Handling Common Types of Integration

Kevin Dockx
ARCHITECT

@KevinDockx https://www.kevindockx.com

Coming Up

Integrating with an API

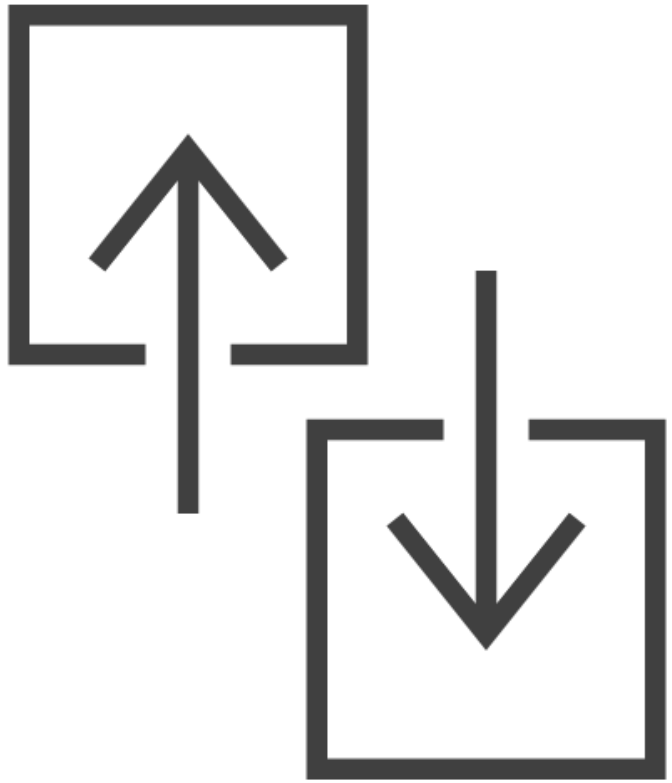
- Read (GET)
- Create (POST)
- Update (PUT)
- Delete (DELETE)
- Studying different approaches will lead us to the best practice

Content Negotiation

Demo

Getting a Resource

Working with Headers and Content Negotiation



HTTP headers allow passing additional information with each request or response

- name : value
- name : partial value1, partial value2

Working with Headers and Content Negotiation

Request headers

Contain information on the resource to be fetched, or about the client itself

Are provided by the client

Accept: application/json
Accept: application/json, text/html

Response headers

Contain information on the generated response, or about the server

Are provided by the server

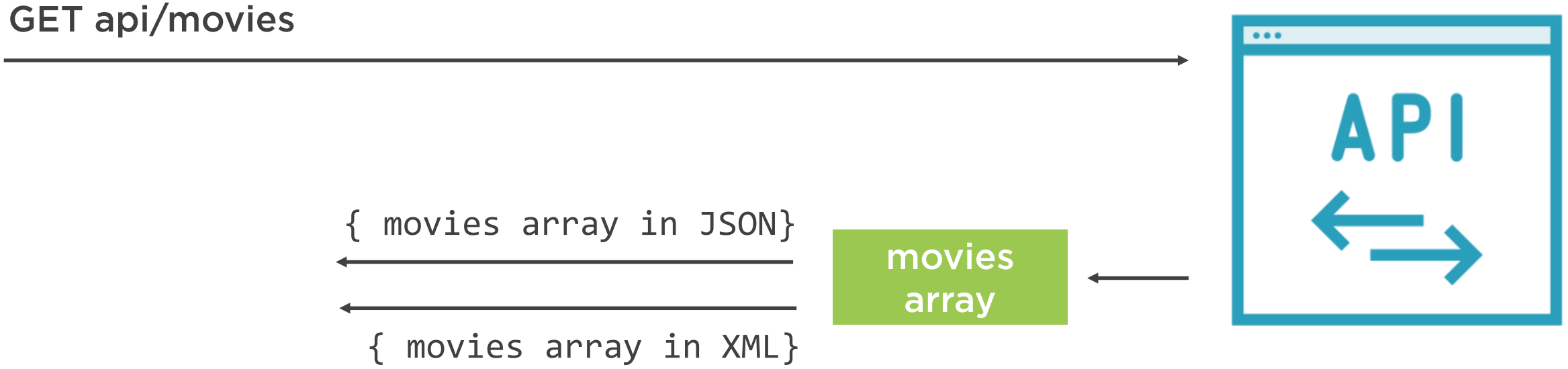
Content-Type: application/json

Working with Headers and Content Negotiation

It's best practice to be as strict as possible

- For example, setting an Accept header (obligatory in RESTful systems) improves reliability

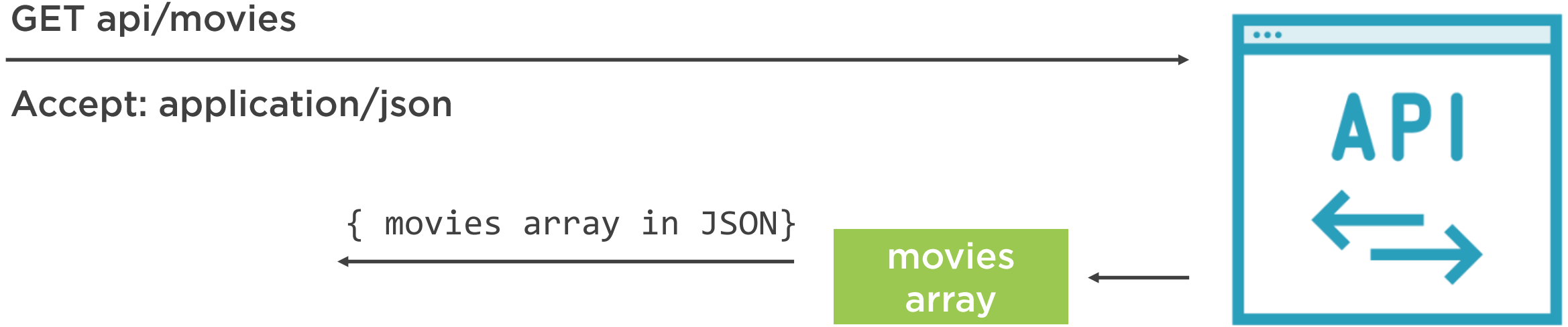
Working with Headers and Content Negotiation



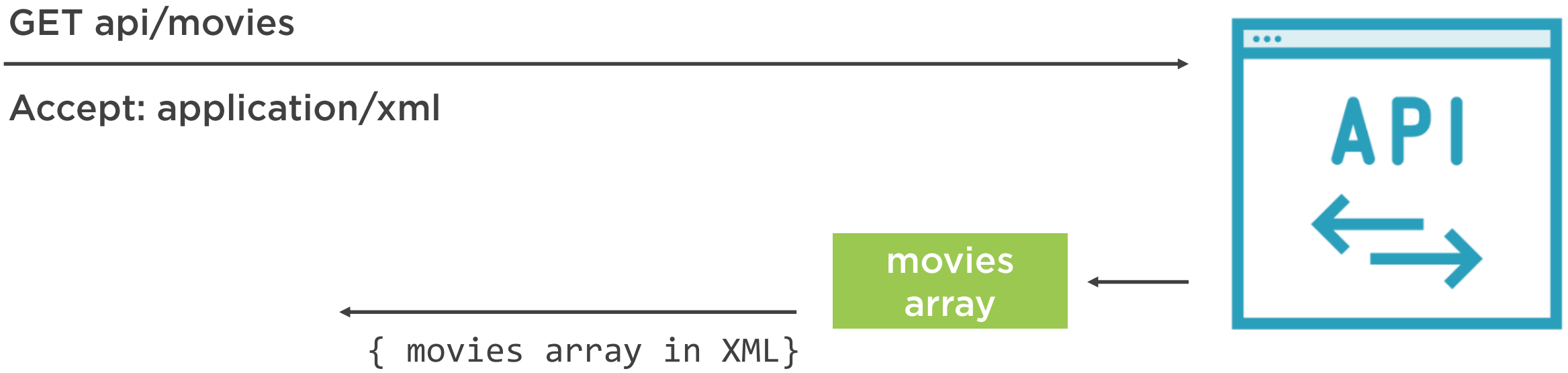
Working with Headers and Content Negotiation

GET api/movies

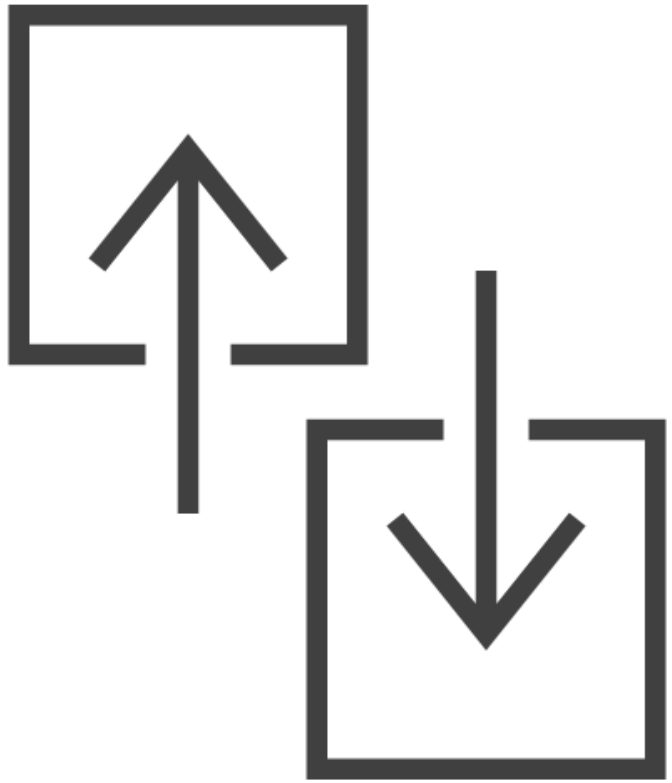
Accept: application/json



Working with Headers and Content Negotiation



Working with Headers and Content Negotiation



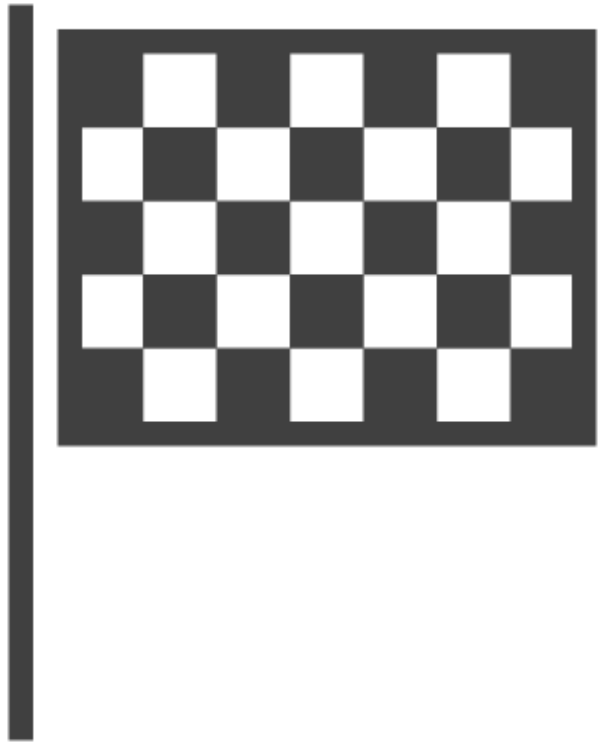
Content negotiation is the mechanism used for serving different representations of a resource at the same URI

- Accept, Accept-Encoding, Accept-Language, Accept-Charset

Demo

Manipulating Request Headers

Indicating Preference with the Relative Quality Parameter



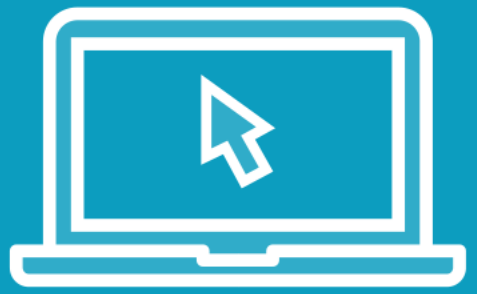
Equal preference

- Accept: application/json, application/xml

Indicating preference

- Accept: application/json, application/xml;q=0.9

Demo



Indicating Preference with the Relative Quality Parameter

Demo

Working With HttpRequestMessage Directly

Demo

Creating a Resource

Setting Request Headers

HttpClient .DefaultRequest Headers

For defaults across requests

HttpRequest Message .Headers

Headers applicable whether or not a request has a body

HttpRequest Message .Content .Headers

Headers related to the body of a request

Inspecting Content Types

HttpRequestMessage.Content is of type HttpContent

Use a derived class that matches the content of the message

- StringContent, ObjectContent, ByteArrayContent, StreamContent, ...
  - Optimized for their type of content

Demo

Updating a Resource

Demo

Deleting a Resource

Demo

Using Shortcuts

Summary

Request headers contain more information on the resource to be fetched, or about the client itself

- We are responsible for setting these

The headers of a response contain information on the generated response or server

- We are responsible for reading these and acting accordingly

Summary

Default values that remain the same across requests

- HttpClient.DefaultRequestHeaders

Headers that apply to requests regardless of it having a request body

- HttpRequestMessage.Headers

Headers related to the request body

- HttpRequestMessage.Content.Headers

Summary

Shortcuts can come in handy, but if you need full control it's best to use HttpRequestMessage directly