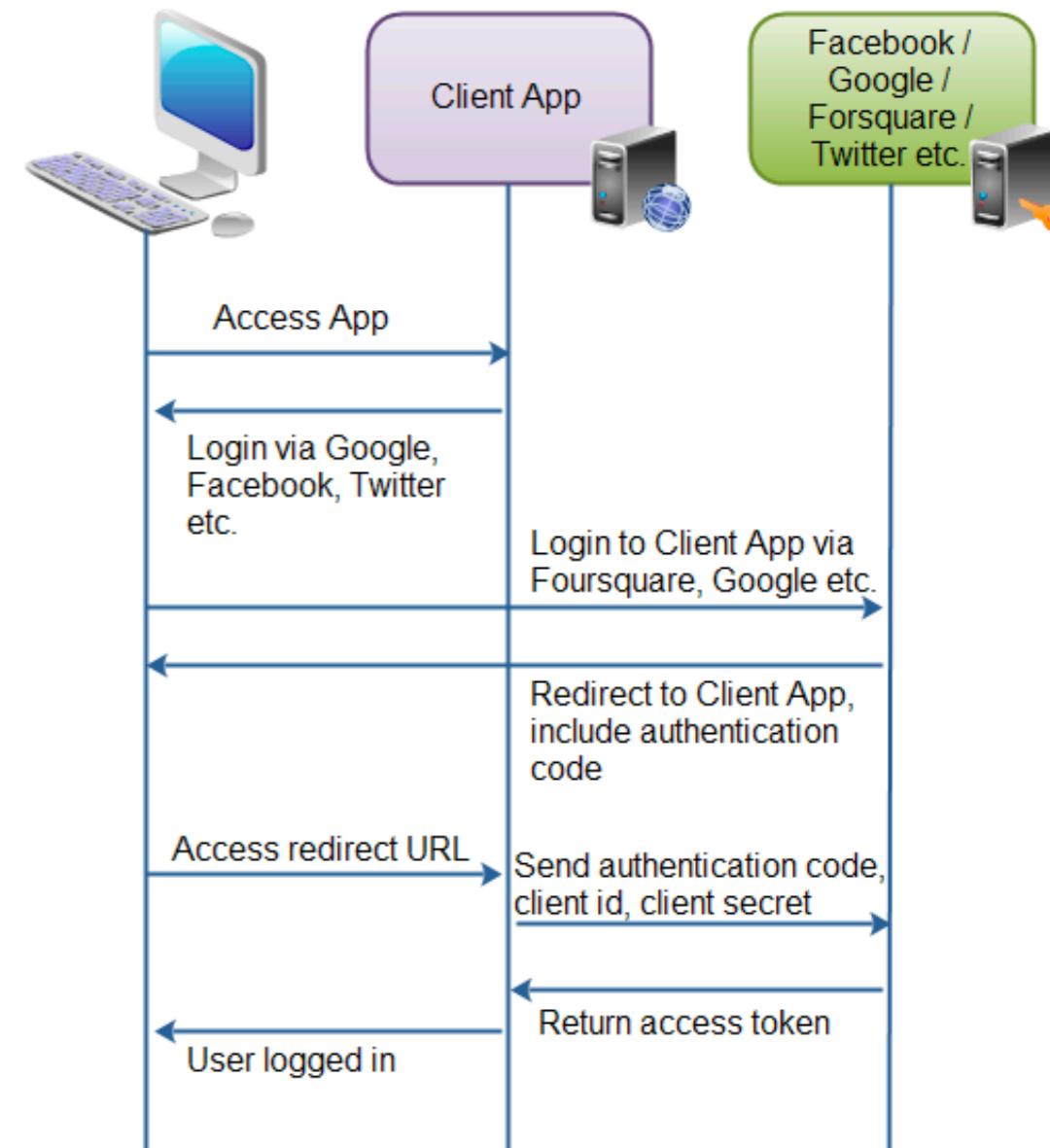
1. Användaren går till **klientapplikationen**, som har en knapp typ "logga in via Facebook/Google/Twitter".

2. Användaren klickar och hamnar på den **autentiserande applikationen** (dvs FB etc). Användaren loggar in på denna plattform och får frågan om hen vill acceptera att **klienten** får access till vissa av de data som finns på den **autentiserande applikationen**.

3. Den autentiserande applikationen skickar användaren vidare till en URI som **klientapplikationen** har specat. Att tillhandahålla denna görs normalt av **klienten** när den registrerar sig hos **autentiseraren**. Då får även **klienten** sitt *client id* och *client password*. **Autentiseraren** lägger till en auktoriseringskod till URL:en.

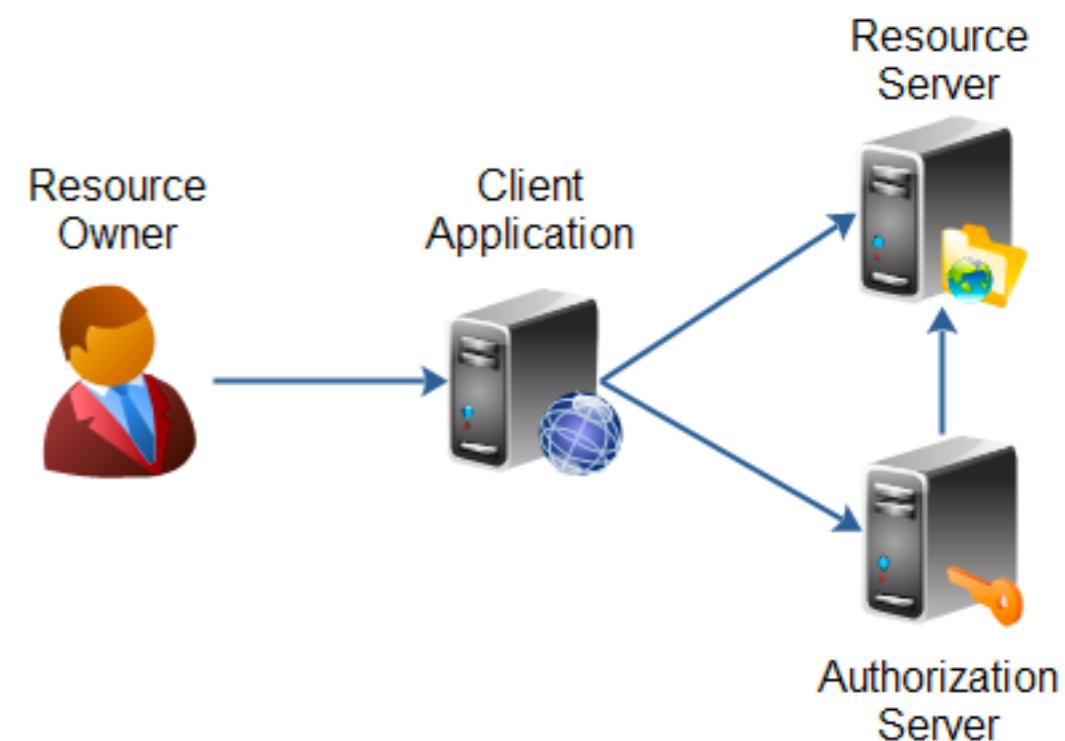
4. Användaren kommer till redirect-sidan specad av **klienten**. I bakgrunden kontaktar **klienten** **autentiseraren** och skickar client id, client password och auktoriseringskoden. **Autentiseraren** skickar tillbaka ett access token.

5. När **klienten** har fått detta access token kan detta användas för att anropa resurser hos FB/Google/Twitter etc.

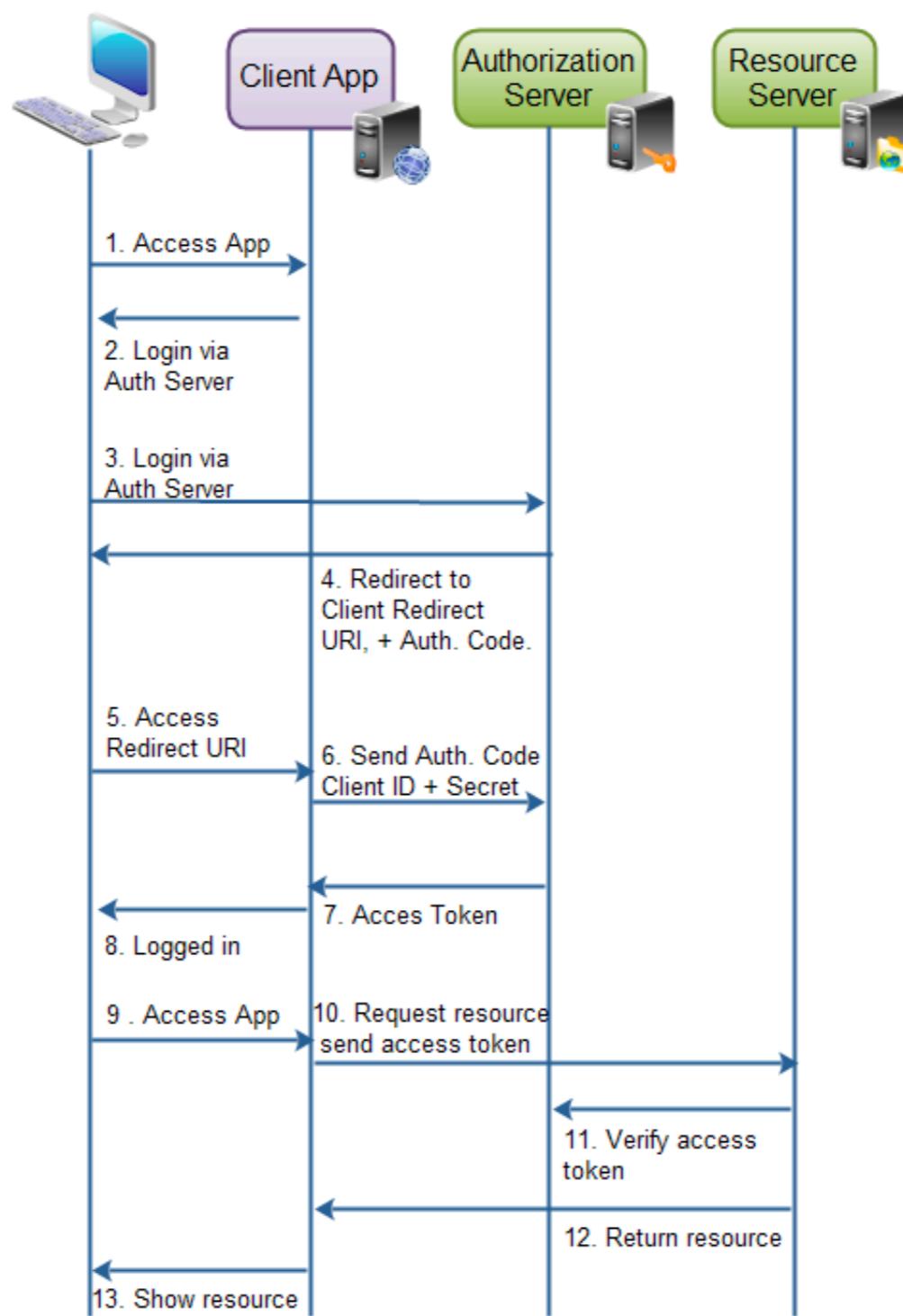


# Oauth2

- Resursägaren är den person eller resurs som äger datat. En användare på FB kan t ex vara en resursägare. Resursen är oftast användarens data.
- Resursservern är servern som har datat, t ex FB.
- Klienten är applikationen som vill ha åtkomst till data på resursservern. Kan t ex vara ett spel som vill ha åtkomst till användarens FB-konto.
- Autentiseringsservern är servern som behandlar åtkomst. Kan vara samma som resursservern, men måste inte vara.



# Oauth2 - Autentisering



# Oauth2 - Endpoints

- Authorization Endpoint

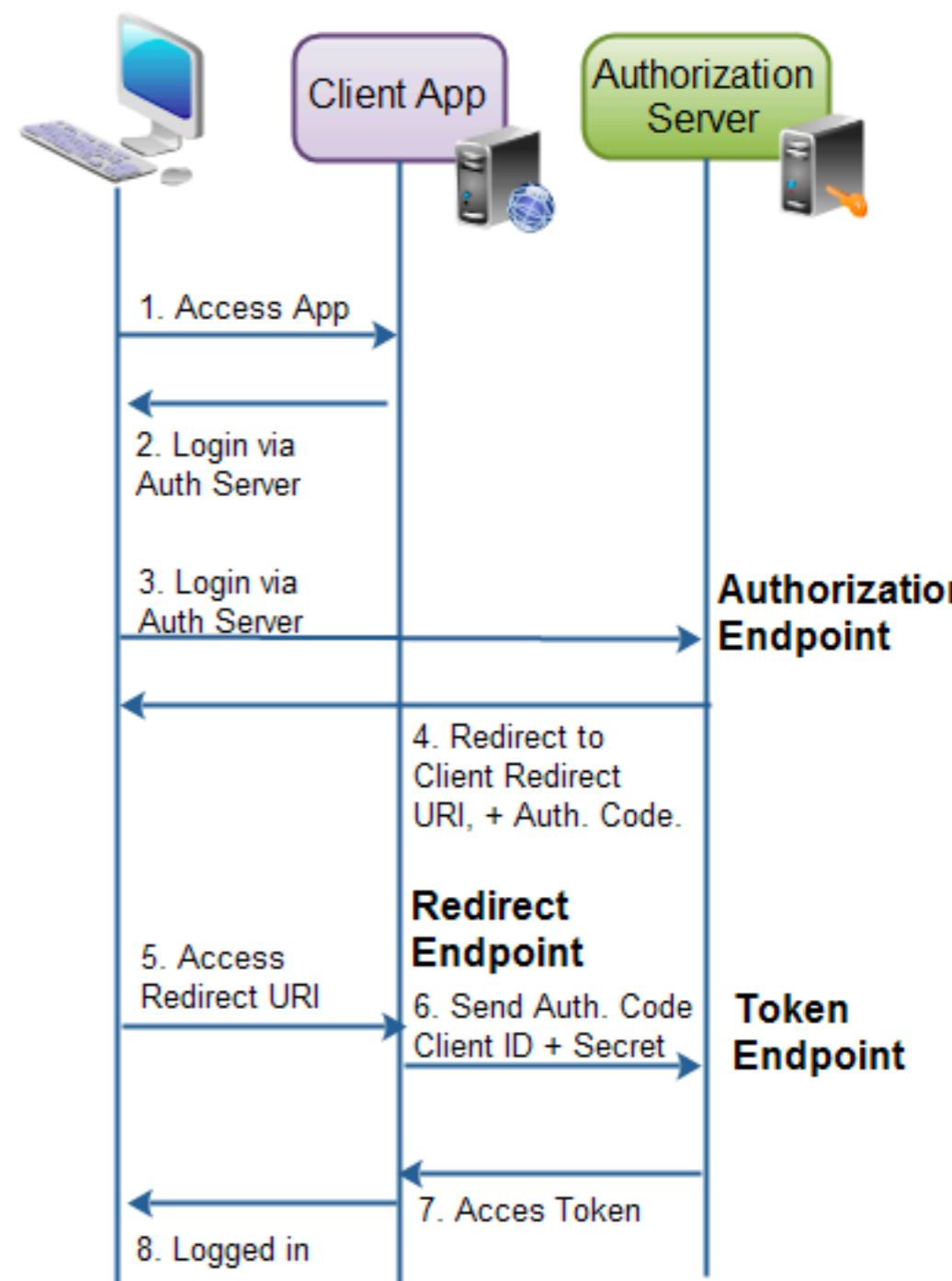
- Där på autentiseraren som resursägaren loggar in och godkänner (grants) autentisering till klienten.

- Token Endpoint

- Där på autentiserare som klienten byter autentiseringskoden, client id och client password mot en access token.

- Redirect Endpoint

- Där på klienten som resursägaren skickas vidare till efter att ha fått (granted) access.



# curl

- Curl är ett program som finns inbyggt i de flesta macar och Linux-datorer.
- Finns även till Windows. Lättast via git bash.
- Används för att föra över data med url:er.
- <https://curl.haxx.se/>
- PHP har stöd för att använda curl, mer om det senare.

# curl --help

- Curl har en massa options.

```
→ curl curl --help
Usage: curl [options...] <url>
Options: (H) means HTTP/HTTPS only, (F) means FTP only
  --anyauth      Pick "any" authentication method (H)
  -a, --append    Append to target file when uploading (F/SFTP)
  --basic        Use HTTP Basic Authentication (H)
  --cacert FILE  CA certificate to verify peer against (SSL)
  --capath DIR   CA directory to verify peer against (SSL)
  -E, --cert CERT[:PASSWD] Client certificate file and password (SSL)
  --cert-status  Verify the status of the server certificate (SSL)
  --cert-type TYPE Certificate file type (DER/PEM/ENG) (SSL)
  --ciphers LIST SSL ciphers to use (SSL)
  --compressed   Request compressed response (using deflate or gzip)
  -K, --config FILE Read config from FILE
  --connect-timeout SECONDS Maximum time allowed for connection
  --connect-to HOST1:PORT1:HOST2:PORT2 Connect to host (network level)
  -C, --continue-at OFFSET Resumed transfer OFFSET
  -b, --cookie STRING/FILE Read cookies from STRING/FILE (H)
  -c, --cookie-jar FILE Write cookies to FILE after operation (H)
  --create-dirs   Create necessary local directory hierarchy
  --crlf         Convert LF to CRLF in upload
  --crlfile FILE Get a CRL list in PEM format from the given file
  -d, --data DATA  HTTP POST data (H)
  --data-raw DATA  HTTP POST data, '@' allowed (H)
  --data-ascii DATA  HTTP POST ASCII data (H)
  --data-binary DATA  HTTP POST binary data (H)
  --data-urlencode DATA  HTTP POST data url encoded (H)
  --delegation STRING GSS-API delegation permission
  --digest        Use HTTP Digest Authentication (H)
  --disable-eprt Inhibit using EPRT or LPRT (F)
  --disable-epsv Inhibit using EPSV (F)
  --dns-servers  DNS server addrs to use: 1.1.1.1;2.2.2.2
  --dns-interface Interface to use for DNS requests
  --dns-ipv4-addr TPy4 address to use for DNS requests, dot notation
```

# Fejk-API

- Det finns många siter som agerar som API:er men som inte låter en exempelvis skapa poster på riktigt.
- Kan användas för tester.
- <http://jsonplaceholder.typicode.com/>

## Routes

All HTTP methods are supported.

|        |                                    |
|--------|------------------------------------|
| GET    | <a href="#">/posts</a>             |
| GET    | <a href="#">/posts/1</a>           |
| GET    | <a href="#">/posts/1/comments</a>  |
| GET    | <a href="#">/comments?postId=1</a> |
| GET    | <a href="#">/posts?userId=1</a>    |
| POST   | <a href="#">/posts</a>             |
| PUT    | <a href="#">/posts/1</a>           |
| PATCH  | <a href="#">/posts/1</a>           |
| DELETE | <a href="#">/posts/1</a>           |

# Verb

- Vad kan jag göra med nedanstående endpoint?
- <http://jsonplaceholder.typicode.com/>  
[posts](#)

## Routes

All HTTP methods are supported.

|        |                                    |
|--------|------------------------------------|
| GET    | <a href="#">/posts</a>             |
| GET    | <a href="#">/posts/1</a>           |
| GET    | <a href="#">/posts/1/comments</a>  |
| GET    | <a href="#">/comments?postId=1</a> |
| GET    | <a href="#">/posts?userId=1</a>    |
| POST   | <a href="#">/posts</a>             |
| PUT    | <a href="#">/posts/1</a>           |
| PATCH  | <a href="#">/posts/1</a>           |
| DELETE | <a href="#">/posts/1</a>           |

# GET /posts

```
[  
 {  
   "userId": 1,  
   "id": 1,  
   "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
   "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quia  
,  
 {  
   "userId": 1,  
   "id": 2,  
   "title": "qui est esse",  
   "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\\nfugiat blanditiis nisi nulla"  
,  
 {  
   "userId": 1,  
   "id": 3,  
   "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",  
   "body": "et iusto sed quo iure\\nvoluptatem occaecati omnis eligendi aut ad\\nvoluptatem doloribus vel accusant:  
,  
 {  
   "userId": 1,  
   "id": 4,  
   "title": "eum et est occaecati",  
   "body": "ullam et saepe reiciendis voluptatem adipisci\\nsit amet autem assumenda provident rerum culpa\\nquis i  
,  
 {  
   "userId": 1,  
   "id": 5,  
   "title": "nesciunt quas odio",  
   "body": "repudiandae veniam quaerat sunt sed\\nalias aut fugiat sit autem sed est\\nvoluptatem omnis possimus e
```

# Hämta med curl

- Med curl kan vi hämta vilken site som helst.

```
➔ curl curl https://www.nt.se/nyheter/
<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie10 lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie10 lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>        <html class="no-js lt-ie10 lt-ie9"> <![endif]-->
<!--[if IE 9]>        <html class="no-js lt-ie10"> <![endif]-->
<!--[if gt IE 9]><!-->
<html class="no-js">
<!--<![endif]-->
<head><meta charset="utf-8" /><meta http-equiv="X-UA-Compatible" content="IE=edge" />

<script type="text/javascript" src="/public/js/cookie.min.js"></script>
<script type="text/javascript" src="/public/js/ads.js"></script>
<script type="text/javascript" src="https://storage.googleapis.com/glimr-static/glimrsdk-js/>
<script type="text/javascript">
  if(typeof (Glimr) !== 'undefined' && Glimr !== null) {
    Glimr.setTagCacheTimeInSeconds(300);
    var glimrTags = Glimr.getCachedBehaviorTagsAndUpdateInBackground('NT02FP00PL
  }
</script>

<script type="text/javascript">
```

# Hämta med curl

- Vi kan inkludera headers.

```
→ curl curl -i https://www.nt.se/nyheter/
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-Aid: 9443518
X-Pids: 33
X-Metering-Duration: 30
X-Metering-Quota: 1
X-Site-Class: nypon
X-Auth-Failed: https://www.nt.se/inc/eprencallback.aspx?action=denyaccess&redirect=%2fnyheter%2f&meter=False
X-Auth-Denied: https://www.nt.se/inc/eprencallback.aspx?action=denyaccess&redirect=%2fnyheter%2f&meter=False
X-Session-Denied: https://www.nt.se/inc/eprencallback.aspx?action=denysession&redirect=%2fnyheter%2f&meter=False
X-AspNet-Version: 4.0.30319
Date: Thu, 09 May 2019 04:42:37 GMT
cache-control: public, max-age=60
X-Status: 200
X-Length:
X-First-Node: false
X-Pre-TTL: 60.000
X-Backend: nt_live
X-Varnish: 771757234 757868272
Age: 26
Via: 1.1 varnish-plus-v4
X-SN: NRK-WCACHE-001
X-Met-Period-Duration: 30d
X-Quota-Noauth: 1
Transfer-Encoding: chunked
Connection: keep-alive
Accept-Ranges: bytes

<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie10 lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie10 lt-ie9 lt-ie8"> <![endif]-->
```

# Hämta med curl

- Vi kan hämta enbart headers.

```
→ curl curl --head https://www.nt.se/nyheter/
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-Aid: 9443518
X-Pids: 33
X-Metering-Duration: 30
X-Metering-Quota: 1
X-Site-Class: nypon
X-Auth-Failed: https://www.nt.se/inc/eprencallback.aspx?action=denyaccess&redirect=%2fnyheter%2f&meter=False
X-Auth-Denied: https://www.nt.se/inc/eprencallback.aspx?action=denyaccess&redirect=%2fnyheter%2f&meter=False
X-Session-Denied: https://www.nt.se/inc/eprencallback.aspx?action=denysession&redirect=%2fnyheter%2f&meter=False
X-AspNet-Version: 4.0.30319
Date: Thu, 09 May 2019 04:29:26 GMT
cache-control: public, max-age=60
X-Status: 200
X-Length:
X-First-Node: false
X-Pre-TTL: 60.000
X-Backend: nt_live
X-Varnish: 739546870 541747606
Age: 20
Via: 1.1 varnish-plus-v4
X-SN: NRK-WCACHE-001
X-Met-Period-Duration: 30d
X-Quota-Noauth: 1
Connection: keep-alive
```

# Hämta med curl

- Vad är det för typ av innehåll?
- HTTP-status?

```
→ curl curl --head http://jsonplaceholder.typicode.com/posts
HTTP/1.1 200 OK
Date: Thu, 09 May 2019 04:54:21 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Set-Cookie: __cfduid=d231aebbc8f643e248dae90e9c7abdde21557377661; expires=Fri, 08-May-20 04:54:21 GMT; path=/; domain=.typicode.com; HttpOnly
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=14400
Pragma: no-cache
Expires: Thu, 09 May 2019 08:54:21 GMT
X-Content-Type-Options: nosniff
Etag: W/"6b80-Ybsq/K6GwqrYkAsFxqDXGC7DoM"
Via: 1.1 vegur
CF-Cache-Status: HIT
Server: cloudflare
CF-RAY: 4d41132fbcc52d147-GOT
```

# Hämta med curl

- Vad har hänt här?
- Vad har vi fått för status?  
Vad betyder den?

```
→ curl curl --head https://www.nt.se/
HTTP/1.1 302 Found
Cache-Control: public, max-age=3600
Content-Type: text/html; charset=utf-8
Location: /nyheter/
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
Date: Thu, 09 May 2019 06:26:12 GMT
Content-Length: 126
X-Status: 302
X-Length: 126
X-First-Node: false
X-Pre-TTL: 3600.000
X-Backend: nt_live
X-Varnish: 357337218 414663256
Age: 266
Via: 1.1 varnish-plus-v4
X-SN: LNK-WCACHE-002.nt.se
Connection: keep-alive
```

# Hämta med curl

- Vi kan låta curl följa redirects med flaggan -L.

```
→ curl curl --head -L https://www.nt.se/
HTTP/1.1 302 Found
Cache-Control: public, max-age=3600
Content-Type: text/html; charset=utf-8
Location: /nyheter/
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
Date: Thu, 09 May 2019 06:26:12 GMT
Content-Length: 126
X-Status: 302
X-Length: 126
X-First-Node: false
X-Pre-TTL: 3600.000
X-Backend: nt_live
X-Varnish: 392940116 414663256
Age: 274
Via: 1.1 varnish-plus-v4
X-SN: LNK-WCACHE-002.nt.se
Connection: keep-alive

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
X-Aid: 9443518
X-Pids: 33
X-Metering-Duration: 30
X-Metering-Quota: 1
X-Site-Class: nypon
X-Auth-Failed: https://www.nt.se/inc/eprencallback.aspx?action=failed&id=9443518
X-Auth-Denied: https://www.nt.se/inc/eprencallback.aspx?action=denied&id=9443518
X-Session-Denied: https://www.nt.se/inc/eprencallback.aspx?action=sessiondenied&id=9443518
X-AspNet-Version: 4.0.30319
Date: Thu, 09 May 2019 06:30:24 GMT
cache-control: public, max-age=60
```

# Hämta med curl

- Vi kan hämta innehållet till en fil.

```
→ curl curl -o nt.html https://www.nt.se/nyheter/
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  453k     0  453k      0       0  151k          0 --:--:-- 0:00:02 --:--:-- 151k
→ curl ls -la
total 1024
drwxr-xr-x  3 micke  staff   96  9 Maj 06:31 .
drwxr-xr-x  6 micke  staff  192  8 Maj 18:44 ..
-rw-r--r--  1 micke  staff 463945  9 Maj 06:31 nt.html
```

# Hämta med curl

- Vi kan hämta filer.

```
→ curl curl -O https://imgs.xkcd.com/comics/standards.png
% Total    % Received % Xferd  Average Speed   Time   Time    Time  Current
                                         Dload  Upload   Total Spent  Left  Speed
100 24305  100 24305    0      0  43235      0 --:--:-- --:--:-- --:--:-- 43170
→ curl ls -la
total 1072
drwxr-xr-x  4 micke  staff   128  9 Maj 07:02 .
drwxr-xr-x  6 micke  staff   192  8 Maj 18:44 ..
-rw-r--r--  1 micke  staff  463945  9 Maj 06:31 nt.html
-rw-r--r--  1 micke  staff  24305  9 Maj 07:02 standards.png
→ curl █
```

# Hämta med curl

- Vi kan hämta olika typer av innehåll, inte bara HTML.

```
➔ curl curl http://jsonplaceholder.typicode.com/posts
[{"id": 1, "userId": 1, "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae eveniet architecto"}, {"id": 2, "userId": 1, "title": "qui est esse", "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\\nfuga ut reiciendis\\nqui aperiam non debitis possimus qui neque nisi nulla"}, {"id": 3, "userId": 1, "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut", "body": "et iusto sed quo iure\\nvoluptatem occaecati omnis eligendi aut ad\\nvoluptatem doloribus v\\s odio et labore et velit aut"}, {"id": 4, "userId": 1, "title": "et harum quidem sunt ut labore et dolore magnam aliquam quaerat voluptatem", "body": "ut enim ad minima veniam quis nostrum eiusmodi temporibus culpa officia deserunt mollitia animi ut labore et dolore magnam aliquam quaerat voluptatem"}]
```

# Andra requests med curl

- Hittills har vi bara gjort GET-requests, men vi kan såklart göra andra typer av requests.
- Nu kommer raderna att bli längre, men vi kan skicka in kommandon med radbrytningar för läsbarhet med hjälp av backslash-tecknet.

```
→ curl curl \  
  > -H "Content-Type: application/json" \  
  > -d '{"title":"Hello","body":"World"}' \  
  > -i \  
  > http://jsonplaceholder.typicode.com/posts
```

# Andra requests med curl

- `-H` lägger till en header till requestet. I det här fallet vill vi berätta att vårt request innehåller JSON och inte application/x-www-form-urlencoded (**formulär**), som annars är standard för `-d`.
- `-X` är vilket verb vi vill skicka.
- `-d` betyder att parametern är det data vi vill skicka med requestet. Om vi hade velat skicka "formulär" hade vår data sett ut såhär:  
`-d "title=Hello&body=World"`
- `-i` betyder inkludera headers i svaret.

```
→ curl curl \
> -H "Content-Type: application/json" \
> -X POST \
> -d '{"title": "Hello", "body": "World"}' \
> -i \
> http://jsonplaceholder.typicode.com/posts
```

# Andra requests med curl

- Vad fick vi tillbaka för status?
- Vad fick vi tillbaka för data?

```
→ curl curl \
> -H "Content-Type: application/json" \
> -X POST \
> -d '{"title":"Hello","body":"World"}' \
> -i \
> http://jsonplaceholder.typicode.com/posts
HTTP/1.1 201 Created
Date: Thu, 09 May 2019 05:32:42 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 54
Connection: keep-alive
Set-Cookie: __cfduid=d2ecfbe6b4df444c096f67d0706f0c8631557379962; expires=Fri, 08-May-2020 05:32:42 GMT; path=/; domain=.jsonplaceholder.typicode.com; HttpOnly; SameSite=Lax
X-Powered-By: Express
Vary: Origin, X-HTTP-Method-Override, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Access-Control-Expose-Headers: Location
Location: http://jsonplaceholder.typicode.com/posts/101
X-Content-Type-Options: nosniff
Etag: W/"36-75/hoVNRAuw1Yo+G++GCvGp5pB0"
Via: 1.1 vegur
Server: cloudflare
CF-RAY: 4d414b5b7bd4d147-GOT

{
  "title": "Hello",
  "body": "World",
  "id": 101
}%
→ curl
```

# Hämta post

- Vad gör vi i detta request?

```
→ curl curl http://jsonplaceholder.typicode.com/posts/3
{
  "userId": 1,
  "id": 3,
  "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
  "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem dolor
odio et labore et velit aut"
}%
→ curl |
```

# Andra requests med curl

- Vad gör vi i detta request?
- Hur vet API:et vilken post som ska uppdateras?

```
→ curl curl \
-H "Content-Type: application/json" \
-X PUT \
-d '{"title":"Hello","body":"World"}' \
-i \
http://jsonplaceholder.typicode.com/posts/3
HTTP/1.1 200 OK
Date: Thu, 09 May 2019 06:19:15 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 52
Connection: keep-alive
Set-Cookie: __cfduid=d5209608e8147d84865b795c321832efc1557382755; ex
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Content-Type-Options: nosniff
Etag: W/"34-wGoUuk620TZFzwsPi/1w+0Weaec"
Via: 1.1 vegur
Server: cloudflare
CF-RAY: 4d418f8daf0d13f-GOT

{
  "title": "Hello",
  "body": "World",
  "id": 3
}%
```

# Andra requests med curl

- Vi kan såklart även ta bort poster.

```
➔ curl curl \
-X DELETE \
-i \
http://jsonplaceholder.typicode.com/posts/3
HTTP/1.1 200 OK
Date: Thu, 09 May 2019 06:26:52 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 2
Connection: keep-alive
Set-Cookie: __cfduid=d3a558380759bd0dccf6083681f
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Content-Type-Options: nosniff
Etag: W/"2-vyGp6PvFo4RvsFtPoIWeCReyIC8"
Via: 1.1 vegur
Server: cloudflare
CF-RAY: 4d419ab62cc1d143-GOT
```

# PHP & curl

- De flesta installationer av PHP har tillgång till curl.
- Man kan kolla med `phpinfo();`

curl

|                  |   |
|------------------|---|
| cURL support     | enabled   |
| cURL Information | 7.60.0  |
| Age              | 4   |
| Features         |   |
| AsynchDNS        | Yes   |
| CharConv         | No  |
| Debug            | No  |
| GSS-Negotiate    | No  |
| IDN              | No  |
| IPv6             | Yes   |
| krb4             | No  |
| Largefile        | Yes   |
| libz             | Yes   |
| NTLM             | Yes   |
| NTLMWB           | Yes   |
| SPNEGO           | No  |
| SSL              | Yes   |
| SSPI             | No  |
| TLS-SRP          | Yes   |
| HTTP2            | Yes   |
| GSSAPI           | No  |
| KERBEROS5        | No  |
| UNIX_SOCKETS     | Yes   |
| PSL              | No  |
| Protocols        | dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, tftp |
| Host             | x86_64-apple-darwin13.4.0   |
| SSL Version      | OpenSSL/1.0.2o  |
| ZLib Version     | 1.2.8   |

# PHP exempel

- 15-curl/01-get.php
- 15-curl/02-post.php
- 15-curl/03-put.php
- 15-curl/04-delete.php

# Postman

- Utveckling av API:er består av flera delar.
- Postman låter oss fokusera på en del i taget. När vi vet vilket data vi kan få ut och hur kan vi fokusera på att se till att vår applikation tar hand om data på rätt sätt.

# Exempel

- Hämta växlingskurser mot den svenska kronan.
  - <https://exchangeratesapi.io/>
  - Se först till att hämta rätt resultat med Postman
  - Gör därefter en enkel applikation som hämtar ditt data och skriver ut växlingskursen mot Euro.
  - Använd gärna:  
15-get\_data/01-consume.php

# Fler exempel

- Om du får tid över:
  - Verifiera telefonnr:  
<https://numverify.com/>
  - Hämta ett random Chuck Norris-skämt:  
<http://www.icndb.com/api/>
  - Hämta jobb från Github (PHP/San Fransisco):  
<https://jobs.github.com/api>
- Hitta något annat som låter kul:  
<https://github.com/toddmotto/public-apis>

# Implementerat

- Hittills har vi implementerat:
  - GET (Read)
  - POST (Create)
- För entiteten:
  - Product

# Fler entiteter

- Hur skulle vi bära oss åt om vi ville skapa ett liknande API för customers?
- Kopiera klassen?
- What about redundancy? Finns det saker som är gemensamma för hur klasserna skulle fungera? Kan vi dra nytta av det?
- Hur gjorde vi med Student / Teacher? Vehicle / Car?

# API:ets URI

| Protokoll         | Server | Port   | Sökväg            |
|-------------------|--------|--|-------------------|
| • https://medint: | 8890   | /exercises/03-routing/01-simple-router/index.php | ?product/S10_1679 |
| Fil               |        |  | Querystring       |

# Router

- En router är en bit kod som bestämmer vad som ska hänta när användaren skriver in en viss adress.
- Ex: example.com/about -> include views/about.php;

# Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?