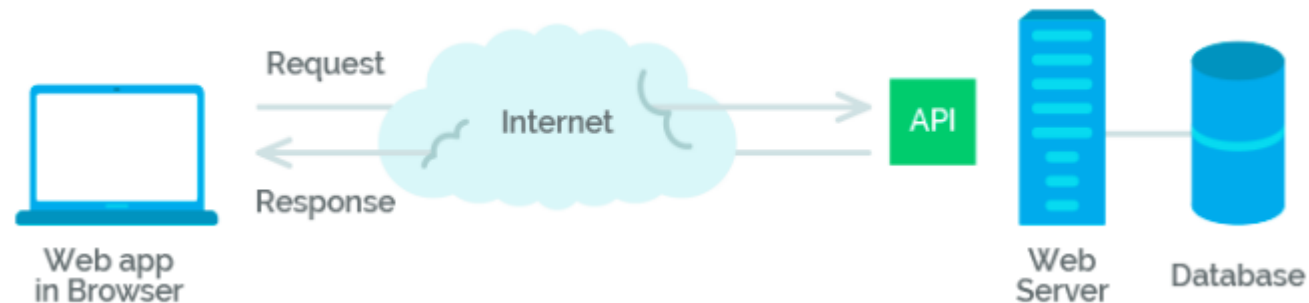


REST API

# API

- Ett API är ett "Application programming interface". Det är en uppsättning regler som gör det möjligt för program att prata med varandra. Utvecklaren skapar API på servern och låter klienten prata med den.



# REST

- REST avgör hur API:et ser ut. Det står för "Representational State Transfer". Det är en uppsättning regler som utvecklare följer när de skapar sitt API
- Varje URL kallas en begäran(**request**) medan data som skickas tillbaka till dig kallas ett svar (**response**).

# Anatomi av en "request"

- Endpoint
- Metod
- Headers
- Data (body)

# Endpoint

- Endpoint:en (route) är URL där du gör en request. Följer denna struktur:

```
root-endpoint/?
```

- Root-endpoint är utgångspunkt från Api:et du begär från.
- Root-endpoint av Github är:

```
https://api.github.com
```

# Endpoint/Path

- För att förstå vilka sökvägar som finns tillgängliga i API: et måste du titta igenom API-dokumentationen.
- Om du vill kolla vilka repositories en specifik användare har genom githubs API. [Dokumentationen](#) säger att vi ska använda den path:

```
/user/:username/repos
```

- Alla kolon (:) på en sökväg betecknar en variabel. Du bör ersätta dessa värden med faktiska värden när du skickar din begäran/request

```
https://api.github.com/users/{dittAnvändarnamn}/repos
```

```
https://api.github.com/users/ dittAnvändarnamn /repos?sort=pushed
```

# Metoder

- GET
- POST
- PUT
- PATCH
- DELETE

CRUD	SQL	HTTP
create	INSERT	PUT
read	SELECT	GET
update	UPDATE	PUT
delete	DELETE	DELETE

# GET

- Denna begäran används för att hämta en resurs från en server. Om du utför en "GET" förfrågan så letar servern efter de uppgifter du begärde och skickar tillbaka den till dig. Med andra ord utför en "GET" förfrågan en "READ" -operation. Detta är en standard-förfrågningsmetod.



# POST

- Denna begäran används för att skapa en ny resurs på en server. Om du utför en "POST" förfrågan så skapar servern en ny post i databasen och berättar dessutom om skapandet lyckades eller inte. Med andra ord utför en "POST" förfrågan av en "CREATE" -åtgärd.

# PUT & PATCH

- Dessa två förfrågningar används för att uppdatera en resurs på en server. Om du utför en "PUT" - eller "PATCH" -förfrågan uppdaterar servern en post/"entry" i databasen och berättar om uppdateringen lyckades eller inte. Med andra ord utför en "PUT" eller "PATCH" en begäran om "UPDATE".

# DELETE

- Denna begäran används för att ta bort en resurs från en server. Om du utför en "DELETE" förfrågan raderar servern en post/"entry" i databasen och berättar om att borttagningen lyckades. Med andra ord utför en "DELETE" begäran en "DELETE" åtgärd.

# HEADERS

- **Headers** används för att tillhandahålla information till både klienten och servern
- Den kan användas för många ändamål, t.ex. autentisering och information om kroppsinnehållet även kallat "body".
- [HTTP Headers referens](#)

# Headers

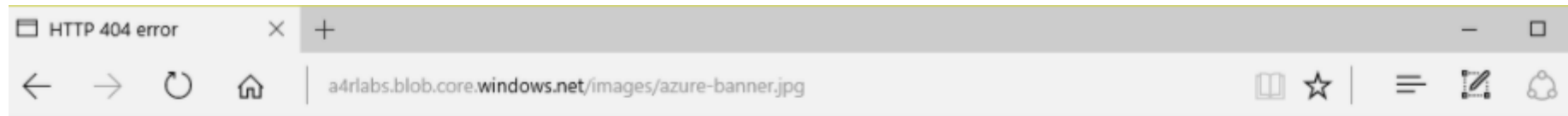
- Detta exempel säger till servern att förvänta sig ett "JSON content"

"Content-Type: application/json" <https://api.github.com>

```
{
  "current_user_url": "https://api.github.com/user",
  "current_user_authorizations_html_url": "https://github.com/settings/connections/applications{/client_id}",
  "authorizations_url": "https://api.github.com/authorizations",
  "code_search_url": "https://api.github.com/search/code?q={query}{&page,per_page,sort,order}",
  "commit_search_url": "https://api.github.com/search/commits?q={query}{&page,per_page,sort,order}",
  "emails_url": "https://api.github.com/user/emails",
  "emojis_url": "https://api.github.com/emojis",
  "events_url": "https://api.github.com/events",
  "feeds_url": "https://api.github.com/feeds",
  "followers_url": "https://api.github.com/user/followers",
  "following_url": "https://api.github.com/user/following{/target}",
  "gists_url": "https://api.github.com/gists{/gist_id}",
  "hub_url": "https://api.github.com/hub",
  "issue_search_url": "https://api.github.com/search/issues?q={query}{&page,per_page,sort,order}",
  "issues_url": "https://api.github.com/issues",
  "keys_url": "https://api.github.com/user/keys",
  "label_search_url": "https://api.github.com/search/labels?q={query}&repository_id={repository_id}{&page,per",
  "notifications_url": "https://api.github.com/notifications",
  "organization_url": "https://api.github.com/orgs/{org}",
  "organization_repositories_url": "https://api.github.com/orgs/{org}/repos{?type,page,per_page,sort}",
  "organization_teams_url": "https://api.github.com/orgs/{org}/teams",
  "public_gists_url": "https://api.github.com/gists/public",
  "rate_limit_url": "https://api.github.com/rate_limit",
  "repository_url": "https://api.github.com/repos/{owner}/{repo}",
  "repository_search_url": "https://api.github.com/search/repositories?q={query}{&page,per_page,sort,order}",
  "current_user_repositories_url": "https://api.github.com/user/repos{?type,page,per_page,sort}",
  "starred_url": "https://api.github.com/user/starred{/owner}/{repo}",
  "starred_gists_url": "https://api.github.com/gists/starred",
  "user_url": "https://api.github.com/users/{user}",
  "user_organizations_url": "https://api.github.com/user/orgs",
  "user_repositories_url": "https://api.github.com/users/{user}/repos{?type,page,per_page,sort}",
  "user_search_url": "https://api.github.com/search/users?q={query}{&page,per_page,sort,order}"
}
```

Body (JSON)

# HTTP 404 error? Vad betyder detta?



HTTP 404 error  
That's odd... Microsoft Edge can't find this page

Try this

- Retype the web address
- [Go back to the last page](#)

# HTTP status koder / Error meddelande

- 200+ betyder att begäran har lyckats
- 300+ betyder att begäran omdirigeras till en annan URL
- 400+ betyder ett fel som kommer från klienten.
- 500+ menas ett fel som kommer från servern.



# LAB

- 1. Skapa ett Github konto och skapa ett repository
- 2. Ladda ner [Postman](#)
- 3. Slå mot githubs api och hämta ut dina repository och se ditt JSON objekt i Postman.
- 4. Installera [request](#): **pip install requests**
- 5. Gör samma sak som punkt 3, men denna gång med hjälp av python.
- 6.