

REST

Sebastian
Lindgren

REST

REST

Sebastian Lindgren

Välkomna!

REST

Sebastian
Lindgren

REST

Välkomna till denna lektion där vi kommer att gå igenom **REST** API:er.

Från webb till API

REST

Sebastian
Lindgren

REST

Det vi hållit på med under de första två veckorna med Laravel har varit Webb-relaterat. Vi har skapat en webbsida som man kommer åt så som vi är vana vid, via en browser.

Introduktion

REST

Sebastian
Lindgren

REST

Det vi kommer att arbeta med idag är att se hur vi kan skapa API:er som är så kallat RESTful. Vi börjar med en introduktion där vi går igenom:

- Vad är RESTful?
- Hur arbetar man med RESTful?
- Varför ska man förstå sig på detta?

Introduktion

REST

Sebastian
Lindgren

REST

Vi fortsätter sedan med koncept i RESTful:

- Principer
- Uniformt
- Resurser och representationer
- Stateless

Efter detta kollar vi på ett litet hands-on exempel. Ni kommer att få arbeta med detta mer på onsdag när ni får skapa ett API i Laravel.

REST

REST

Sebastian
Lindgren

REST

Vad är REST?

REST

REST

Sebastian
Lindgren

REST

Innan vi börjar där, låt oss prata om tjänster på webben.

REST

REST

Sebastian
Lindgren

REST

Tjänster på webben syftar som ni vet till någon form av webbapplikation. Ett enklare sätt att resonera kring tjänster på webben är att tänka att de har API:er.

REST

REST

Sebastian
Lindgren

REST

Så vad är ett API nu igen? Det står för **Application Programming Interface** och dessa används för i princip allting när det kommer till tjänster online.

Öppna och Stängda

REST

Sebastian
Lindgren

REST

API:er kan vara både öppna och stängda.

- De öppna har ofta vissa restriktioner, till exempel i hur ofta man får komma åt dem. Men de går att komma åt öppet via internet.
- De stängda API:erna är bara menade att komma åt inifrån en väldigt begränsad miljö och är inte menad att komma åt via internet såvida man inte går genom VPN eller en väldigt ordentlig brandvägg.

<https://www.techtarget.com/searchapparchitecture/definition/open-API-public-API>

Exempel på API:er

REST

Sebastian
Lindgren

REST

Exempel på API:er:

- Google Sheets
- TMDB
- Trafikinformation
- Väder

API

REST

Sebastian
Lindgren

REST

API används och beskriver hur applikationer kommunicerar med varandra. Det är alltså API:et som är det som används för att hantera kommunikationen.

REST

Sebastian
Lindgren

REST

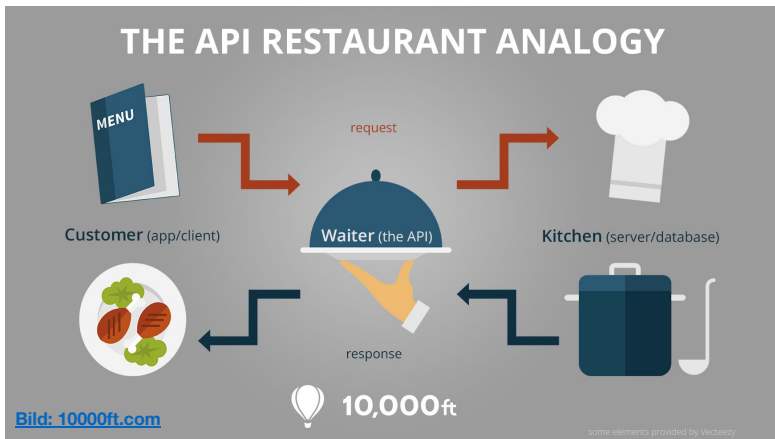
För att minnas vad ett API är och hur det fungerar kan man använda sig av restauranganalogin.

REST

REST

Sebastian
Lindgren

REST



Det vi använder är en server-klient arkitektur.

REST

REST

Sebastian
Lindgren

REST

22



Client-Server architecture

REST

REST

Sebastian
Lindgren

REST

23



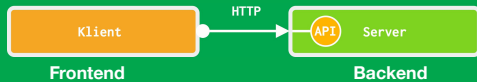
REST

REST

Sebastian
Lindgren

REST

24



REST

REST

Sebastian
Lindgren

REST

25

CREATE

READ

UPDATE

DELETE

API 1

Server
A

Formalisera

REST

Sebastian
Lindgren

REST

Det här är all-well and good. Men det kan lätt bli lite vilda västern om vi inte har regler för hur API:erna ska fungera rent praktiskt. Vi skulle gärna vilja strukturera eller formalisera kommunikationen.

Regler och struktur

REST

Sebastian
Lindgren

REST

Det är när vi vill införa lite regler och struktur som **REST** kommer in i bilden.

REST

REST

Sebastian
Lindgren

REST

REST står för

- **R**epresentational
- **S**tate
- **T**ransfer

REST

REST

Sebastian
Lindgren

REST

Så vad är då REST? Det är helt enkelt en samling konventioner för hur API:er ska utformas och fungera.

Konventioner i REST

REST

Sebastian
Lindgren

REST

Så vilka konventioner finns i REST?

- ① Separation av klient och server
- ② Statelessness
- ③ Svar som går att cache:a
- ④ Uniformt interface

Separation av klient och server

REST

Sebastian
Lindgren

REST

- Vår frontend och backend API kan utvecklas oberoende av varandra.
- Om kod ändras i FE påverkas inte BE, och vice versa.
- Så länge varje sida vet hur meddelanden ska formatteras och skickas, så kan de hållas isär
- I REST kan olika klienter träffa samma URI:er i ett och samma API och få samma svar

Statelessness

REST

Sebastian
Lindgren

REST

API:er som följer REST är s.k. "stateless". Detta betyder att servern inte behöver veta något om i vilket tillstånd klienternas applikation är i. På ren svenska: *"varje fråga från klient till server måste innehålla all information som behövs för att förstå frågan..."*

Sessions

REST

Sebastian
Lindgren

REST

På grund av statelessness så kommer inte en RESTful web service server att ta hand om olika sessionsdata. Det är upp till klienten att ha med relevant kontext och t.ex. användarsession och sessionsinformation. En client skickar alltså nödvändig information till servern som plockar den och använda den för att fortsatt ta hand om klientens request. https:

[//www.tutorialspoint.com/restful/restful_statelessness.htm](https://www.tutorialspoint.com/restful/restful_statelessness.htm)

Cache

REST

Sebastian
Lindgren

REST

Svar ska gå att cachea! Detta kommer vi att titta närmare på vid ett senare tillfälle. Nyfikna kan läsa lite mer här https://www.tutorialspoint.com/restful/restful_caching.htm.

Uniformt interface

REST

Sebastian
Lindgren

REST

Detta innefattar följande åtaganden:

- ① Identifiering av resurser
- ② Manipulation av resurser genom representationer
- ③ Självbeskrivande meddelanden
- ④ Länka resurser med varandra

Identifiering av resurser

REST

Sebastian
Lindgren

REST

Saker=Resurser

Identifisering av resurser

REST

Sebastian
Lindgren

REST

53

/api/books

Books = sak = resurs

Manipulation av resurser genom representationer

REST

Sebastian
Lindgren

REST

Representationer syftar till olika former av resurser.

Manipulation av resurser genom representationer

REST

Sebastian
Lindgren

REST

Beroende på HTTP verb (representation) får vi olika former och händelser på resurser

Manipulation av resurser genom representationer

REST

Sebastian
Lindgren

REST

57

CREATE

READ

UPDATE

DELETE

API 1

Server
A

**REMEMBER
ME?!**

Manipulation av resurser genom representationer

REST

Sebastian
Lindgren

REST

**GET — hämta en specifik resurs(via id)
eller en samling resurser (i.e. /books/1
eller /books)**

58

POST — skapa en ny resurs

**PUT — uppdatera en specifik resurs (via
id)**

DELETE — ta bort en specifik resurs via id

Självbeskrivande meddelanden

REST

Sebastian
Lindgren

REST

Svar som returneras från servern ska vara självbeskrivande. Den ska innehålla både:

- Status (HTTP)
- Svarsdatan på förfrågan i lämpligt format

Oftast brukar man nuförtiden använda formatet JSON.

Länka resurser med varandra

REST

Sebastian
Lindgren

REST

Ett praktexempel på det här brukar vara paginering i långa resultat. Föreställ er att ett svar på en listning av t.ex. orders ser ut såhär (se nästa sida).

Länka resurser med varandra

REST

Sebastian
Lindgren

REST

67

```
[
  // "GET /api/orders"
  {
    "order": {
      "amount": "11",
      "product": {
        "ref":
"http://store.company.com/api/products/1231"
      },
      "customer": {
        "ref":
"http://store.company.com/api/customers/553"
      }
    }
  }
  //...
]
```

Länka resurser med varandra

REST

Sebastian
Lindgren

REST

68

```
[
  // "GET /api/orders"
  {
    "order": {
      "amount": "11",
      "product": {
        "ref":
"http://store.company.com/api/products/1231"
      },
      "customer": {
        "ref":
"http://store.company.com/api/customers/553"
      }
    }
  }
  //...
]
```

HATEOAS

Hypermedia as the engine of application state

Länka resurser med varandra

REST

Sebastian
Lindgren

REST

Utöver dessa principer och riktlinjer finns det även ytterligare två, som inte är direkt relevanta för oss i nuläget

För de nyfikna

REST

Sebastian
Lindgren

REST

Läs gärna mer här: <https://restfulapi.net/>

Varför

REST

Sebastian
Lindgren

REST

Så varför ska vi lära oss detta? Jo eftersom majoriteten av API:er som skapas applicerar dessa (men oftast inte alla) principer.

Underlättar

REST

Sebastian
Lindgren

REST

De underlättar vid dokumentation och förståelse för andra utvecklare etc.

Skrivuppgift

REST

Sebastian
Lindgren

REST

Nu har vi hunnit gå igenom en del om RESTful. Så här kommer en skrivuppgift.

- Berätta lite om vad ett API är
- Ge en introduktion till REST
- Berätta vad som krävs för att göra API:et RESTful
- Berätta lite närmare om de olika konventionerna i REST API:er
- Läs på lite mer om de olika koncepten i REST (e.g. här <https://restfulapi.net/>) och berätta lite närmare om RESTful API:er.

Det var det

REST

Sebastian
Lindgren

REST

Låt oss se om vi kan börja titta på hur man skulle kunna börja bygga ett API med Laravel.

REST

Sebastian
Lindgren

REST