

**Glue Software Engineering AG**

Schwarztorstrasse 31, CH-3007 Bern

+41 31 385 30 11

[www.glue.ch](http://www.glue.ch)

[info@glue.ch](mailto:info@glue.ch)

# Angular for Plain Old Java Engineers

## An introduction to SPA



# What will we see?

- **Part 0 – Introduction**
- **Part 1 – Architectural aspects**
- **Part 2 – Development with Angular**



# Who is the speaker ?

**Ciril Saner**

**Since 2013 Software Engineer @ Glue**

**Stay in touch**

- [ciril.saner@glue.ch](mailto:ciril.saner@glue.ch)
- <https://github.com/c-saner>

What about you?

# Important for part 2!

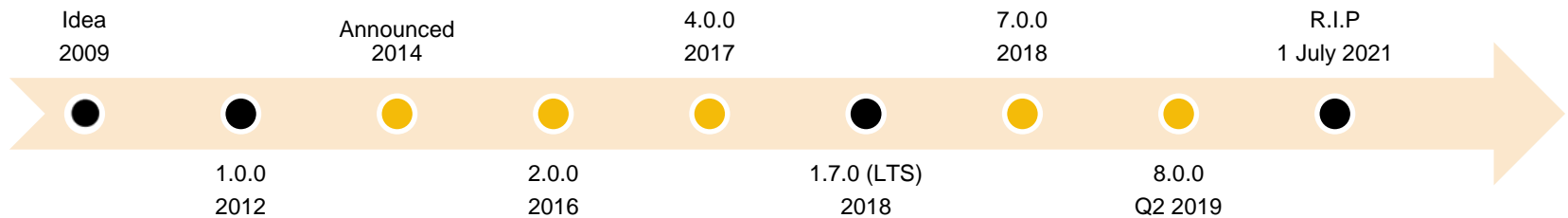
- **Install**
  - Node.js -> <https://nodejs.org/en/>
  - npm (installed with Node.js)
- **Demo with Visual Studio Code & Chrome**

Angular != AngularJS

# History of Angular & AngularJS

AngularJS ●

Angular ●





# Angular != AngularJS - differences

Angular	AngularJS
Component, directive	MVC design
TypeScript	JavaScript
(), [] bindings	ng –directives
Mobile support	No mobile support

SPA

# SPA ?

- **Wellness**
- **Société protectrice des animaux**
- **City (Belgium)**
- **S.p.A - Società per Azioni**
- **...**

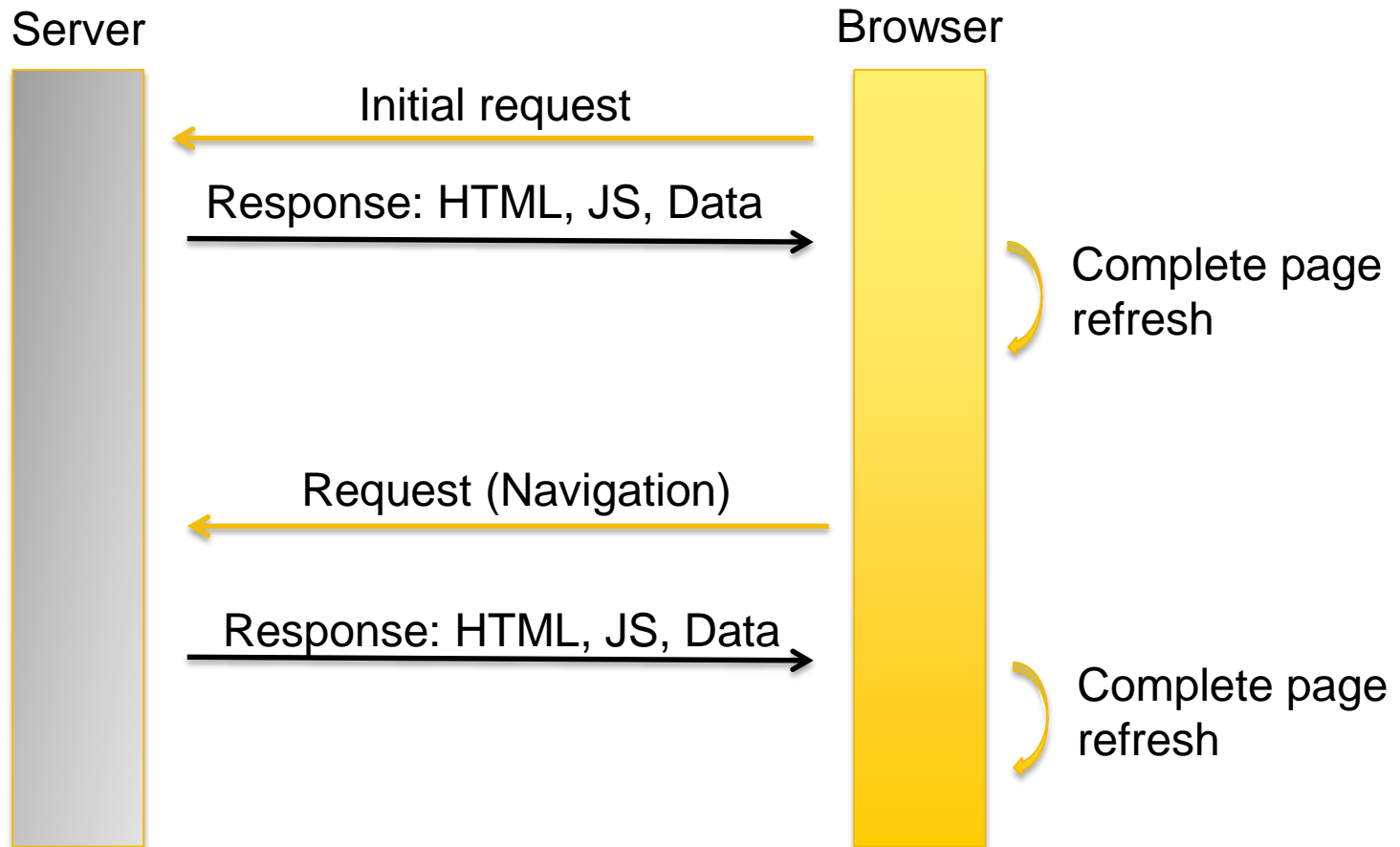
## SINGLE-PAGE APPLICATION

# SPA – small definition

- **A single-page application (SPA) is a web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server.<sup>1</sup>**

1. [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application)

# Traditional page refresh



# Traditional page refresh

The screenshot displays the Swiss Criminal Register (Strafregister) website. The breadcrumb navigation at the top reads: Bundesverwaltung > Departement: EJPD > Bundesamt für Justiz > Strafregister. The header includes the Swiss flag and the text 'Schweizerische Eidgenossenschaft', 'Confédération suisse', 'Confederazione Svizzera', and 'Confederaziun svizra'. The main title is 'EJPD Strafregisterauszug'. The left sidebar contains links: Startseite, Auszug bestellen, Privatauszug (highlighted), Sonderprivatauszug, Überprüfung von Auszügen, Tracking / Quittung, FAQ, and Impressum. The main content area is titled 'Lesen Sie zuerst aufmerksam...' and 'Das müssen Sie tun:'. It lists three steps: 1. Personalien & Adressen, 2. Bestell- & Zahlungsdetails, and 3. Drucken & abschliessen. A clock icon indicates a duration of 'Dauer ca. 10-15 Min.'. Below this, a red text box says 'und'. Another section titled 'Formular unterschreiben und mit der Ausweiskopie als Beilage einsenden.' includes a red warning: 'Ohne Ihre Unterschrift und Beilagen kein Auszug!'. A 'Datenerfassung' button is at the bottom. The right sidebar has a section 'Ich besitze eine SuisseID:' with a link 'Bestellung mit SuisseID' and the SuisseID logo. The URL at the bottom is 'https://www.e-service.admin.ch/crx/cms/content/strafregister/uebersicht\_de'.

Bundesverwaltung > Departement: EJPD > Bundesamt für Justiz > Strafregister

Helpdesk FAQ DE FR IT EN

Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

EJPD  
Strafregisterauszug

Startseite  
Auszug bestellen  
Privatauszug  
Sonderprivatauszug  
Überprüfung von Auszügen  
Tracking / Quittung  
FAQ  
Impressum

Lesen Sie zuerst aufmerksam...

Das müssen Sie tun:

Daten erfassen & drucken

- 1 Personalien & Adressen
- 2 Bestell- & Zahlungsdetails
- 3 Drucken & abschliessen

Dauer ca. 10-15 Min.

und

Formular unterschreiben und mit der Ausweiskopie als Beilage einsenden.

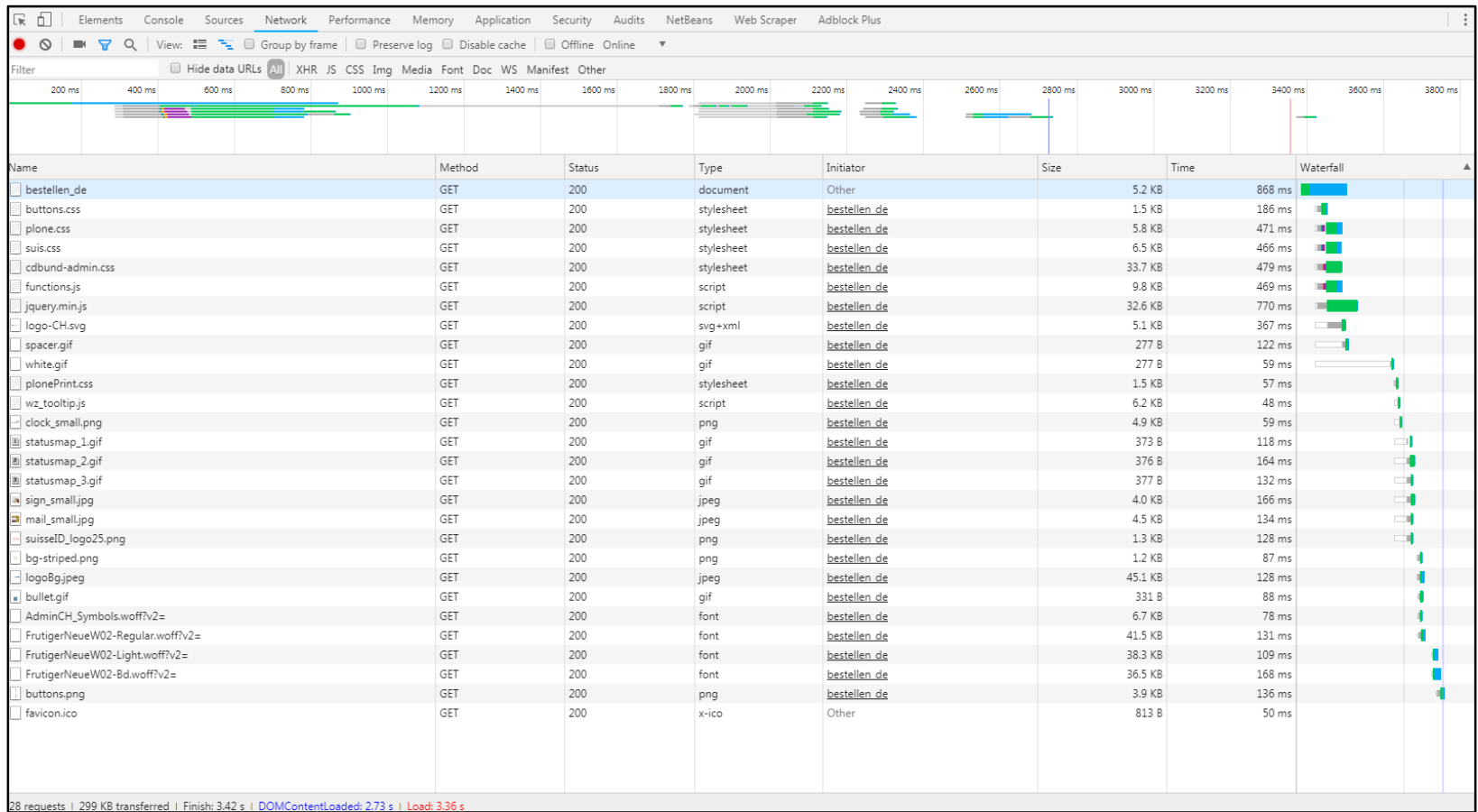
Ohne Ihre Unterschrift und Beilagen kein Auszug!

Datenerfassung

Ich besitze eine SuisseID:  
[Bestellung mit SuisseID](#)  
suisseID

[https://www.e-service.admin.ch/crx/cms/content/strafregister/uebersicht\\_de](https://www.e-service.admin.ch/crx/cms/content/strafregister/uebersicht_de)

# Traditional page refresh



# Traditional page refresh

Bundesverwaltung

Departement: EJPD

Bundesamt für Justiz

Strafregister

Helpdesk


FAQ

DE

FR

IT

EN



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

EJPD

Strafregisterauszug

TAN: B606D2838D6

Startseite

Auszug bestellen

**Privatauszug**

Sonderprivatauszug

Überprüfung von Auszügen

Tracking / Quittung

FAQ

Impressum

1

Personalien & Adressen

2

3

Bitte wählen Sie das Zutreffende aus

Staatsangehörigkeit(en)

☒ Schweizer Staatsbürger(in)

☐ ausländische(r) Staatsbürger(in)

☐ sowohl Schweizer als auch ausländische(r) Staatsbürger(in)

☐ staatenlos

Zivilstand

☐ ledig

☐ verheiratet (oder getrennt)

☐ geschieden

☐ verwitwet

☐ in eingetragener Partnerschaft

☐ aufgelöste Partnerschaft

Heutiger Wohnsitz

☐ in der Schweiz

☐ im Ausland

Abbrechen

Weiter

Erweiterte Optionen aktivieren

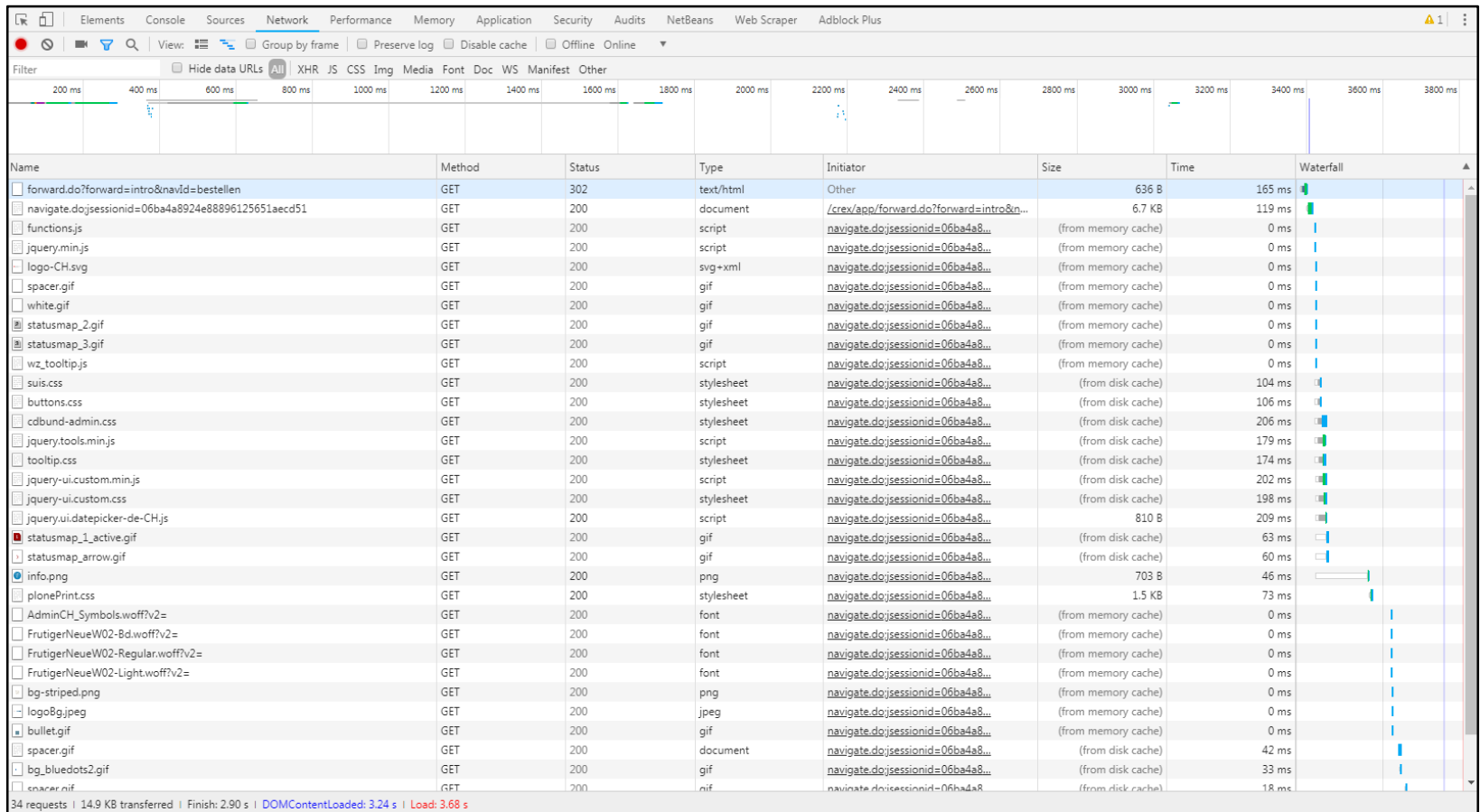
Zustellung an Dritte:

Beachten Sie die erweiterten Optionen.

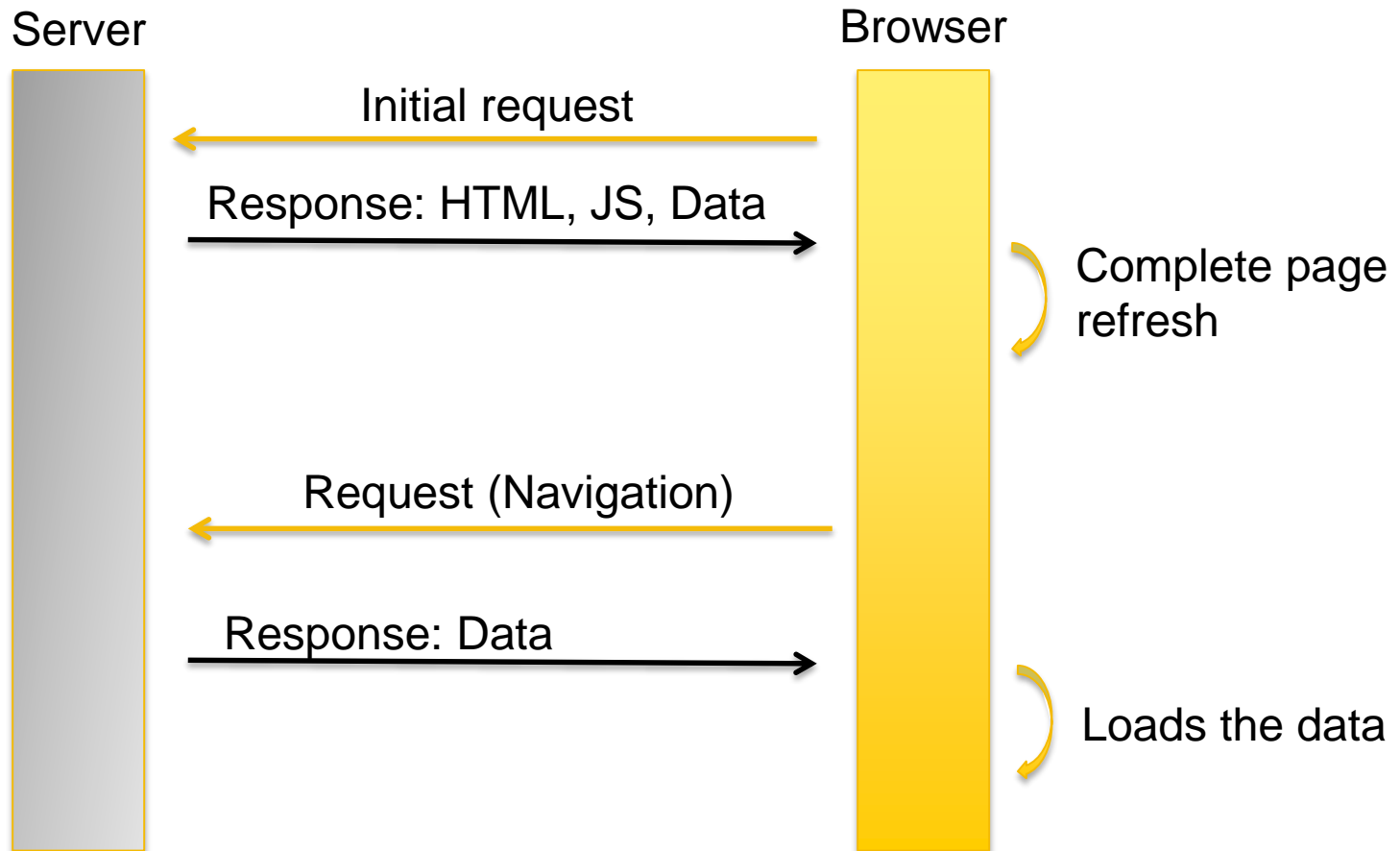




# Traditional page refresh



# SPA – page refresh



# SPA – page refresh

Vollmachten verwalten | x

https://www.suisse-tax.admin.ch/portal/#/verwaltung/bevollmaechtigung-verwalten

Apps Projects ToolChain Docs Bugs

Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

Eidgenössische Steuerverwaltung  
SuisseTax

Saner Cyril

Übersicht Allgemein Informationsaustausch Mehrwertsteuer Abgabe Radio TV Verrechnungssteuer Verwaltung

## Daten

Benutzerprofil verwalten

Bankverbindung korrigieren

Unternehmensdaten verwalten

## Berechtigungen

Berechtigungen verwalten

## Vollmachten verwalten

## Vollmachten verwalten

Unternehmen können Vollmachten an Benutzer von ESTV SuisseTax vergeben und sie damit zum sogenannten Superuser ernennen. Diese Superuser erhalten die Berechtigung, im Namen der Unternehmen Geschäfte über ESTV SuisseTax zu bearbeiten, können aber auch weitere Benutzer für ESTV SuisseTax berechtigen, ohne dass eine zusätzliche Vollmacht benötigt wird.

Falls bereits ein Superuser für ein Unternehmen existiert, sind Berechtigungen über diesen und nicht via Vollmacht zu beantragen.

### Pendente Vollmachten

Neue Vollmacht beantragen

Unternehmen	Vollmacht für	Beantragt	Status
Gemeinde Vals, Abwasserbeseitigung	VST	23.08.2018	Versendet
UBS AG	MWST	25.07.2018	Versendet
Hunde-Energie-Training GmbH	RTVG	01.06.2018	Versendet

## Kontakt

Unser Servicedesk unterstützt Sie gerne bei Fragen oder Problemen rund um ESTV SuisseTax.

Telefon (Bürozeiten)

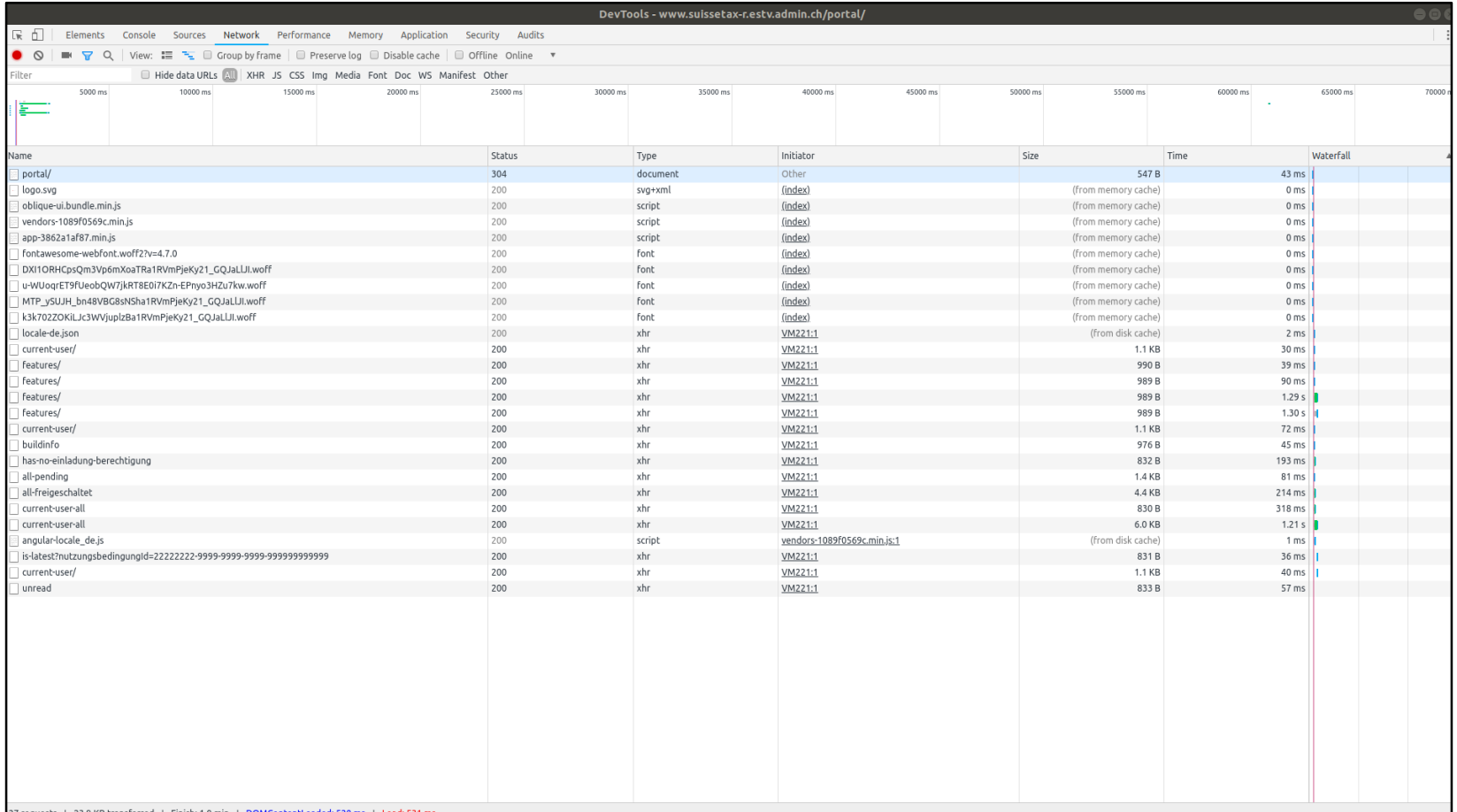
058 464 54 01

Oder nutzen Sie unser [Kontaktformular](#).

© 2019 Eidgenössische Steuerverwaltung v 1.3.0-33

[Rechtliche Grundlagen](#) [Kontakt](#)

# SPA – page refresh



# SPA – page refresh

The screenshot shows a web browser window with the URL <https://www.suisse-tax-estv.admin.ch/portal/#/verwaltung/bevollmaechtigung-einfuehrung>. The page is titled 'Neue Vollmacht beantragen' (New Power of Attorney). The left sidebar contains navigation links: 'Daten' (Benutzerprofil verwalten, Bankverbindung korrigieren, Unternehmensdaten verwalten), 'Berechtigungen' (Berechtigungen verwalten), and 'Vollmachten verwalten' (highlighted). The main content area has a sub-header 'Einleitung' (Introduction) and a section 'Wozu eine Vollmacht?' (Why a Power of Attorney?). It explains that businesses need a power of attorney to handle tax matters with ESTV SuisseTax. A blue information box states: 'Nach der Freischaltung einer Vollmacht für ESTV SuisseTax werden für die betroffenen Steuerbereiche keine Formulare per Post verschickt.' (After activation of a power of attorney for ESTV SuisseTax, no forms will be sent by post for the affected tax areas). Below this is the 'Ablauf der Bevollmächtigung' (Process of Authorization) section, which describes the step-by-step process from data entry to final activation. At the bottom right, there are two buttons: 'Abbrechen' (Cancel) and 'Vollmachtsantrag starten' (Start Power of Attorney Application). The right sidebar contains a 'Kontakt' (Contact) section with a phone number (058 464 54 01) and a link to the 'Kontaktformular' (Contact Form). The footer includes copyright information (© 2019 Eidgenössische Steuerverwaltung), version (v 1.3.0-33), and links to 'Rechtliche Grundlagen' (Legal Basis) and 'Kontakt'.

Neue Vollmacht beantragen

Einleitung

Wozu eine Vollmacht?

Um Geschäfte eines steuer- oder meldepflichtigen Unternehmens über ESTV SuisseTax bearbeiten zu können, müssen Sie für dieses Unternehmen bevollmächtigt sein. Dies gilt auch für Inhaber von Einzelfirmen. Eine Bevollmächtigung für mehrere unterschiedliche steuer- oder meldepflichtige Unternehmen ist ebenfalls möglich, diese müssen aber einzeln beantragt werden.

Nach der Freischaltung einer Vollmacht für ESTV SuisseTax werden für die betroffenen Steuerbereiche keine Formulare per Post verschickt.

Ablauf der Bevollmächtigung

Sie werden nun schrittweise durch die Datenerfassung für die Beantragung einer neuen Vollmacht geführt. Der Vollmachtsantrag wird anschliessend durch die ESTV dem steuerpflichtigen Unternehmen zugestellt. Die Vollmacht erlangt dabei erst ihre Gültigkeit, nachdem diese durch die zeichnungsberechtigten Vertreter des steuerpflichtigen Unternehmens unterzeichnet und anschliessend von der ESTV freigeschaltet wurde.

Abbrechen Vollmachtsantrag starten

Kontakt

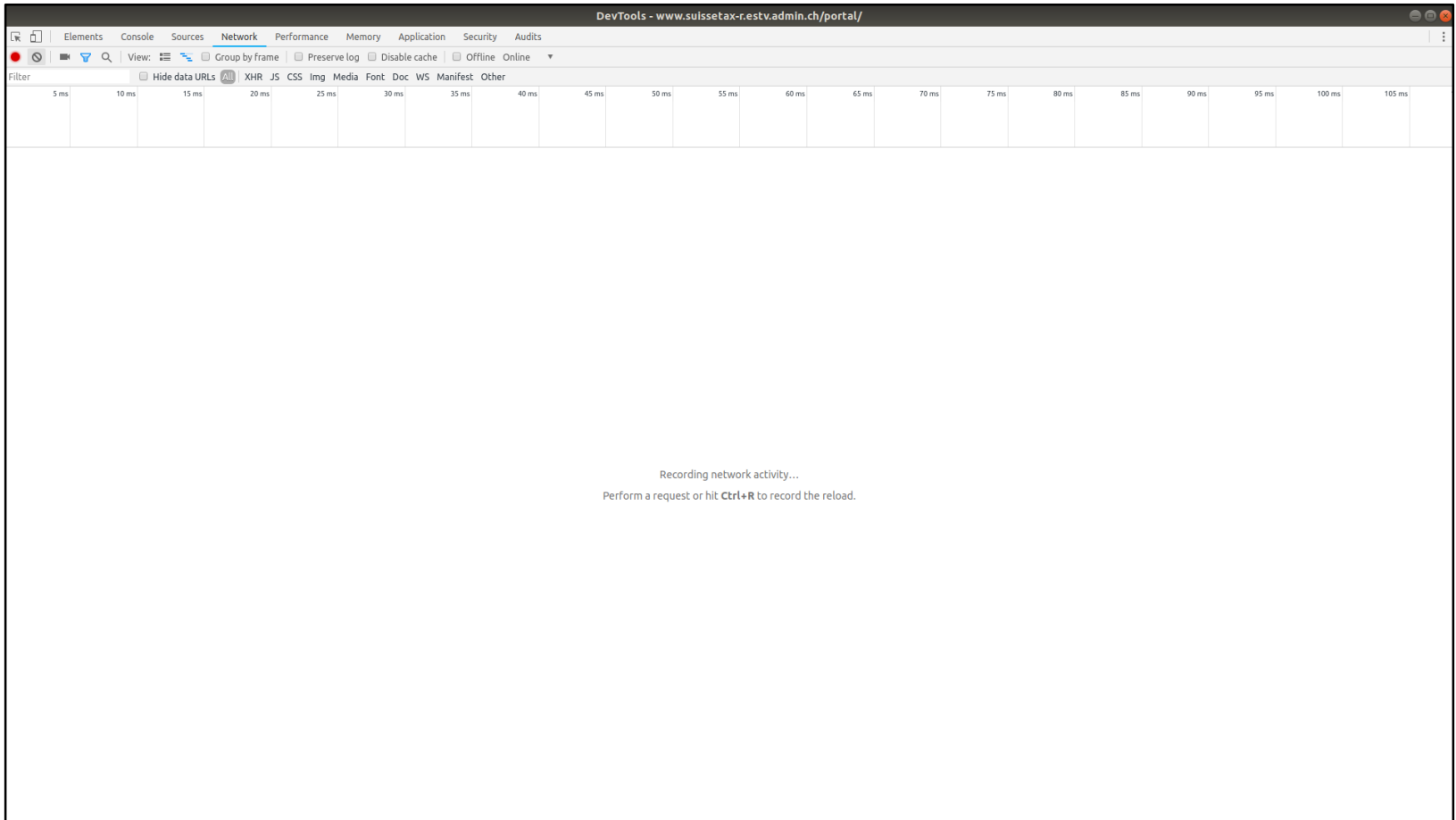
Unser Servicedesk unterstützt Sie gerne bei Fragen oder Problemen rund um ESTV SuisseTax.

Telefon (Bürozeiten)

058 464 54 01

Oder nutzen Sie unser [Kontaktformular](#).

# SPA – page refresh



# SPA – page refresh

Neue Vollmacht beantragen

Schritt 1: Wahl des steuerpflichtigen Unternehmens

Einzelnes Unternehmen | Mehrere Unternehmen

Unternehmen suchen

Suchen Sie nach dem steuerpflichtigen Unternehmen, für welches die Vollmacht gelten soll.

Suche nach Firmenname oder UID

Firmenname oder UID

Beispiele: Muster AG, CHE-123.456.789

Suchen

Kontakt

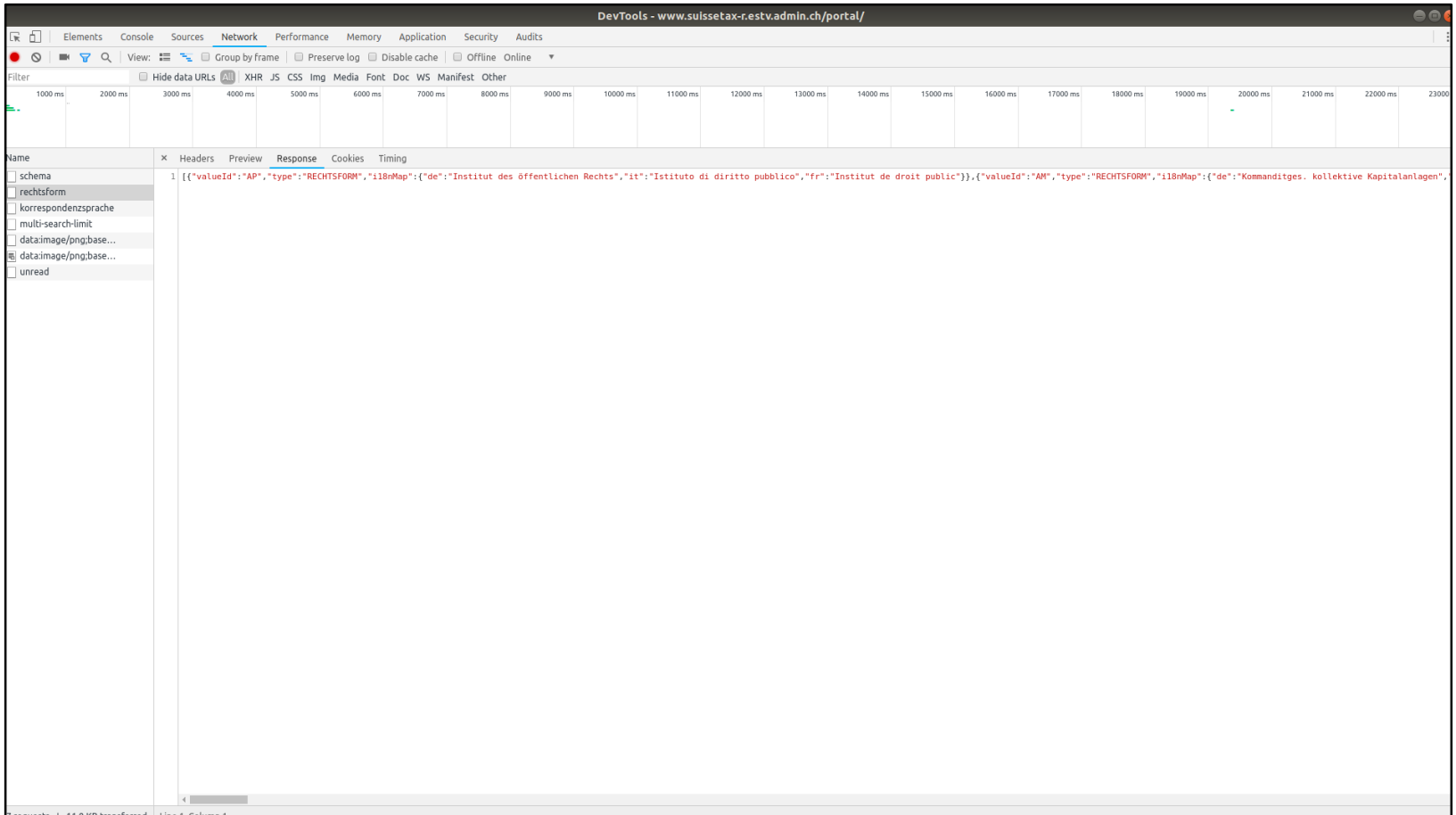
Unser Servicedesk unterstützt Sie gerne bei Fragen oder Problemen rund um ESTV SuisseTax.

Telefon (Bürozeiten)

058 464 54 01

Oder nutzen Sie unser [Kontaktformular](#).

# SPA – page refresh





SPA – Is resistance futile ?

# Technologies

# Technologies – History

- **HTML: 1993**
- **JavaScript: 1995**
- **JAVA: 1995**
- **CSS: 1996**
- **SPA : 2003**

# Used technologies

- **HTML5: 2014**
- **CSS3: Release by modules started in 2001**
  - Sass: 2006
  - Less: 2009
- **JavaScript – ECMAScript 6 : 2015**
  - TypeScript: 2012

# Frameworks – History

- **AngularJS: 2010**
- **ReactJS: 2013**
- **VueJS: 2014**
- **Spring boot: 2014**
- **Angular: 2016**

# Technologies – Problem?

- **Browser support**
  - Internet Explorer doesn't support ECMAScript 6
- **Technologies move quickly**
- **Framework selection**

JavaScript that scales

TypeScript

# TypeScript – JavaScript that scales

**TypeScript is typed superset of JavaScript that  
compiles to plain JavaScript.**

<https://www.typescriptlang.org/>



# TypeScript – JavaScript that scales



# TypeScript – JavaScript that scales

**ECMAScript (ES) is a trademarked scripting-language specification standardized by Ecma International in ECMA-262 and ISO/IEC 16262.**

**ECMAScript 2016 or ES7 is the edition of the specification.**

<https://www.ecma-international.org/>  
<https://en.wikipedia.org/wiki/ECMAScript>

# TypeScript – JavaScript that scales

**JavaScript (JS)** is a high-level, interpreted programming language that conforms to the ECMAScript specification.

<https://en.wikipedia.org/wiki/JavaScript>

# TypeScript – JavaScript that scales

- **ECMAScript: the specification**
- **ECMAScript 2016: the edition of the specification**
- **JavaScript: implementation of ECMAScript**
- **TypeScript: superset of JavaScript**

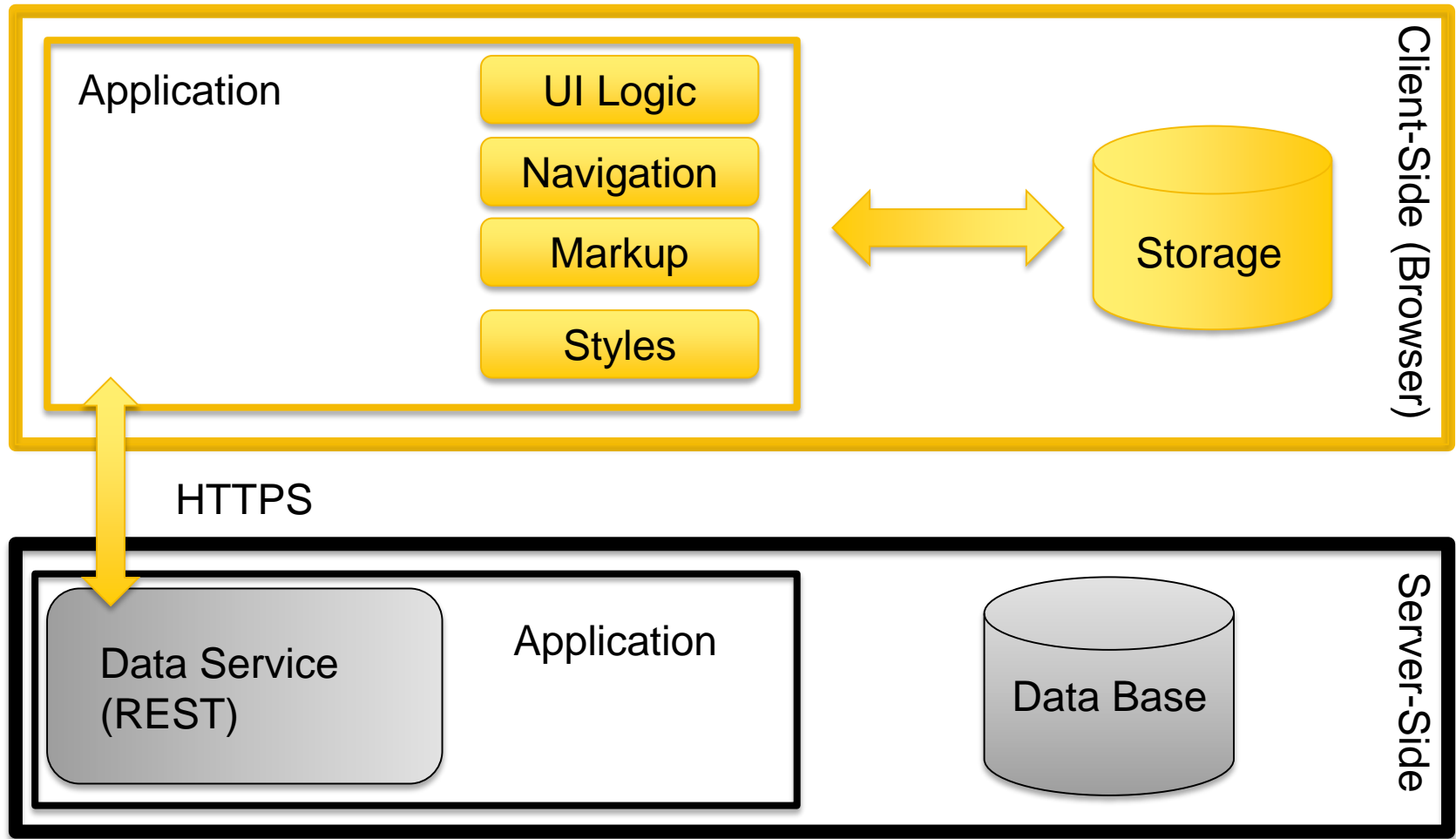
# TypeScript – JavaScript that scales

## Additional features

- Type annotation & compile-time type checking
- Generics
- Enums
- ...

# Part 1 - Architectural aspects

# SPA – Big Picture



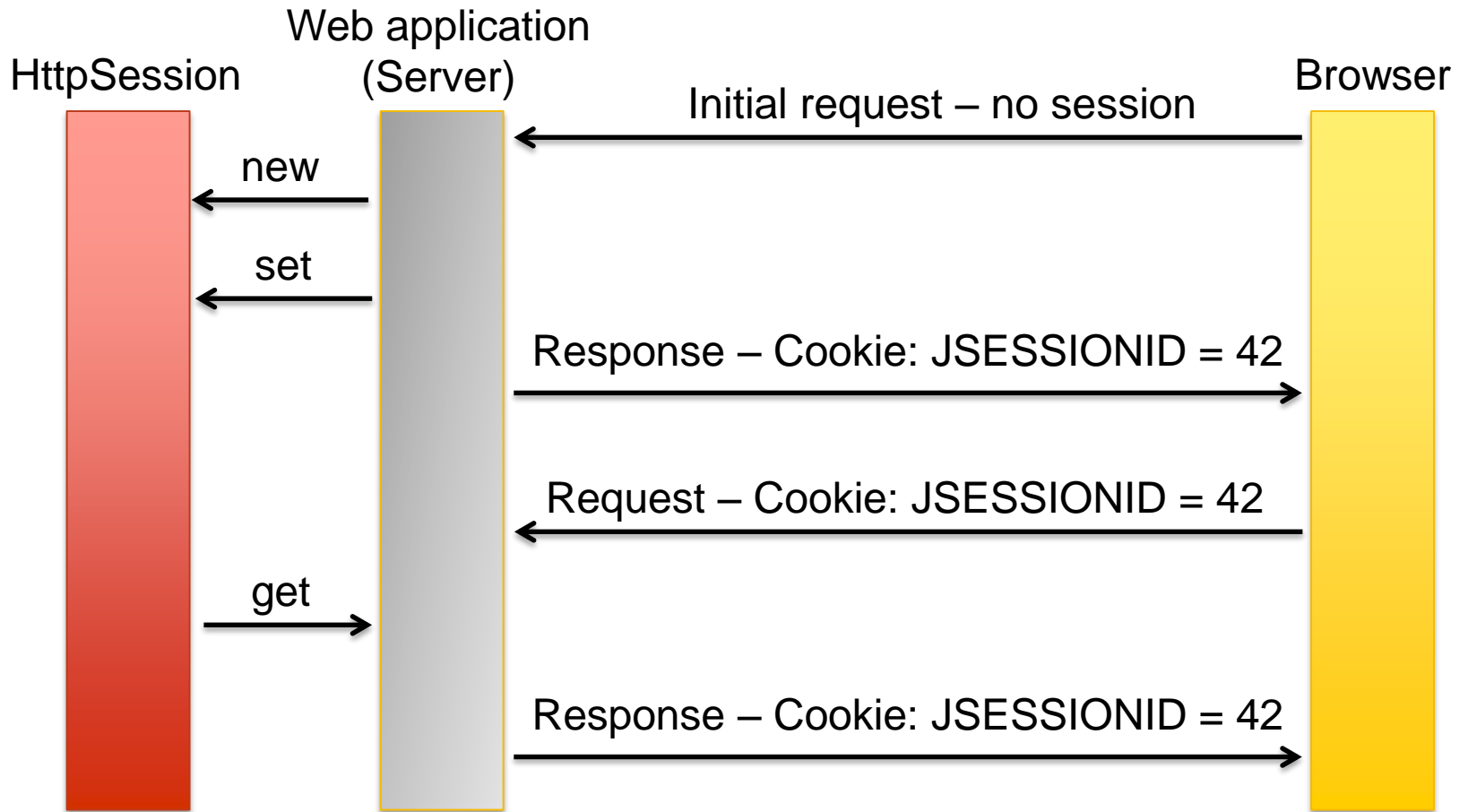
# Session handling



# What is a session ?

**Series of related requests that arrive at the server at a specific time interval from the same user agent.**

# Session handling – J2EE



# Session handling – SPA

- **HTTP is stateless protocol**
- **RESTful web services are stateless**

**How is the session managed in a SPA?**

# Session handling – SPA

## The frontend application

- Standalone & lives in the browser
- No HttpSession

## Storage before HTML5

- Cookies

## HTML5 Web Storage

- Local storage
- Session storage

# Session handling – SPA

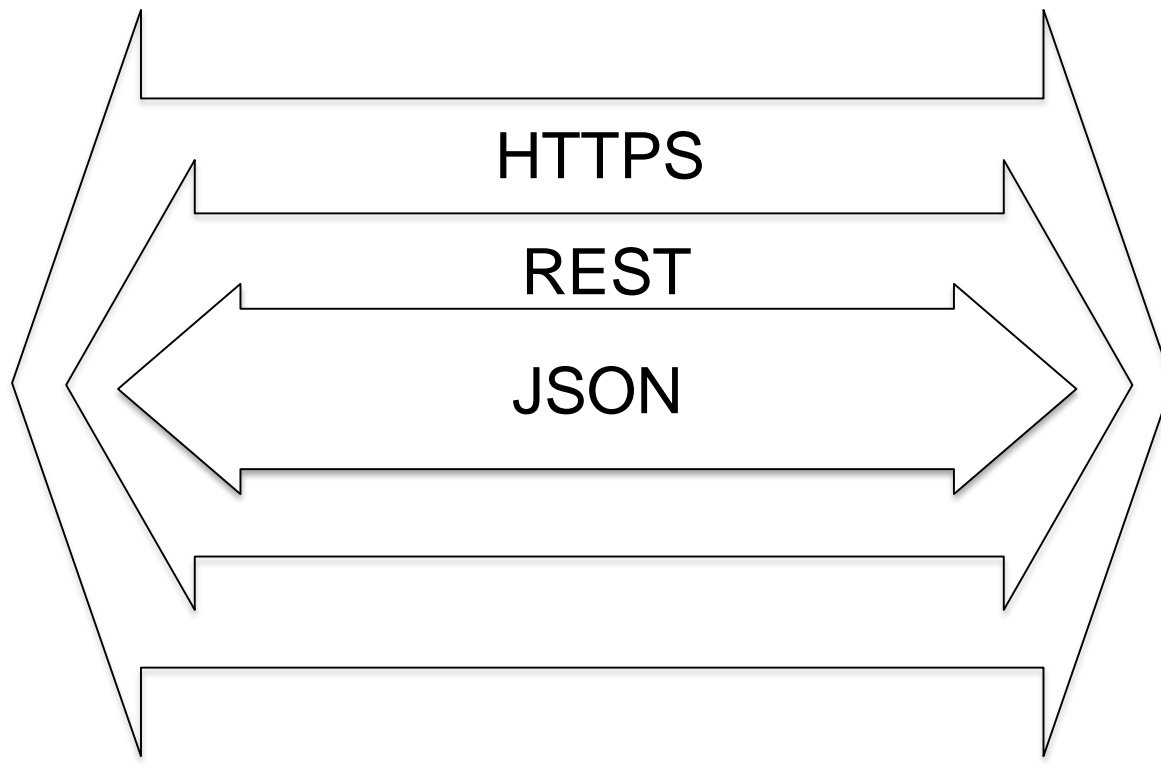
**We exchange an auth token if authentication is required.**

# Communication with the backend

# Communication with the backend

backend

frontend



# Communication with the backend





# Communication with the backend

## Developer changes...

- Variable name
- URLs
- Method types
- HTTP status returned
- DTO returned
- ...

**...only in the frontend or in the backend**

# Communication with the backend

## Java

```
public class Address {  
    private String strasse;  
}
```

## Typescript

```
export class Address {  
    street: string;  
}
```

# Communication with the backend

## Large / useless JSON exchanged

```
{
  "cases": [
    {
      "id": "1",
      "name": "Test 1",
      "documents": [
        {
          "id": "1",
          "name": "PDF 1",
          "content": "SOME PDF CONTENT"
        }
      ]
    }
  ]
}
```

# Security

# XSS: Cross-site scripting

**Type of injection, in which malicious scripts are injected into otherwise benign and trusted websites.**

**XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.**

[https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

# XSS: Cross-site scripting

## What can be achieved ?

- Modification of the **D**ocument **O**bject **M**odel (DOM)
- Send information to a third party

# XSS: Cross-site scripting

## Modus operandi (summarized)

- User can write content in a guestbook, forum, etc
- Web site allows uncontrolled content to be supplied by users
- Attacker enter malicious code in the content
- Victim open the page with the malicious content
- Script gets executed

# XSS: Cross-site scripting

## Countermeasures

- Input validation
- Output encoding: `<script>` is encoded `&lt;script&gt;`



# CSRF: Cross-Site Request Forgery

**Attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.**

[https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))

# CSRF: Cross-Site Request Forgery

## What can be achieved ?

The impact is limited to the capabilities exposed by the application.

- Victim is a normal user  
Can force the user to perform state changing requests like transferring funds, changing their email address, etc.
- Victim is an administrative account  
CSRF can compromise the entire web application.

# CSRF: Cross-Site Request Forgery

## Modus operandi (summarized)

- Attacker creates a reproducible link that executes a specific action on the target page
- Attacker places the link in a location that the victim is likely to visit while logged into using XSS
- Victim clicks on the link
- Attack is performed

# CSRF: Cross-Site Request Forgery

## Countermeasures

- No Cross-Site Scripting (XSS) vulnerabilities
- The server and client must cooperate to thwart this attack
- Token based mitigation

# XSS – CSRF – conclusion

## XSS

- Exploits the trust a user has in a website
- The website sends content, that the user executes because it comes from this site.

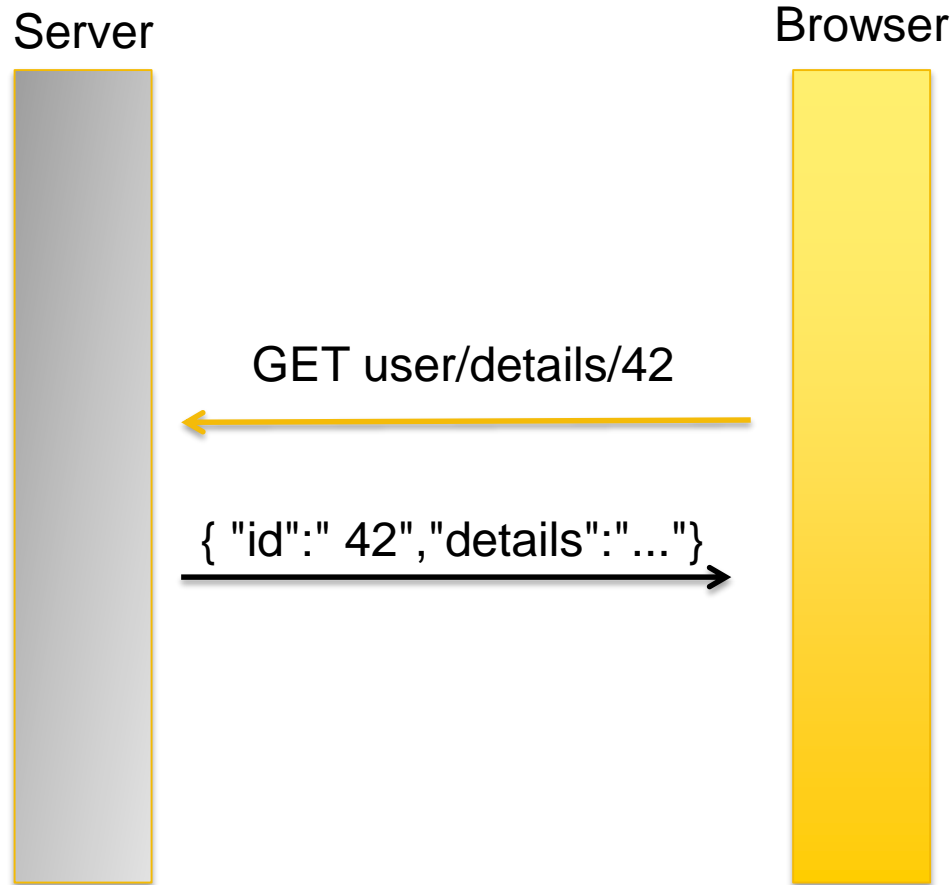
## CSRF

- Exploits the trust the site has in a user
- By making a request appear to come from a trusted user
- The server receives requests from the user and think it was sent on purpose.

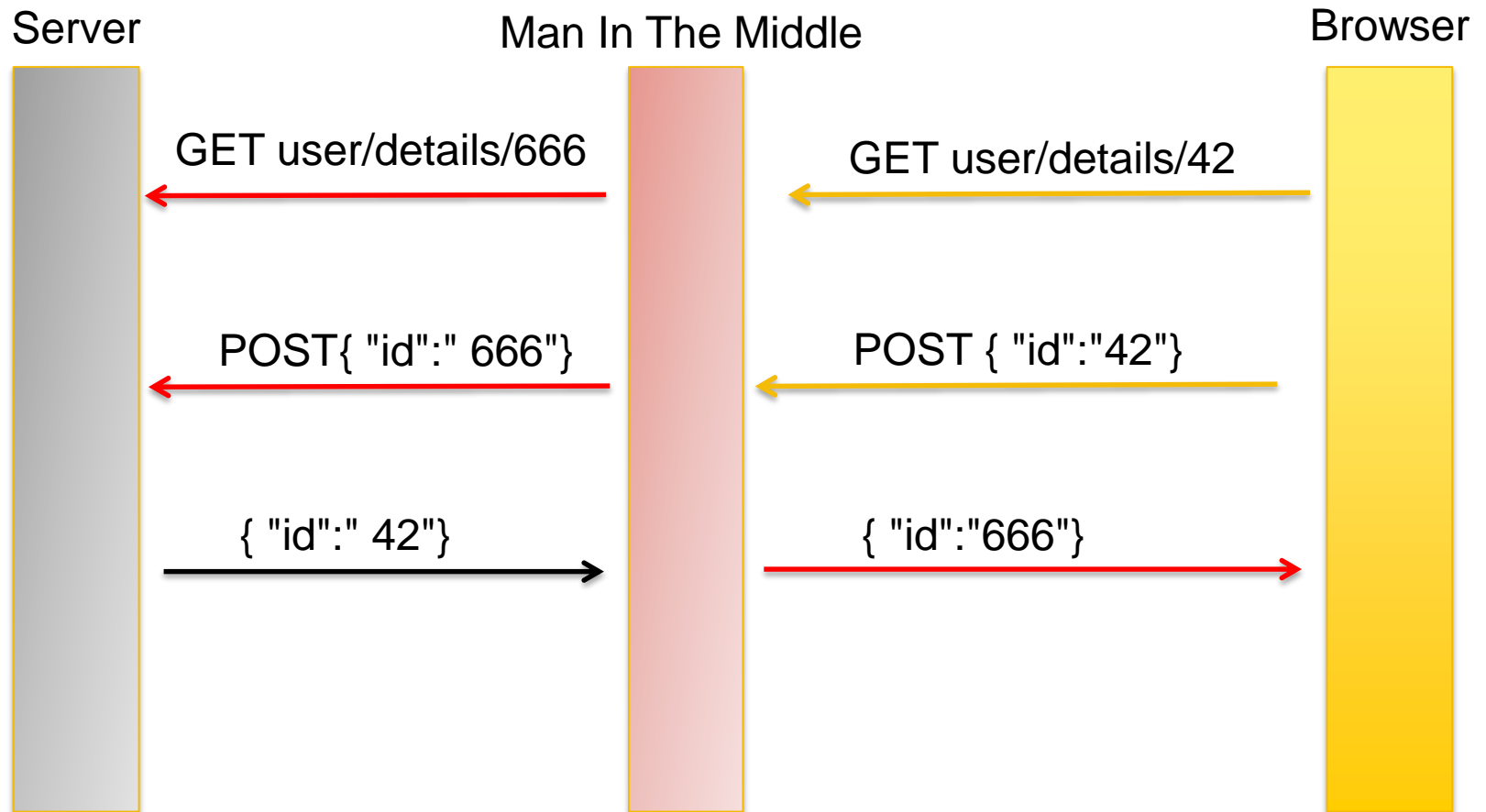
Authentication / Authorisation / Validation

Trust is good, control is better.

# Trust is good, control is better.



# Trust is good, control is better.

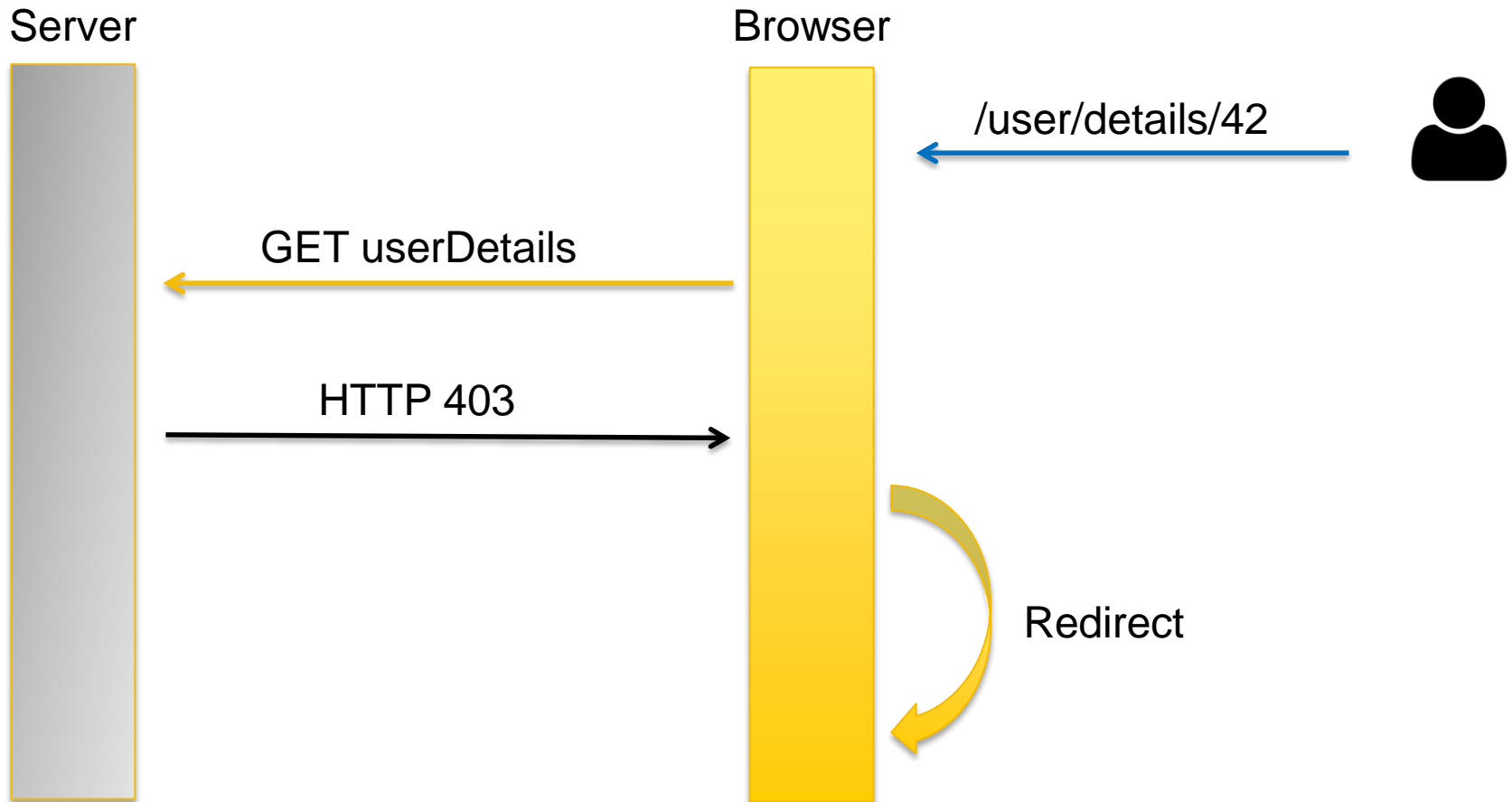




# Trust is good, control is better.

- **Frontend checks for**
  - Usability / User experience
  - HTTP errors
    - 400 – Bad Request
    - 401 – Unauthorized
    - 403 – Forbidden
- **Backend checks for**
  - Data validation
  - Authentication
  - Authorisation

# Authorisation



# Authentication

**There are actually several options for the authentication. You have to define which one best suits your needs.**

# Authentication

## Login page

- **Directly in our single page application**
- **Hosted by a third-party Authentication provider such as Auth0**
- **Reverse proxy / SSO (E.g. Nevis)**

# Authentication

## Token based

The frontend & backend applications exchanges an auth token.

## Several different ways to sent the token

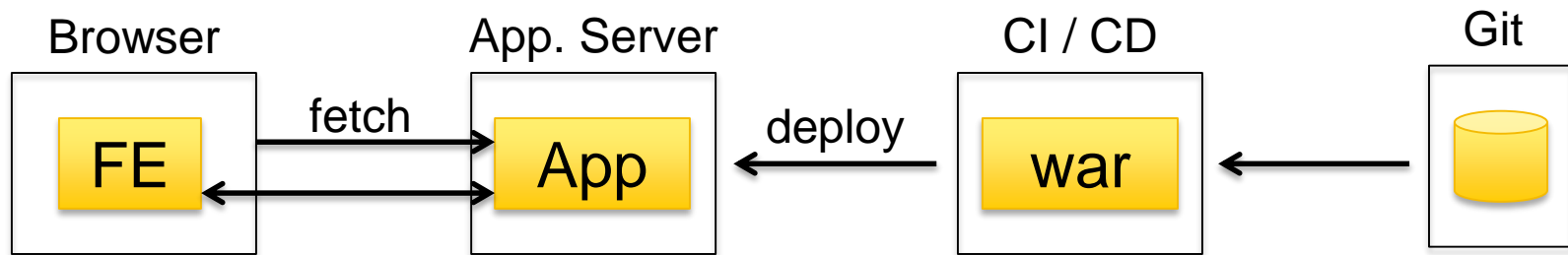
- Cookie
- Request Body
- HTTP Header

# Authentication



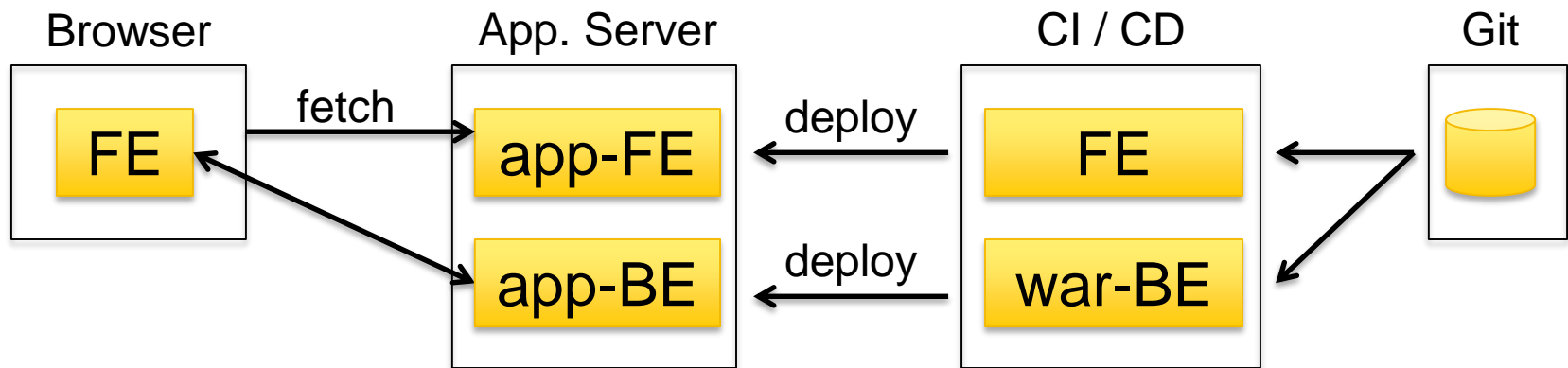
# Version-control, packaging & deployment

# Version-control, packaging & deployment

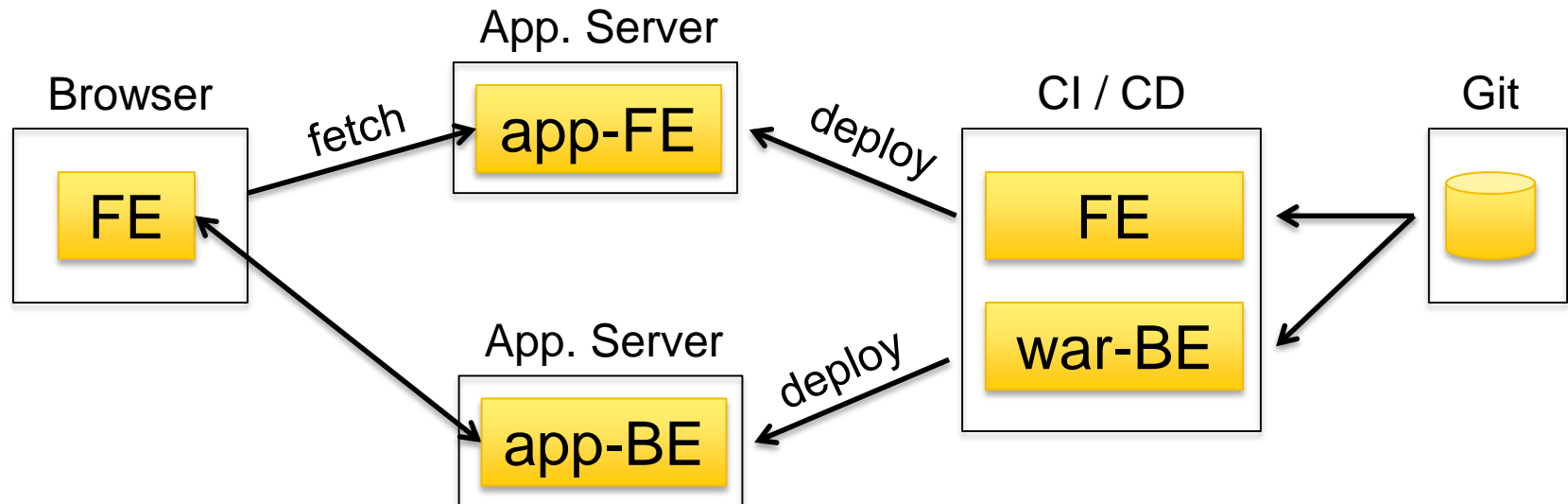




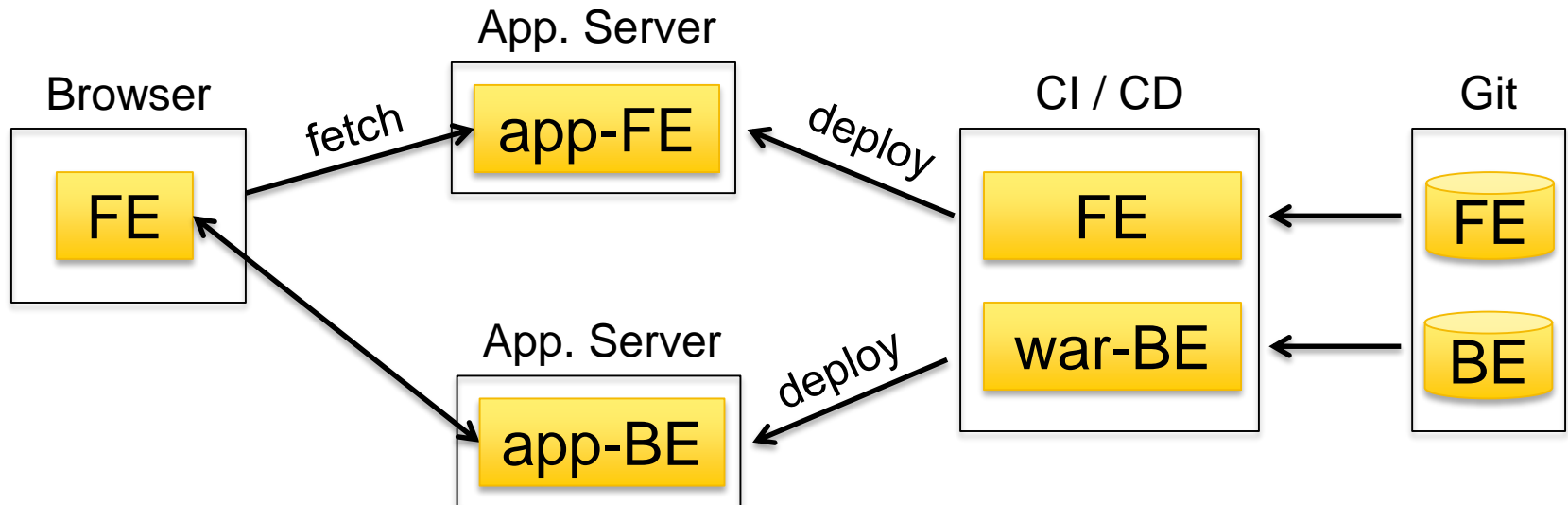
# Version-control, packaging & deployment



# Version-control, packaging & deployment



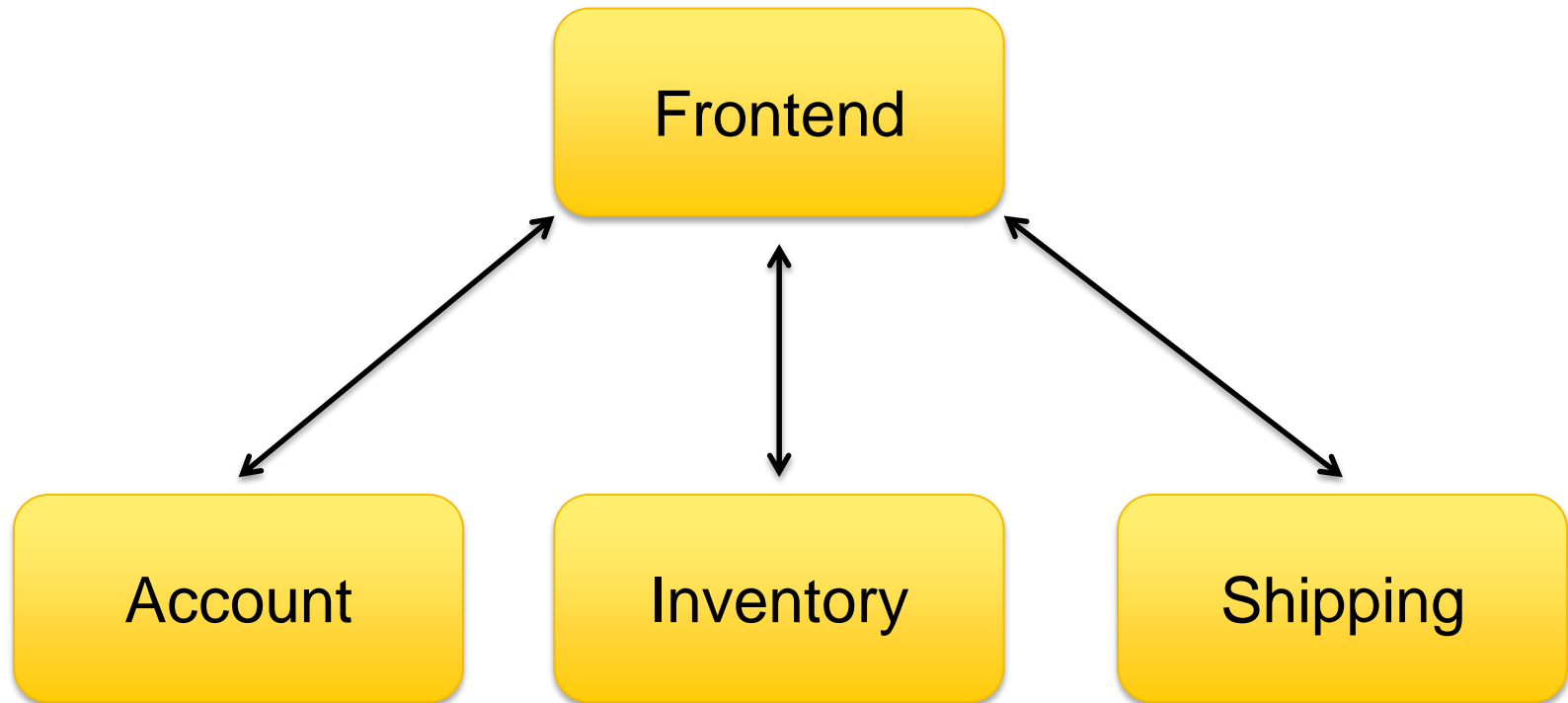
# Version-control, packaging & deployment



# Microservices

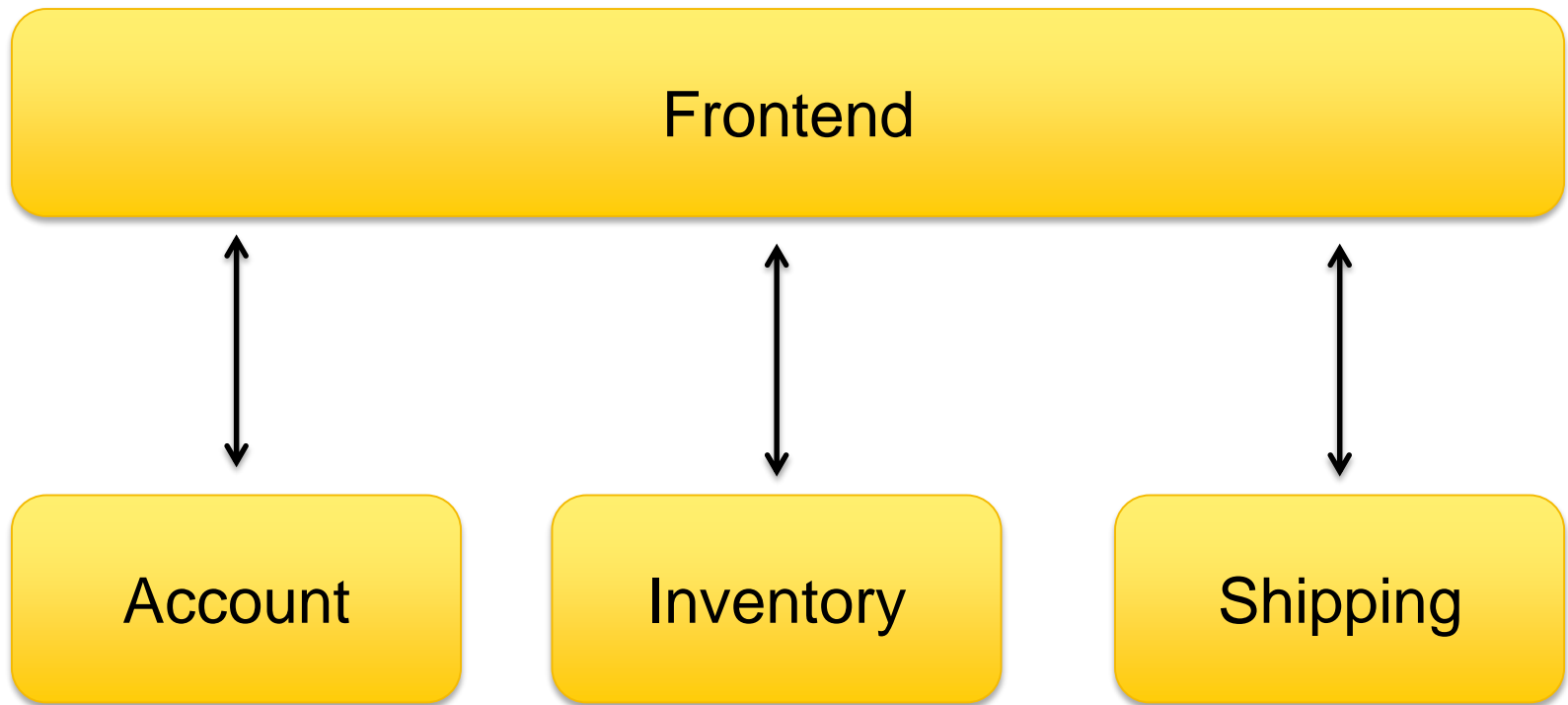
# Microservices

## E-commerce

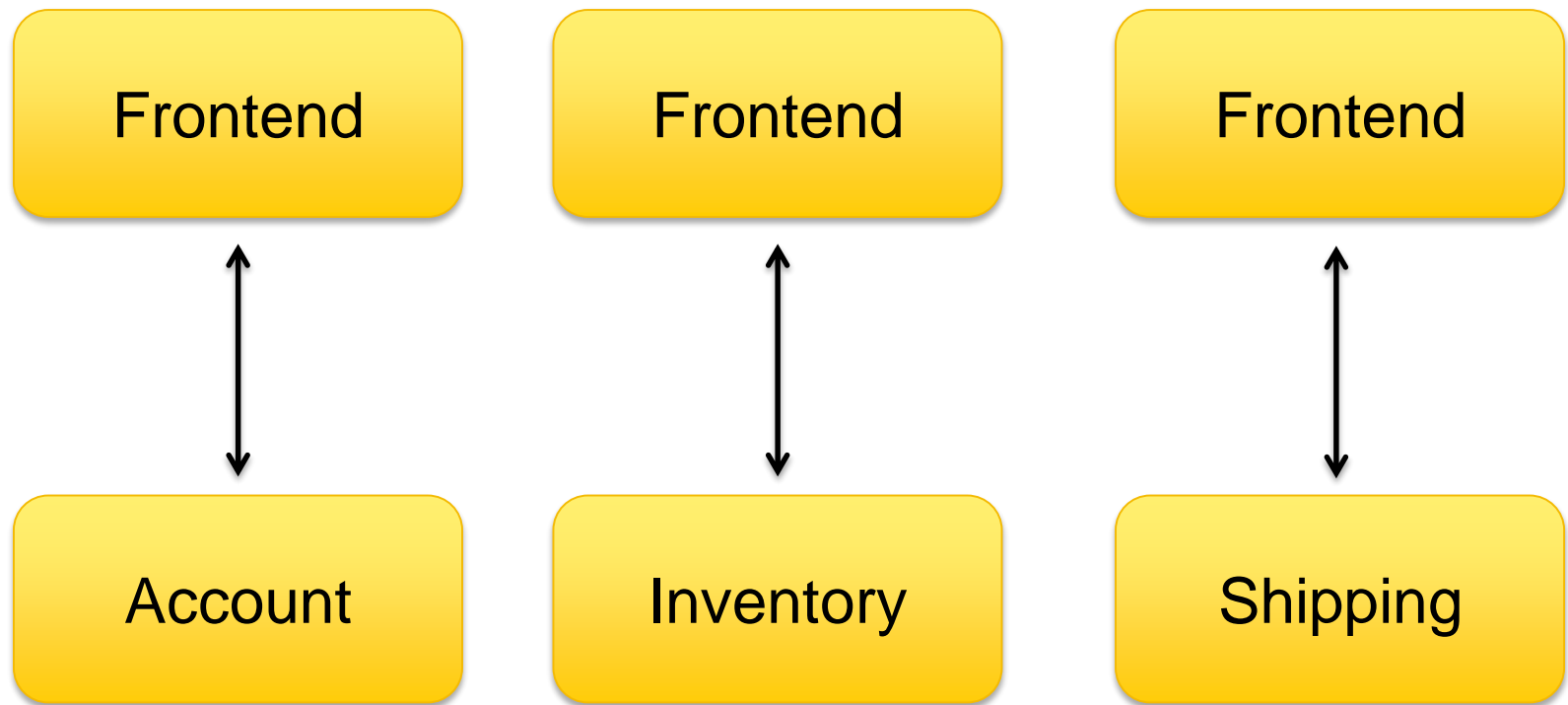


# Microservices

## Frontend monolith!



# Micro frontends



# Micro frontends – Organisation

Team  
Account

Team  
Inventory

Team  
Shipping



# Micro frontends – Advantages

- **Self-contained**
- **Independent delivery**
- **Technology Agnostic**

# Micro frontends – Challenges

- **Shared information**
- **UX**

# Micro frontends – Shared information

## How to share some common information?

- Who is logged-in?
- Selected language

# Micro frontends – Shared information

- **Through the backend**
- **HTML5 Web Storage**
- **Single Sign On (SSO)**

# Micro frontends – UX

**How to create a seamless and consistent UI experience?**

- **Applications must implement the same CI/CD**
- **Sharing common parts. E.g. header & footer**
- **Iframes**
- **UI composition**
- **Web components**

# Micro frontends

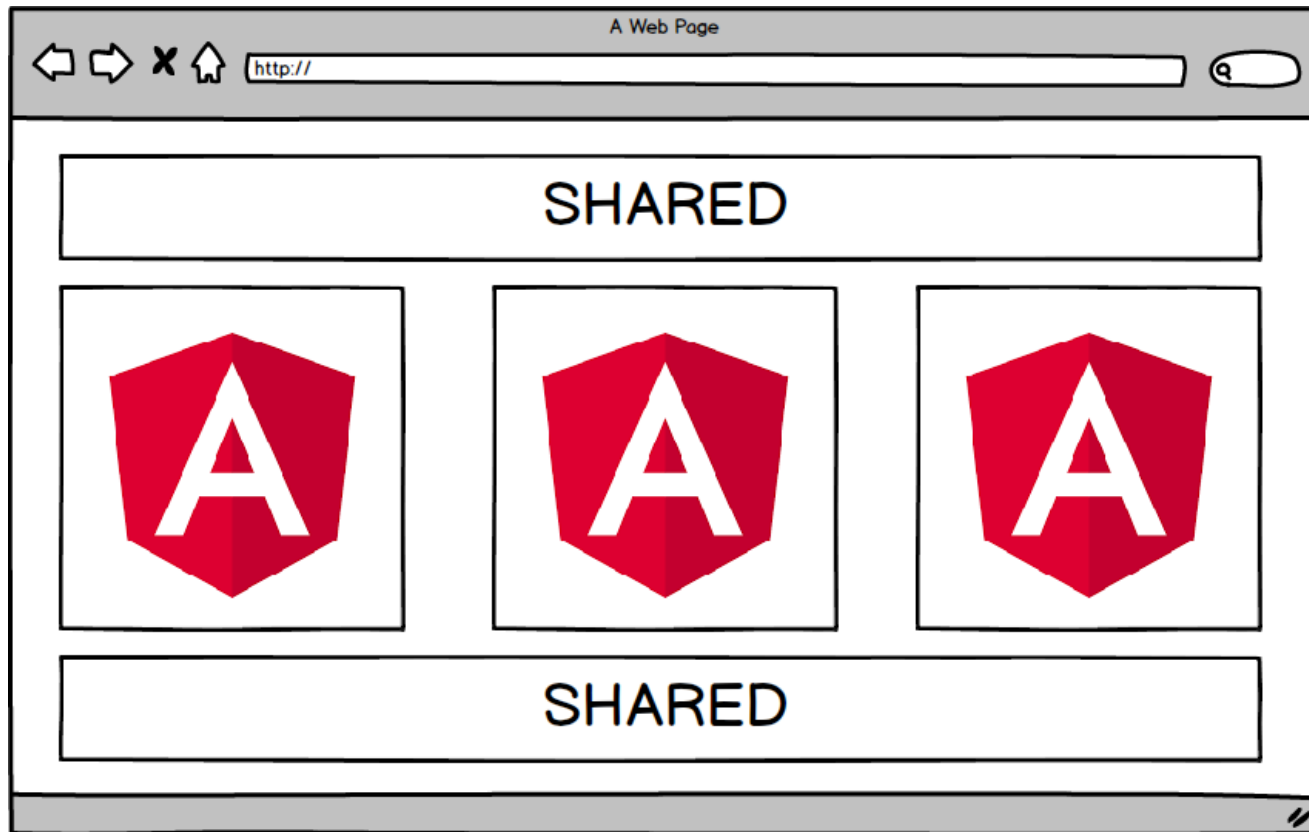
## Web components

- Set of web platform APIs
- Create new custom, reusable, encapsulated HTML tags to use in web pages and web apps.
- Work across modern browsers, and can be used with any JavaScript library or framework that works with HTML.

<https://www.webcomponents.org/>

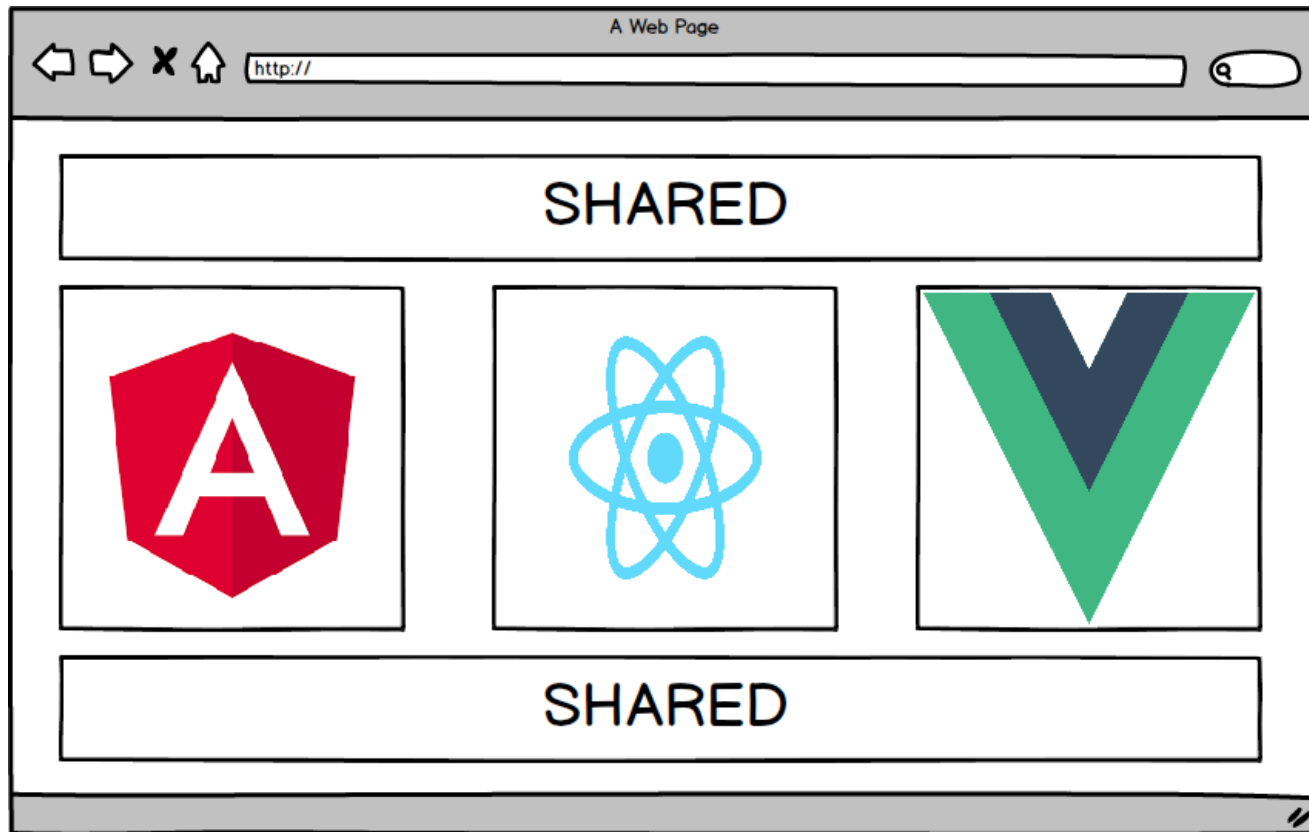
# Micro frontends

## Web components



# Micro frontends

## Web components





Dependencies / Lifecycle

Maintenance

# npm – Node.js package manager

## Node.js

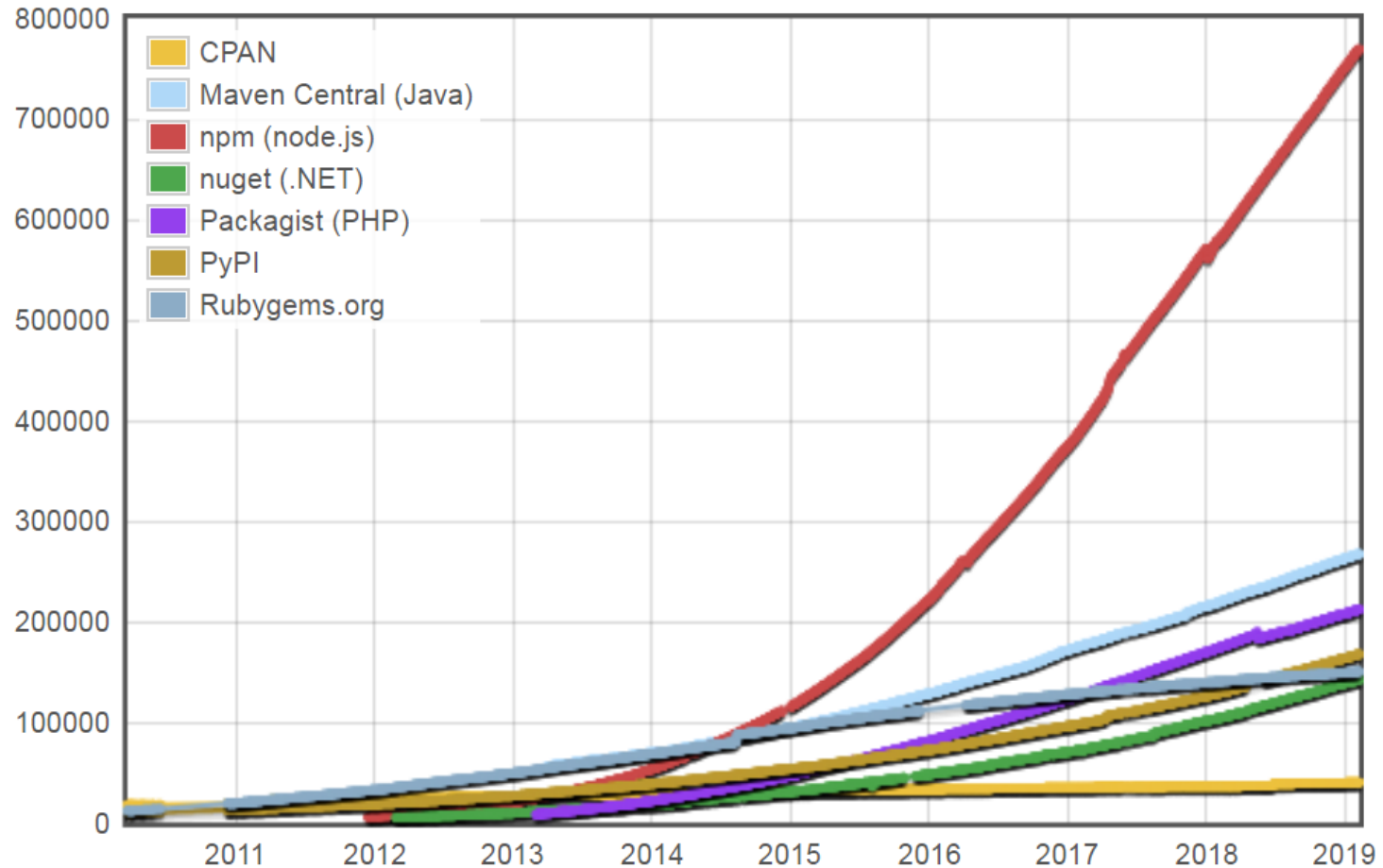
- open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser

## npm – Node.js package manager

- Package manager for JavaScript
- Default package manager for Node.js
- World's largest software registry

# npm – Node.js package manager

**Module Counts** [www.modulecounts.com](http://www.modulecounts.com)



# Semantic Versioning – SemVer

- **npm use Semantic Versioning**
  - <https://semver.org/>
  - <https://docs.npmjs.com/files/package.json>
- MAJOR.MINOR.PATCH
- **EXCEPTION:** Major version zero (0.y.z) is for initial development. Anything may change at any time. The public API should not be considered stable.

# npm dependencies

```
"dependencies": {  
  "rxjs": "6.3.3" // Must match version exactly  
}
```

```
"dependencies": {  
  "rxjs": "^6.3.3" // Compatible with version  
}
```

```
"dependencies": {  
  "rxjs": "~6.3.3" // Approximately equivalent to version  
}
```

# npm dependencies

```
"dependencies": {  
  "rxjs": "6.3.x" // 6.3.0, 6.3.1, ... but not 6.4.x or 7.x.y  
}
```

```
"dependencies": {  
  "foo": "1.0.0 - 2.9999.9999",  
  "bar": ">=1.0.2 <2.1.2",  
  "baz": ">1.0.2 <=2.3.4",  
  "qux": "<1.0.0 || >=2.3.1 <2.4.5 || >=2.5.2 <3.0.0",  
  "asd": "http://asdf.com/asdf.tar.gz",  
  "lat": "latest",  
  "dyl": "file:../dyl"  
}
```

# Maintenance – Angular

## Release frequency

Expect the following release cycle:

- A major release every 6 months
- 1-3 minor releases for each major release
- A patch release almost every week

# Maintenance – Angular

## Release schedule

DATE	STABLE RELEASE	COMPATIBILITY
March/April 2019	8.0.0	^7.0.0
September/October 2019	9.0.0	^8.0.0



# Maintenance – Angular

## Support policy and schedule

VERSION	STATUS	RELEASED	ACTIVE ENDS	LTS ENDS
^7.0.0	Active	Oct 18, 2018	Apr 18, 2019	Apr 18, 2020
^6.0.0	LTS	May 3, 2018	Nov 3, 2018	Nov 3, 2019
^5.0.0	LTS	Nov 1, 2017	May 1, 2018	May 1, 2019

# Maintenance – Angular

## What's new with Angular 8

- Differential Loading of Modern JavaScript
- Angular Router Backwards Compatibility
- TypeScript 3.3
- Template parser will no longer auto-correct "invalid" HTML
- ...

Why should I use, or not, a SPA?

# Why should I use, or not, a SPA?

## Pros

- Resources: Calculation shared between browser and server
- Fast, but...
- Rich, Complex User Interface Requirements -> UX, responsive
- Microservice
- Progressive Web Application (PWA)

# Why should I use, or not, a SPA?

## Cons

- ... take time for initial load.
- Minimal Client-Side Application Behavior -> Overkill
- Browser needs javascript (?)
- Knowledge of javascript / typescript (?)

Radically better web experience

# Progressive Web Application



# PWA – Progressive Web Application

**Web application takes advantage of new features supported by modern browsers.**

- **Characteristics**

- Progressive
- Responsive
- Connectivity independent
- App-like
- Installable
- ...

<https://developers.google.com/web/progressive-web-apps/>

[https://en.wikipedia.org/wiki/Progressive\\_web\\_applications](https://en.wikipedia.org/wiki/Progressive_web_applications)

# SPA – challenges

- **Client/Server code partitioning**
- **Browser history**
- **Search engines / analytics**
- **Speed of initial load**
- **Security**
  
- **Technology moves quickly**
- **Learn something new**



SPA – Is resistance futile ?



ciril.saner@glue.ch

Thank you for your feedback

# Part 2 - Development with Angular

# Angular 8

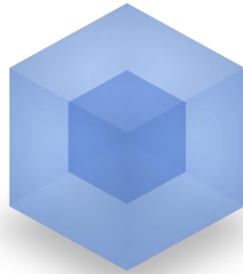
# Frontend development

# What are we going to see / use ?

TypeScript



Jasmine



webpack  
MODULE BUNDLER



**glue**

Address book application

What is our goal ?



# Visual Studio Code

# Visual Studio Code

- **Lightweight but powerful source code editor**
- **Developed by Microsoft**
- **Runs on Windows, macOS and Linux**
- **Rich ecosystem of extensions**
- **MIT License**

<https://code.visualstudio.com/>

# TypeScript

# TypeScript – Configuration

- **tsconfig.json** – TypeScript compiler configuration
- **typings** – TypeScript declaration files

# TypeScript – Variable declarations

## ***var* declarations**

- scope of current execution context

## ***let* declarations**

- block scope local variable

## ***const* declarations**

- for everything you don't have to "replace"

# TypeScript – Variable declarations

```
for (var i = 0; i < 5; i++) {  
  setTimeout(  
    function() {console.log(i)}, 100);  
}  
// result: 5 5 5 5 5
```

```
for (let i = 0; i < 5; i++) {  
  setTimeout(  
    function() {console.log(i)}, 100);  
}  
// result: 1 2 3 4 5
```

# TypeScript – Basis types

**boolean**

**number**

- floating point
- decimal, hexadecimal, binary, octal literals

**string**

**Array**

# TypeScript – Interfaces

**Type-checking focuses on the shape that values have**

**Duck typing: If it walks like a duck and it quacks like a duck, then it must be a duck.**

**In TypeScript, interfaces fill the role of naming these types.**



# TypeScript – Classes

- **Object-oriented class-based approach**
- **Starting with ECMAScript 2015**
- **Inheritance**

# TypeScript – Operations & functions

## Arrow functions

- An arrow function expression is a syntactically compact alternative to a regular function expression

# TypeScript – Operations & functions

## Classic function

```
const numbers: number[] = [1, 2, 3, 4, 5];  
const even = numbers.filter(  
  function (value: number) {  
    return value % 2 === 0;  
  }  
);
```

# TypeScript – Operations & functions

## One line arrow function

```
const numbers: number[] = [1, 2, 3, 4, 5];  
const even = numbers.filter(  
  value => value % 2 === 0  
);
```

# TypeScript – Operations & functions

## Block arrow function

```
const numbers: number[] = [1, 2, 3, 4, 5];  
const even = numbers.filter(  
  value => {  
    return value % 2 === 0;  
  }  
);
```

# TypeScript – Decorators

**Provide a way to add annotations and a meta-programming syntax for class declarations and members.**

**Angular rely heavily on decorators:**

- @NgModule
- @Component
- @Injectable
- ...

**We will see more about decorators when coding...**

# Angular Big Picture

- **Opinionated Framework**
- **How does Angular work?**
  - Modules
  - Components
  - Services
  - ...

# Angular – modules

- **Angular NgModule != JavaScript modules**
- **These are complementary module systems that you can use together to write your apps.**



# JavaScript – Modules

- In JavaScript each file is a module and all objects defined in the file belong to that module.
- The module declares some objects to be public by marking them with the export key word.
- Other JavaScript modules use import statements to access public objects from other modules.

# Angular – NgModule

- **Containers for a cohesive block of code dedicated to an application domain.**
- **Can contain components, service providers, and other code files whose scope is defined by the containing NgModule.**
- **Can import functionality that is exported from other NgModules, and export selected functionality for use by other NgModules.**

# Angular – components

- **Controls a patch of screen called a view.**
- **You define a component's application logic, what it does to support the view, inside a class.**
- **The class interacts with the view through an API of properties and methods.**

# Angular – components – data binding

```
<li>{{contact.name}}</li>  
<app-contact-detail  
  [contact]="selectedContact">  
</app-contact-detail>  
<li (click)="selectContact(contact)"></li>
```

- {{contact.name}} – displays the value
- [contact] – property binding
- (click) – event binding

# Angular – services

- **Broad category encompassing any value, function, or feature that an app needs.**
- **Class with a narrow, well-defined purpose. It should do something specific and do it well.**
- **Angular distinguishes components from services to increase modularity and reusability. Makes your component classes lean and efficient.**

# Angular – dependency injection

- DI is wired into the Angular framework
- To define a class as a service, use the `@Injectable()` decorator to provide the metadata that allows Angular to inject it into a component as a dependency.

# Angular – dependency injection

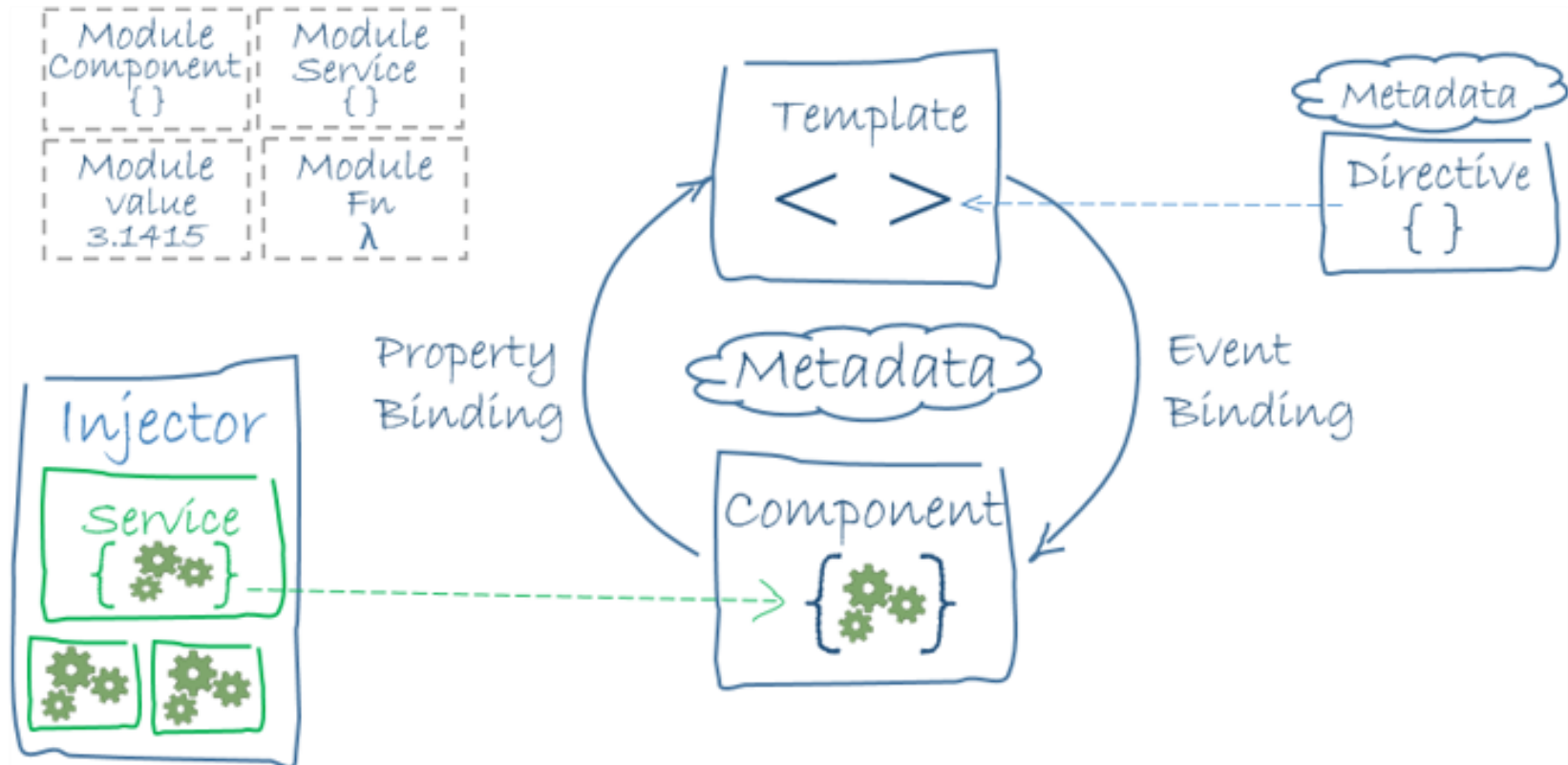
- **The injector is the main mechanism.**
- **Angular creates an application-wide injector for you during the bootstrap process.**
- **An injector creates dependencies, and maintains a container of dependency instances that it reuses if possible.**

# Angular – dependency injection

- **A provider is an object that tells an injector how to obtain or create a dependency.**
- **For any dependency that you need in your app, you must register a provider with the app's injector, so that the injector can use the provider to create new instances.**
- **For a service, the provider is typically the service class itself.**



# Angular Big Picture



# Angular CLI

- **Command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications**
- **We will learn to use it by developing our application...**
- <https://cli.angular.io/>
- <https://angular.io/cli>

Initial project setup

# Exercise 00

# Angular CLI – Create a new app

## Install Angular CLI

- Install it globally using npm

## Create the app

- Create an app called address-book with `ng new`
- prompts:
  - ? Would you like to add Angular routing? Yes
  - ? Which stylesheet format would you like to use? CSS

## Ready to run

- Run the app with `ng serve`

# Angular CLI – Create a new app

Welcome to address-book!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

# Angular CLI – Create a new app

## What have we got?

- New workspace
- Initial skeleton app project
- End-to-end test project
- Configuration files.

Heavy?

# Angular CLI – Create a new app

## **Exercise: Perform a prod build**

1. Find the commands of the angular cli
2. What are the differences between prod and dev mode?

# Dependencies Management

- **npm is used by default with Angular**
- **package.json**
- Dependencies vs devDependencies
- ncu
- ng update



Contact overview

Address book

Add contact table

## Exercise 01

# Address book – Contact overview

1. Clean up app.component.html
2. Change the title of the app
3. Add a table displaying: firstname, lastname, phone & email

Problems?

Address book – Contact overview

Add overview component

## Exercise 02

# Address book – Contact overview

## Create a new component

1. Create a component called overview with ng generate
2. Move the table into the HTML of the component
3. Add the selector of the component in the `app.component.html`
4. What have we got?

Add overview module

## Exercise 03

# Address book – Contact overview

## Create a new module

1. Create a new module called overview with ng generate
2. Move the declaration of OverviewComponent
3. Import the new module
4. Export the OverviewComponent
5. What have we got?



Add routing to overview

## Exercise 04

# Address book – Contact overview

Remove export of OverviewComponent

Remove <app-overview>

Add navigation

```
const routes: Routes = [  
  { path: '', redirectTo: '/overview', pathMatch: 'full' },  
  { path: 'overview', component: OverviewComponent }  
];
```

Problems?

Address book – Contact overview

Add contact service

## Exercice 05

# Address book – Contact overview

## Create a new service

1. Create a new service called contact with ng generate
2. Import and Inject the service in the OverviewComponent
3. Create a getContactList method in the service and call it on init in the component
4. What have we got?

JSON Server

Address book

# JSON Server

1. Install JSON server: `npm install -g json-server`
2. Create json file
3. Run server: `json-server --watch db.json`

More infos: <https://github.com/typicode/json-server>

Fetching contact list

## Exercise 06



# Address book – Contact overview

## Fetch the data from the backend

1. `import { HttpClientModule } from '@angular/common/http'`
2. Add HttpClient to the ContactService
3. Fetch the data

# The RxJS library

# The RxJS library

**RxJS (Reactive Extensions for JavaScript) is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code (RxJS Docs).**

**Reactive programming is an asynchronous programming paradigm concerned with data streams and the propagation of change (Wikipedia).**

# The RxJS library

**The essential concepts in RxJS which solve async event management are:**

- **Observable**: represents the idea of an invokable collection of future values or events.
  - **Subscription**: represents the execution of an Observable, is primarily useful for cancelling the execution.
- 
- More: Subject , Observer, Operators, Schedulers

# The RxJS library

## More information:

- [https://en.wikipedia.org/wiki/Reactive\\_programming](https://en.wikipedia.org/wiki/Reactive_programming)
- <https://rxjs-dev.firebaseapp.com/>
- <https://angular.io/guide/rx-library>

Display the contact list

## Exercise 07

# Address book – Contact overview

## Display the data

1. Store the data in the component
2. Iterate in the table
3. Display the values

Add contact type

## Exercise 08



# Address book – Contact overview

## Add type definition

1. Create an interface named contact with ng generate
2. Rename the file contact.type.ts
3. Use Contact as type

# Debugging

# Debugging – Recommended tools

- **Your browser (DevTools)**
- **Augury: extension for Chrome & Firefox**
- **VS Code: Debugger for Chrome**
- **Your brain**
- **Rubber duck**

# Debugging – Browser (DevTools)

- **Network analysis**
- **Console outputs**
- **Breakpoints**

# Debugging – Augury

- <https://augury.rangle.io/>
- Application inspection tool for Angular
- Provides insight into the application structure

# Debugging – VS Code

Some developers prefer to debug inside of their IDE...

- Debugger for Chrome extension required
- Configuration in *lauch.json* in *.vscode*

# Debugging – Brain

**No tools can replace your brain**

# Debugging – Rubber duck

1. **Beg, borrow, steal, buy, fabricate or otherwise obtain a rubber duck (bathtub variety).**
2. **Place rubber duck on desk and inform it you are just going to go over some code with it, if that's all right.**
3. **Explain to the duck what your code is supposed to do, and then go into detail and explain your code line by line.**
4. **At some point you will tell the duck what you are doing next and then realise that that is not in fact what you are actually doing. The duck will sit there serenely, happy in the knowledge that it has helped you on your way.**

<https://rubberduckdebugging.com/>

[https://en.wikipedia.org/wiki/Rubber\\_duck\\_debugging](https://en.wikipedia.org/wiki/Rubber_duck_debugging)



# Debugging – Tips

**Keep the DevTools of your browser open**

Problems?

Address book – Contact overview

# Address book – structure

**If your company, team, project doesn't have a style guide, have a look at :**

- <https://angular.io/guide/styleguide>

Create shared

## Exercise 09

# Address book – structure

- **Move the contact type & ContactService in a folder called shared**

## **Challenge:**

- **Declare the path to shared in tsconfig.json**

## tsconfig.json – paths

In tsconfig.json file add following declaration under compilerOptions

```
"paths": {  
  "shared/*": [  
    "src/app/shared/*"  
  ],  
}
```

# tsconfig.json – paths

Instead of

```
import { Contact } from '../../../shared/contact.type';
```

We get

```
import { Contact } from 'shared/contact.type';
```

# Code quality

## How can we check the code quality?

- Linters
- Tests

## What is a linter?

- Tool that analyzes source code to flag programming errors, bugs, stylistic errors, and suspicious constructs.

## Which linters do I need?

- ng lint
- TSLint

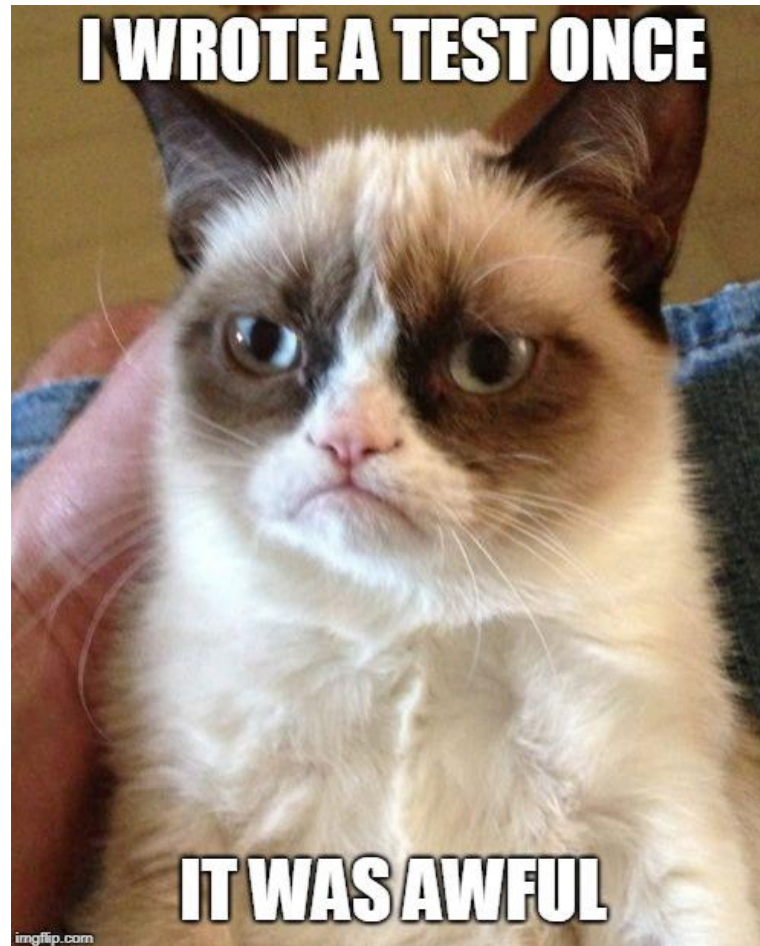


Examples

Linters

# Testing

# Testing



# Testing

**What kind of tests do we have?**

- **Unit tests**
- **Integration tests**
- **Performance tests**
- **Automated browser tests**
- **Accessibililty tests**
- **User tests**

# Testing – Unit Tests

Run the tests using ng. Add new line in the OverviewComponent.

What happens?

Angular uses karma and jasmine by default

# Testing – karma / Jasmine

Karma v3.1.4 - connected

Chrome 72.0.3626 (Windows 7.0.0) is idle

Jasmine 2.99.0 finished in 0.369s

5 specs, 2 failures

Spec List | Failures

overviewComponent should create

Error: StaticInjectorError(DynamicTestModule)[HttpClient]:  
StaticInjectorError(Platform: core)[HttpClient]:  
NullInjectorError: No provider for HttpClient!

Error: StaticInjectorError(DynamicTestModule)[HttpClient]:  
StaticInjectorError(Platform: core)[HttpClient]:  
NullInjectorError: No provider for HttpClient!

at NullInjector.push../node\_modules/@angular/core/fesm5/core.js.NullInjector.get (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:8894:1)  
at resolveToken (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:9139:1)  
at tryResolveToken (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:9083:1)  
at StaticInjector.push../node\_modules/@angular/core/fesm5/core.js.StaticInjector.get (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:8980:1)  
at resolveToken (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:9139:1)  
at tryResolveToken (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:9083:1)  
at StaticInjector.push../node\_modules/@angular/core/fesm5/core.js.StaticInjector.get (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:8980:1)  
at resolveNgModuleDep (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:21120:1)  
at NgModuleRef.push../node\_modules/@angular/core/fesm5/core.js.NgModuleRef.\_get (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:21809:1)  
at injectTestComponent (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:1733:1)

Expected undefined to be truthy.

Error: Expected undefined to be truthy.

at stack (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:2455:17)  
at buildExpectationResult (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:2425:14)  
at Spec.expectationResultFactory (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:901:18)  
at Spec.addExpectationResult (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:524:34)  
at Expectation.addExpectationResult (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:845:21)  
at Expectation.toBeTruthy (http://localhost:9876/absolute:/Users/saner/Documents/projects/address-book/node\_modules/jasmine-core/11b/jasmine-core/jasmine.js:70b1eaf7a13cae32191eadea482cf96ae41fc22b:2369:12)  
at UserContext.<anonymous> (http://localhost:9876/\_karma\_webpack\_/webpack/src/app/overview/overview.component.spec.ts:23:23)  
at ZoneDelegate../node\_modules/zone.js/dist/zone.js.ZoneDelegate.invoke (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/zone.js/dist/zone.js:391:1)  
at ProxyZoneSpec.push../node\_modules/zone.js/dist/zone-testing.js.ProxyZoneSpec.onInvoke (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/zone.js/dist/zone-testing.js:289:1)  
at ZoneDelegate../node\_modules/zone.js/dist/zone.js.ZoneDelegate.invoke (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/zone.js/dist/zone.js:390:1)

ContactService should be created

Error: StaticInjectorError(DynamicTestModule)[HttpClient]:  
StaticInjectorError(Platform: core)[HttpClient]:  
NullInjectorError: No provider for HttpClient!

Error: StaticInjectorError(DynamicTestModule)[HttpClient]:  
StaticInjectorError(Platform: core)[HttpClient]:  
NullInjectorError: No provider for HttpClient!

at NullInjector.push../node\_modules/@angular/core/fesm5/core.js.NullInjector.get (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:8894:1)  
at resolveToken (http://localhost:9876/\_karma\_webpack\_/webpack/node\_modules/@angular/core/fesm5/core.js:9139:1)

# Testing – karma / Jasmine

## Jasmine

- Behavior-driven testing framework

## Karma

- Test runner
- Execute JavaScript code in multiple real browsers.
- Created by the AngularJS team
- Configured in *karma.conf.ts*

# Testing – karma / Jasmine

## Some tips

- Use *xdescribe* to exclude a test suite
- Use *xit* to exclude a test case
- Use *fdescribe* to run only focused test suites
- Use *fit* to run only focused test cases



Testing

## Exercise 10

# Testing – Fixing the tests

Import the *HttpClientTestingModule*

Testing

# Exercise 11

# Testing – *OverviewComponent* add tests

**Test: *ContactService* is called only 1 time on init.**

- Create a *createSpyObj* with jasmine
- Provide the mocked service
- Mock the *getContactList* method
- Create the test
- Reset calls after each tests

# Testing – *ContactService* add tests

## Test: *getContactList* is called with the correct URL

- Define a global *HttpTestingController*
- Create the test
- Call the *getContactList* method
- Use the *HttpTestingController* to check that the URL is correct
- Use the *HttpTestingController* to check that the call was a GET
- Check, with *HttpTestingController*, after each test that no calls are open

# Testing – Drawbacks

- **All tests are rerun on modification**
- **Performance**

# Testing – Alternative to karma / Jasmine

## Jest

- Testing framework from Facebook
- Test runner is included
- Performance is better
- Rerun only changed tests.

Problems?

Address book – Contact overview



# Material Design components for Angular

## Angular Material

# Material Design

- **Design language**
- **Developed by Google in 2014.**

# Angular Material

- **Implementation of Google's Material Design Specification**
- **Provides high-quality UI components built with Angular and TypeScript**

<https://material.angular.io>

# Alternatives

- **Bootstrap**
- **PrimeNg**

Install Angular Material

## Exercise 12

# Angular Material

## Installation

- Install `@angular/material` using `ng add`
- Prompts
  - ? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink
  - ? Set up HammerJS for gesture recognition? Yes
  - ? Set up browser animations for Angular Material? Yes

# Angular Material

## What does `ng add @angular/material`?

- Ensure project dependencies are placed in `package.json`
- Enable the `BrowserAnimationsModule` on your app module
- Add either a prebuilt theme or a custom theme
- Add Roboto fonts to your `index.html`
- Add the Material Icon font to your `index.html`
- Add global styles
  - Remove margins from body
  - Set height: 100% on `html` and `body`
  - Make Roboto the default font of your app
- Install and import `hammerjs` for gesture support in your project

Use Angular Material table

## Exercise 13



# Angular Material

## Replace the overview table

- Using a material table
- Look at the documentation:  
<https://material.angular.io/components/table/api>
- Don't forget to check the tests!

Use Angular Material toolbar

## Exercise 14

# Angular Material

## Replace the title with a toolbar

- Using a material toolbar
- Look at the documentation:  
<https://material.angular.io/components/toolbar/api>
- Don't forget to check the tests!

Use Angular Material card

## Exercise 15

# Angular Material

## Put the content of the overview in a card

- Using a material card
- Look at the documentation:  
<https://material.angular.io/components/card/api>
- Add some CSS for a better layout
- Don't forget to check the tests!

Problems?

Address book – Contact overview

# Angular – Pipes

## Change in json-server:

- Phone number without formatting
- Phone number with strange formatting
- Restart the server

**What is the result?**

# Angular – Pipes

**Displaying the raw values received from the backend is not the always the best idea for good user experience...**



# Angular – Pipes

**Angular pipes allow us to write display-value transformations that you can declare in your HTML.**

# Angular – Pipes

Angular comes with a stock of built-in pipes

Such as DatePipe, UpperCasePipe...

They are all available for use in any template.

**More info:**

- <https://angular.io/guide/pipes#pipes>
- <https://angular.io/api?type=pipe>

Add phone format pipe

## Exercise 16

# Angular – Pipes

## Create a pipe to format the phone number

1. Create a *pipe* named phone-format with ng generate
2. Use the format XXX XXX XX XX
3. Use it in the *OverviewComponent*
4. Write tests

Where do you declare the pipe?

New functionality: add contact

Address book

Add contact page

## Exercise 17

# Address book – add contact

**Create a separate page to add a new entry in the address book.**

1. Create a new module AddContact
2. Create a new component AddContact
3. Create a new route to the new page

Navigation to contact page

## Exercise 18



# Address book – add contact

**Add a button in the overview to navigate to the add contact page.**

- Use a material raised button
- Put it in the card actions
- Perform the navigation on click inside the component with the Router service
- Check the tests!

# Address book – add contact

**We need to use a form, but ...**

**... Angular provides two different approaches to handling user input through forms.**

<https://angular.io/guide/forms-overview>

# Angular – forms

## Reactive forms

- More robust: they're more scalable, reusable, and testable.

## Template-driven forms

- Useful for adding a simple form to an app, such as an email list signup form.

<https://angular.io/guide/forms-overview>

# Angular – forms

	REACTIVE	TEMPLATE-DRIVEN
Setup (form model)	More explicit, created in component class	Less explicit, created by directives
Data model	Structured	Unstructured
Predictability	Synchronous	Asynchronous
Form validation	Functions	Directives
Mutability	Immutable	Mutable
Scalability	Low-level API access	Abstraction on top of APIs

Add contact form

## Exercise 19

# Address book – add contact

**Create a reactive form in the add contact page.**

## **Use**

- FormBuilder
- Angular Material form field
- Angular Material input
- A submit button
- The same design as the overview
- Check your tests

**Simply log the value of the form on submit.**

Save new contact

## Exercice 20

# Address book – add contact

**Send the entered values to the backend.**

- Convert the form value to a Contact
- POST request to <http://localhost:3000/contacts>
- Check your tests!



Form validation

## Exercise 21

# Address book – add contact

## Add some validation to the form

- All fields are required
- Use also *Validators.email* for the email
- Disable the submit button if the form is invalid
- Show corresponding error messages
- Check your tests!

Improvements add contact

## Exercise 22

# Address book – add contact

**Improve the behaviour of the application after adding a contact.**

- Navigate back to the overview after save
- Display a success message for five seconds with a Material Snackbar
- Add a cancel button
- Check your tests!

Delete contact

## Exercice 23

# Address book – delete contact

## Add a delete action on the overview

- Use a Material Icon Button
- Create a Material Dialog to confirm the deletion
- DELETE request to <http://localhost:3000/contacts/{id}>
- Refresh overview after delete
- Show confirmation Snackbar
- Check your tests!

Edit contact

## Exercise 24

# Address book – edit contact

## Create a page to edit a contact

- Create a new page for editing a contact
- Put an edit icon next to the delete in the overview
- Pass the ID of the contact to update to the edit component with the routing
- Display the prefilled edit form
- Send the updated data with PUT request to <http://localhost:3000/contacts/{id}>
- Check your tests!





ciril.saner@glue.ch

Thank you for your feedback