

# Lesson 6: REST & GraphQL

Wannes Fransen & Tom Eversdijk

UC Leuven

2020



# HTTP methods

## Request types:

- ▶ GET : request data from a resource
- ▶ POST : submit data to be processed to a specified resource
- ▶ PUT : replaces all current representations of the resource with the request payload.
- ▶ DELETE
- ▶ HEAD : identical as GET, but without response body
- ▶ OPTIONS : returns supported HTTP methods
- ▶ PATCH : update partial resources

# Idempotent methods

## Definition

Methods that can be called many times without different outcomes.

## Example

- ▶ `elixir_awesomeeness = 42` -> Idempotent
- ▶ `elixir_awesomeeness++` -> Not idempotent

# Safe methods

## Definition

Methods that do not modify resources.

## Example

- ▶ GET localhost:4000/users -> doesn't modify resources
- ▶ POST / PUT / DELETE -> modifies resources

# Overview

Method	Idempotent	Safe
GET	Y	Y
POST	N	N
PUT	Y	N
DELETE	Y	N
HEAD	Y	Y
OPTIONS	Y	Y
PATCH	N	N



# What is REST

## Representational state transfer

- ▶ Software architectural style
  - ▶ 6 constraints how **resources** are addressed/defined on the web
  - ▶ Reminder: URI = Uniform **Resource** Identifier
  - ▶ REST architectural style aka RESTful web services
- 
- ▶ RESTful services provide interoperability between computer systems on the Internet
  - ▶ Allow requesting systems to access & manipulate textual representations of web resources through uniform and predefined set of stateless operations



# REST architectural constraints

- ▶ Client-server architecture
- ▶ Statelessness
- ▶ Cacheability
- ▶ Layered system
- ▶ OPTIONAL: Code on demand
- ▶ Uniform interface

# RESTful Responses

## Response types

- ▶ XML
- ▶ JSON



# GraphQL

Maybe next year...