

NÁSKOK  
DÍKY  
ZNALOSTEM

PROFINIT

# Full Stack není jen frontend a backend

Štěpán Poljak a Jonáš Klimeš

Březen 2022

# Cíl

- › Motivace ke studiu různých konceptů a technologií
  - Přehled vybraných konceptů pro inspiraci
  - Propojení teorie a praxe
  - Fullstack Developer podle Profinitu

# Kdo je Full Stack developer?

- › Přesná definice neexistuje
- › Nejčastěji: dokáže postavit celou aplikaci
- › V našem případě webovou
  - Full Stack Web Developer



# Kdo je Full Stack developer?

- › Nemusí nutně vyvíjet FE i BE současně
- › Závisí na typu projektu, organizaci týmu, ochotě „přepínat“
  - Projekt od projektu, úkol od úkolu
  - Řešení cross-cutting úkolů / problémů
  - Zastupitelnost v případě potřeby
- › Může se více specializovat na jednu oblast, ale orientuje se v celé širší SW vývoje
  - Rozumí v „dostatečné“ nebo velké míře i těm ostatním
  - V závislosti na zkušenostech
- › Kde začíná a končí FE / BE? A je to všechno? Co je dobré znát?

# Koncepty a technologie

# Full Stack koncepty a principy



Architektura

Metodika vývoje

Bezpečnost

Komunikace

Operations

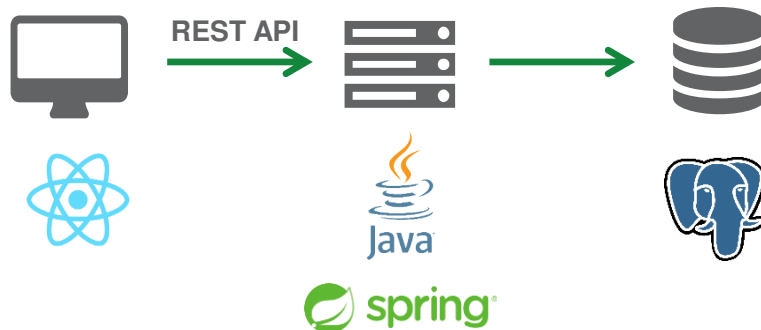
Automatizace

- › Proč jsou koncepty a principy důležité?
  - Technologie se rychle mění, koncepty a principy platí napříč technologiemi
- › SW inženýr orientující se v celé šíři SW vývoje
- › Rozumí tomu, co dělá a dokáže řešit i komplexní problémy

# Moderní trend: JS FE + BE + REST API

## > FE + BE + REST API

- Přehledné, loosely-coupled, škálovatelné
- Možno napojit další FE či aplikace



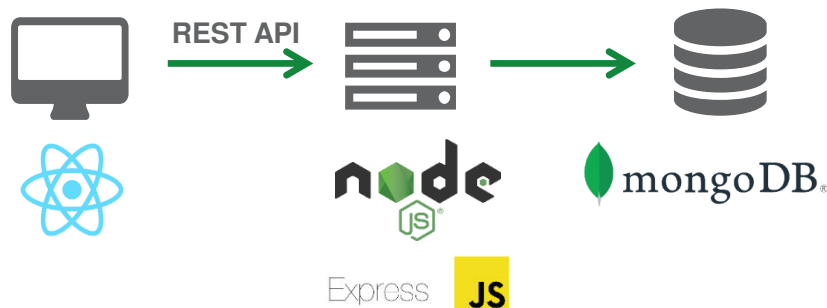
## > FE: JS frameworky:

- React, Angular, Vue...
- Interaktivní, flexibilní, rychlý vývoj a zaučení

## > BE: Mnoho možností

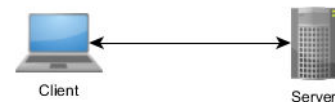
- Java, C#, JS, Python, Ruby...

## > DB: SQL / NoSQL



# Alternativa: Programovací abstrakce

- › Kód píšeme přímo na serveru, framework poskytuje abstrakci
- › FE se generuje na serveru, framework zajišťuje synchronizaci na klientovi
- › Komunikace FE/BE je v rámci jedné JVM
- › Zpočátku jednoduché – prostě volám metodu
- › Nutno důsledně oddělovat FE/BE
  - Jinak spaghetti kód
- › Customizace může být problém
- › Příklad: Vaadin, RAP ...



<https://vaadin.com/docs/v8/framework/introduction/intro-walkthrough.html>

```
import com.vaadin.server.VaadinRequest;
import com.vaadin.ui.Label;
import com.vaadin.ui.UI;

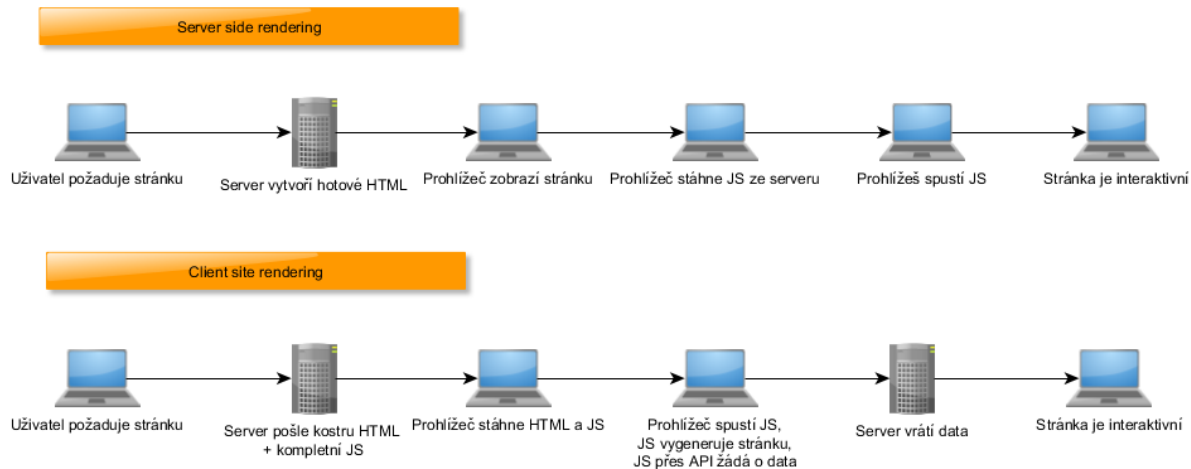
@Title("My UI")
public class HelloWorld extends UI {
    @Override
    protected void init(VaadinRequest request) {
        // Create the content root layout for the UI
        VerticalLayout content = new VerticalLayout();
        setContent(content);

        // Display the greeting
        content.addComponent(new Label("Hello World!"));

        // Have a clickable button
        content.addComponent(new Button("Push Me!",
            click -> Notification.show("Pushed!")));
    }
}
```



# Rendering



- › Server side rendering
  - JavaServer Faces, Vaadin\*, Ruby on Rails, PHP, ...
- › Client side rendering
  - React\*, Angular\*, ...
- › Oba přístupy mají výhody i nevýhody, možno kombinovat

# Budoucnost?

- › Vznikalo postupně, není to jediný přístup
- › Stále vznikají nové přístupy a frameworky
- › Příklad: **WebAssembly**
  - Standard pro spouštění binárního kódu v prohlížeči
  - Rychlejší než JS, mohl by být rychlý jako nativní kód
  - Teoreticky bude možné použít libovolný jazyk, často C/C++

- › Příklad: **Blazor**
- › Místo JS přímo C#
- › Překlad do WebAssembly

```
<h1>Counter</h1>
<p>Current count: @currentCount</p>
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
```

```
@code {
    private int currentCount = 0;

    private void IncrementCount()
    {
        currentCount++;
    }
}
```



# Technologie se rychle mění...

- › Technologie se velmi rychle mění (na FE speciálně)
- › Koncepty – naštěstí – většinou - platí obecně
- › Příklady:
  - Synchronní / asynchronní volání
  - CORS (volání API)
  - State management, úložiště
  - Session, autentizace
- › Ale i koncepty se vyvíjí a mění
  - Přejít od session ke stateless
  - Vývoj asynchronního volání
  - Atd.



# Zpracování asynchronních událostí

- › Motivace: Dlouho trvající / paralelní akce nezasekne aplikaci
- › Příklad:
  - Komponenty na stránce
  - Stažení souboru na pozadí
  - Paralelní volání více služeb

# Zpracování asynchronních událostí

## › Callbacks

```
const getUser = cb => {  
  setTimeout(() => {  
    cb({ name: 'Bob' })  
  }, 2000)  
}  
getUser(user => {  
  console.log(user.name);  
})
```



```
const checkAuth = cb => {  
  setTimeout(() => {  
    cb({ isAuth: true })  
  }, 2000)  
}  
const getUser = (authInfo, cb) => {  
  if (!authInfo.isAuth) {  
    cb(null)  
    return  
  }  
  setTimeout(() => {  
    cb({ name: 'Bob' })  
  }, 2000)  
}  
checkAuth(authInfo => {  
  getUser(authInfo, user => {  
    console.log(user.name)  
  })  
})
```



```
checkAuth(authInfo => {  
  getUser(authInfo, user => {  
    getUserDetails(user, userDetails => {  
      getReport(userDetails, report => {  
        ...  
      })  
    })  
  })  
})
```

# Zpracování asynchronních události

## › Callbacks – callback hell (pyramid of doom)

```
function iIds(startAt, showSessionRoot, iNewNmVal, endActionsVal, iStringVal, seqProp, htmlEncodeRegEx) {
  if (SbUtil.dateDisplayType == 'relative') {
    iRange();
  } else {
    iSelActionType();
  }
  iStringVal = notifyWindowTab;
  startAt = addSessionConfigs.sbRange();
  showSessionRoot = addSessionConfigs.e1HiddenVal();
  var headerDataPrevious = function(tabArray, iNm) {
    iPredicateVal.SBDB.deferCurrentSessionNotifyVal(function(evalOutMatchedTabUrlVal) {
      if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
        iPredicateVal.SBDB.normalizeTabList(function(appMsg) {
          if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
            iPredicateVal.SBDB.detailTxt(function(evalOrientationVal) {
              if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                iPredicateVal.SBDB.neutralizeWindowFocus(function(iTokenAddedCallback) {
                  if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                    iPredicateVal.SBDB.evalSessionConfig2(function(sessionNm) {
                      if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                        iPredicateVal.SBDB.iWindow2TabIdx(function(iURLStringVal) {
                          if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                            iPredicateVal.SBDB.idx7Val(undefined, iStringVal, function(getWindowIndex) {
                              if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                addTabList(getWindowIndex.rows, iStringVal, showSessionRoot && showSessionRoot.length > 0 ? sh
                                  if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                    evalSAllowLogging(tabArray, iStringVal, showSessionRoot && showSessionRoot.length > 0
                                      if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                        BrowserAPI.getAllWindowsAndTabs(function(iSessionVal) {
                                          if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                            SbUtil.currentSessionSrc(iSessionVal, undefined, function(initCurrent
                                              if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                addSessionConfigs.render(matchText(iSessionVal, iStringVal, e
                                                  id: -13,
                                                  unfilteredWindowCount: initCurrentSessionCache,
                                                  filteredWindowCount: iCtrl,
                                                  unfilteredTabCount: parseTabConfig,
                                                  filteredTabCount: evalRegisterValue5Val
                                                } : [], cacheSessionWindow, evalRateActionQualifier, undefine
                                                  if (seqProp) {
                                                    seqProp();
                                                  }
                                                }
                                              }
                                            }
                                          }
                                        }
                                      }
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}, showSessionRoot && showSessionRoot.length > 0 ? showSessionRoot : startAt ? [startAt] : []);
}
```



... a nesmíme zapomenout i na ošetření chyb.

# Zpracování asynchronních událostí

## › Promises

```
checkAuth(authInfo => {  
  getUser(authInfo, user => {  
    getUserDetails(user, userDetails => {  
      getReport(userDetails, report => {  
        ...  
      })  
    })  
  })  
})
```



```
checkAuth()  
  .then(authInfo => getUser(authInfo))  
  .then(user => getUserDetails(user))  
  .then(userDetails => getReport(userDetails))  
  ...  
  .catch(error => ...)
```

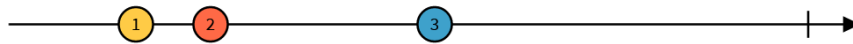
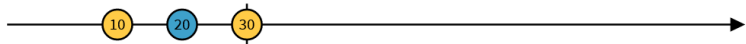


```
try {  
  const authInfo = await checkAuth();  
  const user = await getUser(authInfo);  
  const userDetails = await getUserDetails(user);  
  const report = await getReport(userDetails);  
  ...  
} catch (error) {  
  ...  
}
```

# Reaktivní programování

- › Reakce na stream dat
- › Podobný koncept i na backendu (např. RxJava, Rx.NET)

```
Observable.from([10, 20, 30]).delayWhen(x => timer(x))
```

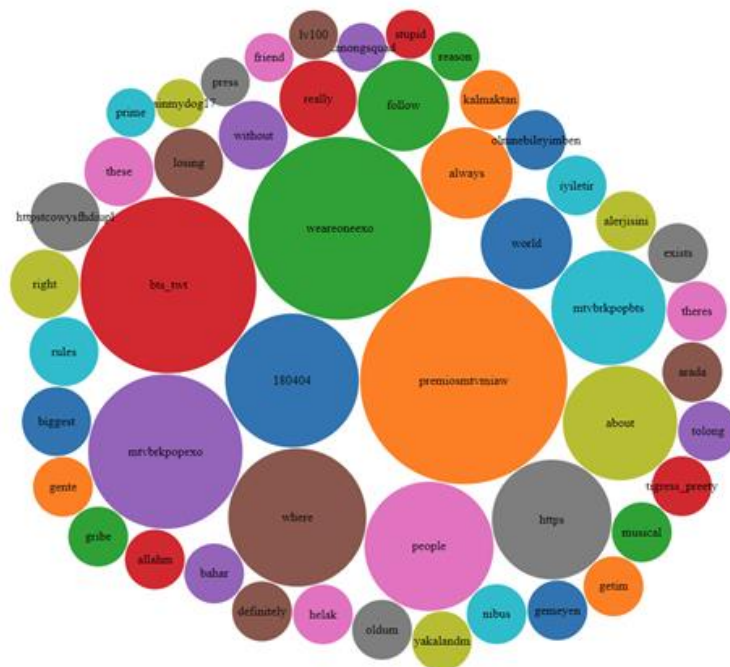


```
map(x => 10 * x)
```





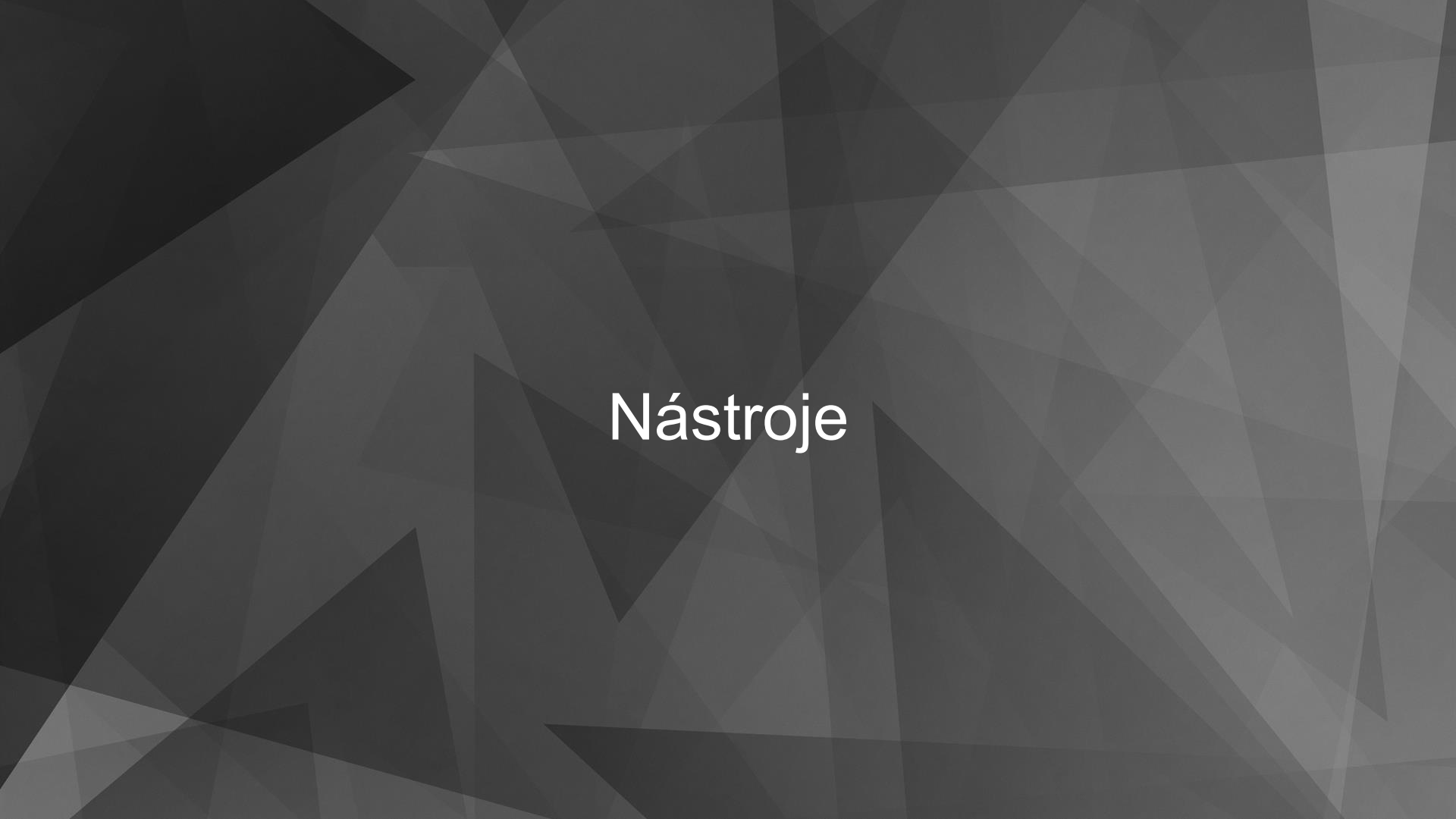
# Ukázka - twitter bubbles



180404: 15 premioumruiaw: 36 weareoneexo: 28 bea\_rxt: 26 mrvtbkpoxes: 20 where: 16 people: 14 https: 12 about: 11 mrvtbkpoxes: 11 world: 7 always: 7 follow: 7 really: 5 without: 4 loung: 4 these: 4 httpscowdydada: 4 right: 4 rules: 4 biggest: 4 gentle: 3 gribe: 3 allahm: 3 bahar: 3 definitely: 3 belak: 3 oldum: 3 yakalandm: 3 muba: 3 gemeyen: 3 getum: 3 musical: 3 agres\_peety: 3 tolong: 3 arada: 3 theres: 3 excits: 3 alerjissani: 3 ryaletir: 3 olmabileyimben: 3 kalmaktan: 3 reason: 3 stupid: 3 kancengquid: 2 N100: 2 friend: 2 prese: 2 trainmydog17: 2 prime: 2

180404: 15 premioumruiaw: 36 weareoneexo: 28 bea\_rxt: 26 mrvtbkpoxes: 20 where: 16 people: 14 https: 12 about: 11 mrvtbkpoxes: 11 world: 7 always: 7 follow: 7 really: 5 without: 4 loung: 4 these: 4 httpscowdydada: 4 right: 4 rules: 4 biggest: 4 gentle: 3 gribe: 3 allahm: 3 bahar: 3 definitely: 3 belak: 3 oldum: 3 yakalandm: 3 muba: 3 gemeyen: 3 getum: 3 musical: 3 agres\_peety: 3 tolong: 3 arada: 3 theres: 3 excits: 3 alerjissani: 3 ryaletir: 3 olmabileyimben: 3 kalmaktan: 3 reason: 3 stupid: 3 kancengquid: 2 N100: 2 friend: 2 prese: 2 trainmydog17: 2 prime: 2

180404: 15 premioumruiaw: 36 weareoneexo: 28 bea\_rxt: 26 mrvtbkpoxes: 20 where: 16 people: 14 https: 12 about: 11 mrvtbkpoxes: 11 world: 7 always: 7 follow: 7 really: 5 without: 4 loung: 4 these: 4 httpscowdydada: 4 right: 4 rules: 4 biggest: 4 gentle: 3 gribe: 3 allahm: 3 bahar: 3 definitely: 3 belak: 3 oldum: 3 yakalandm: 3 muba: 3 gemeyen: 3 getum: 3 musical: 3 agres\_peety: 3 tolong: 3 arada: 3 theres: 3 excits: 3 alerjissani: 3 ryaletir: 3 olmabileyimben: 3 kalmaktan: 3 reason: 3 stupid: 3 kancengquid: 2 N100: 2 friend: 2 prese: 2 trainmydog17: 2 prime: 2



Nástroje

# Frameworky, knihovny, nástroje

- › Jak implementuji autentizaci/autorizaci?
  - › Jak udělám REST API?
  - › Jak zpracuju CSV?
  - › Jak dostanu data do databáze?
  - › Jak vytvořím nový projekt?
- 
- › Není cílem znát všechny frameworky světa
  - › Ale vědět, jaké možnosti existují, umět je vyhodnotit a využít
  - › Příklad pro nový projekt: Spring Initializr + Create React App



LIQUIBASE



# Nástroje – Spring



**Project**

☒ Maven Project

☐ Gradle Project

**Language**

☒ Java

☐ Kotlin

☐ Groovy

**Spring Boot**

☐ 2.5.1 (SNAPSHOT)

☒ 2.5.0

☐ 2.4.7 (SNAPSHOT)

☐ 2.4.6

☐ 2.3.12 (SNAPSHOT)

☐ 2.3.11

## Project Metadata

**Group**

**Artifact**

**Name**

**Description**

**Package name**

**Packaging** ☒ Jar ☐ War

**Java** ☐ 16 ☒ 11 ☐ 8

## Dependencies

ADD ... CTRL + B

No dependency selected

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

Web, Security, JPA, Actuator, Devtools...

Press Ctrl for multiple adds

## DEVELOPER TOOLS

### Spring Native [Experimental]

Incubating support for compiling Spring applications to native executables using the GraalVM native-image compiler.

### Spring Boot DevTools

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

### Lombok

Java annotation library which helps to reduce boilerplate code.

### Spring Configuration Processor

Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).

## WEB

### Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

### Spring Reactive Web

Build reactive web applications with Spring WebFlux and Netty.

### Rest Repositories

Exposing Spring Data repositories over REST via Spring Data REST.

### Spring Session

Provides an API and implementations for managing user session information.

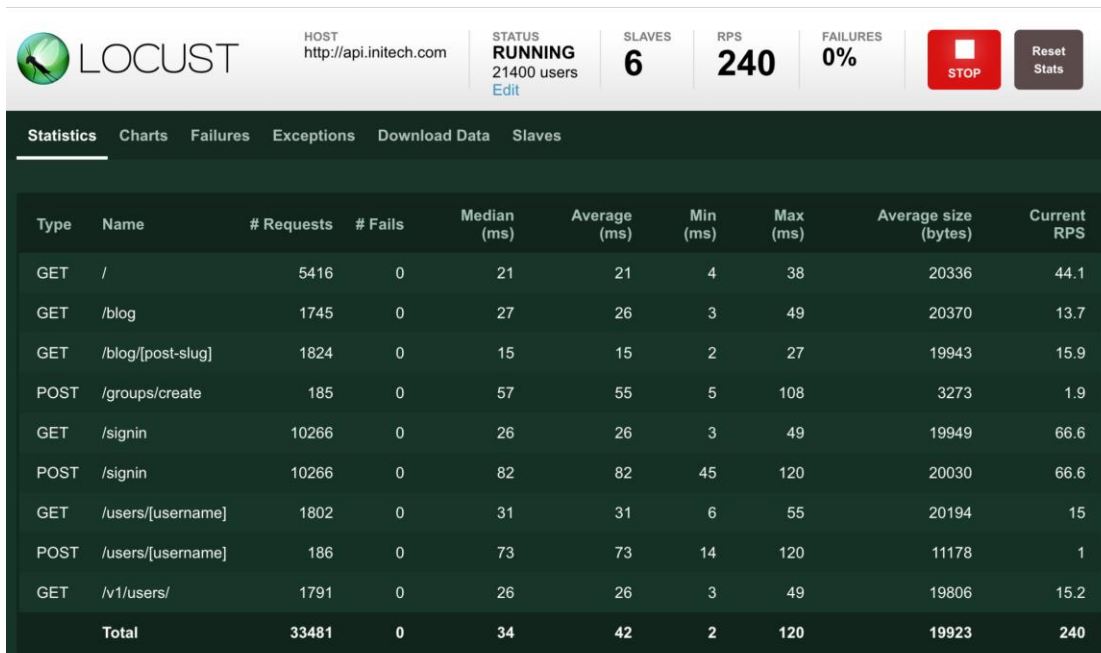
### Rest Repositories HAL Explorer

Browsing Spring Data REST repositories in your browser.

### Spring HATEOAS

# Nástroje – Locust

- › Performance HTTP testy jednoduše
- › Alternativa například jMeter pro další protokoly



```
from locust import HttpUser, between, task

class WebsiteUser(HttpUser):
    wait_time = between(5, 15)

    def on_start(self):
        self.client.post("/login", {
            "username": "test_user",
            "password": ""
        })

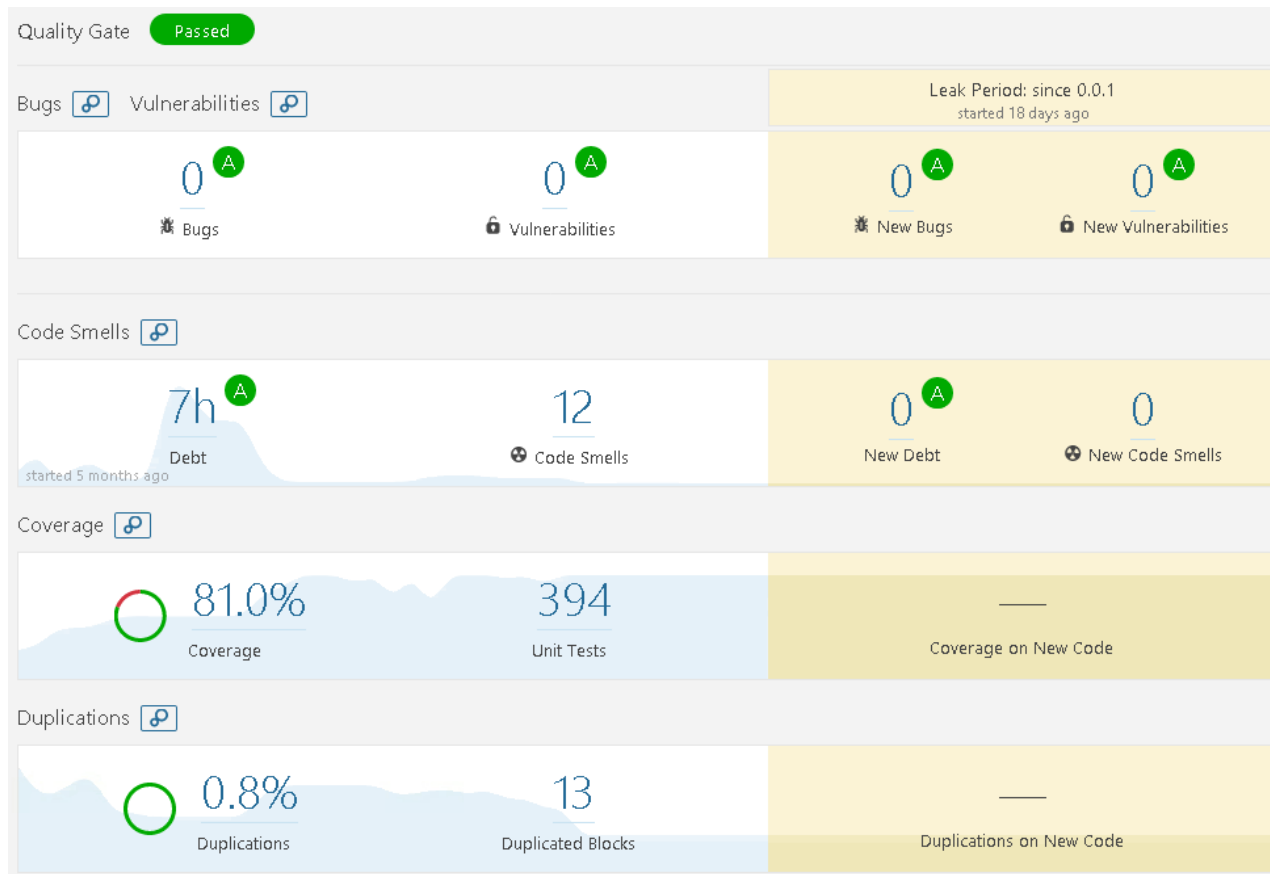
    @task
    def index(self):
        self.client.get("/")
        self.client.get("/static/assets.js")

    @task
    def about(self):
        self.client.get("/about/")
```

# Nástroje - SonarQube

- › Nástroj pro analýzu kvality kódu
  - Statická analýza
  - Testy a test coverage
  - Bezpečnostní kontroly
- › Podporuje
  - Java, C#, JavaScript, SQL
  - A mnoho dalšího ...
- › Doporučení
  - Pouštět analýzu pravidelně
  - Kontinuálně řešit nálezy
  - Nenechat nahromadit technický dluh
  - Nezapomínat na code review

# SonarQube - Přehled



# SonarQube – Issue

```
public static String readFromPreferences(Context context, String preferenceName, String defaultValue) {  
  1 SharedPreferences sharedPreferences = null;  
  try {  
    sharedPreferences = context.getSharedPreferences(PREF_FILE_NAME, Context.MODE_PRIVATE);  
  } catch (Exception e) {  
    e.printStackTrace();  
    new ExceptionHandler(e, "NavigationDrawerFragment - readFromPreferences()");  
  }  
  return 2 sharedPreferences.getString(preferenceName, defaultValue);  
}
```

A "NullPointerException" could be thrown; "sharedPreferences" is nullable here. ...

11 months ago ▾ L147 🔗

🐛 Bug 🚨 Major 🔵 Open ▾ Not assigned ▾ 10min effort [Comment](#)

🔗 cert, cwe ▾

```
}
```

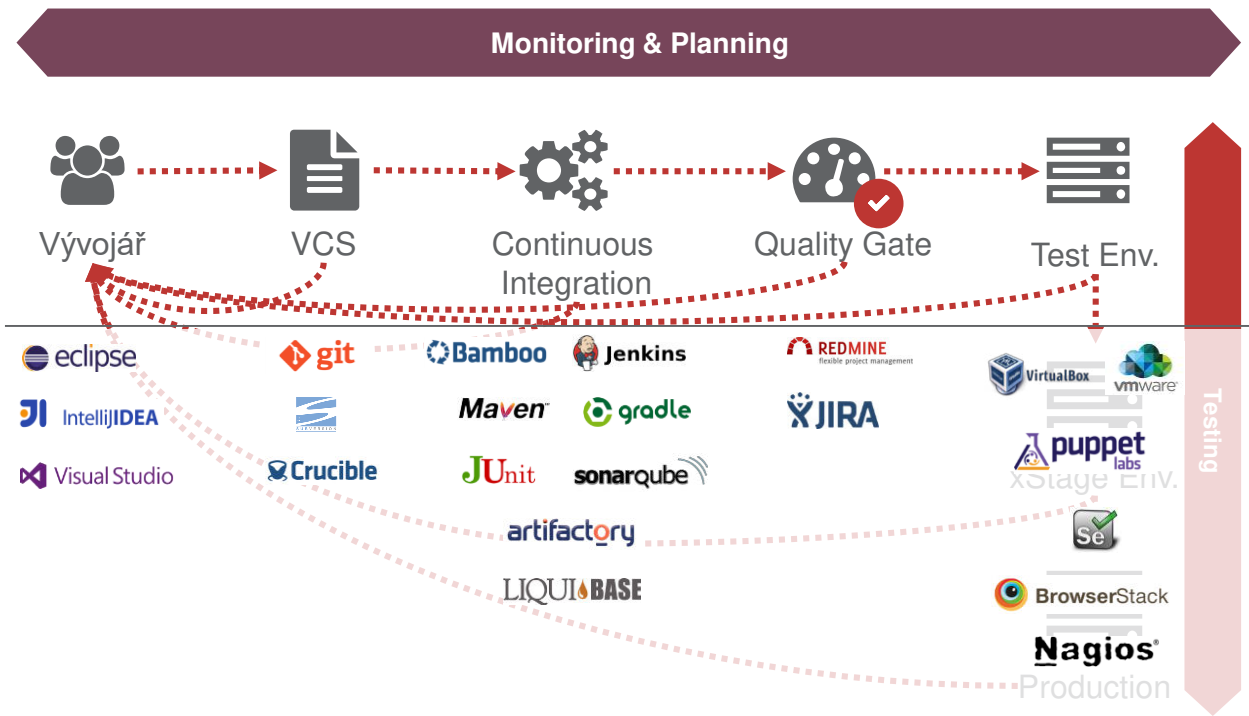


## Příklad - logování

- › Příklad: Voláme WS službu pro tisk objednávky. Co bychom radši viděli v případě chyby?
  - *Communication error.*
  - *Error while sending documents to printservice, response code 400.*
  - *Error while sending documents to printservice, response code 400, missing input parameter 'address'.*
  - *[RQ 1]: Error while sending documents to printservice for order O2, response code 400, missing input parameter 'address'.*
- › Smysluplné, podrobné zprávy, kontext, stacktrace
  - Jaký nastal problém? Proč nastal problém?

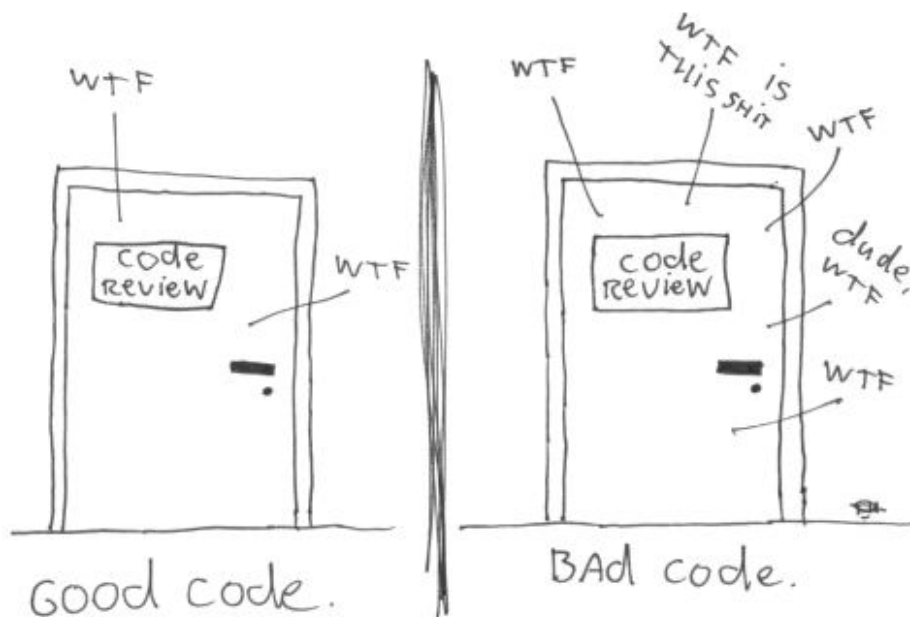


# Proces a tým



- › Odpovědnost vývojáře nekončí commitem
  - Ověřuje po sobě, že všechny kroky proběhly v pořádku
  - Dokáže vyřešit problémy / pomoci / nasměrovat

The ONLY valid measurement  
of code quality: WTFs/minute



# Full Stack myths

- › *„Umí vše průměrně a neumí nic pořádně“*
  - Ne, zná principy a v něčem se specializuje
  - Jen trochu méně než specialista (logicky)
- › *„Full Stack je lepší než specialista nebo naopak“*
  - Ne, obojí má výhody/nevýhody a své uplatnění
- › *„Bud’ jsem nebo nejsem Full Stack“*
  - Ne, specializuji se třeba na BE, ale můžu i přepnout
  - Často vyplyne spontánně časem

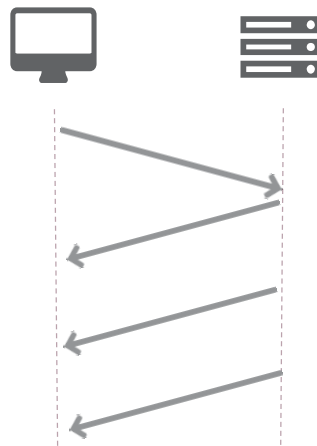
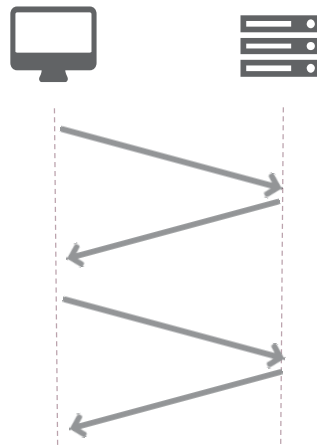
# Materiály

- › NPRG014 Koncepty moderních programovacích jazyků
- › NSWI153 Pokročilé programování webových aplikací
- › NSWI037 Programování pro Windows II
- › NSWI177 Introduction to Linux

# Komunikace a zabezpečení

# Komunikace – protokol

- › Protokol HTTP
  - request/response
  - jednosměrná
- › Příklad: Chat se zákaznickou podporou
  - Jak se zákazník dozví, že dostal novou zprávu?
- › Protokol WebSocket
  - obousměrná



# Komunikace – obsah zpráv

## › Komunikace

- REST
- SOAP
- GraphQL
- gRPC

```
{  
  "id": 5,  
  "username": „jnovak",  
  "firstName": "Jan",  
  "lastName": „Novak",  
  "email": „jan.novak@gmail.com",  
  "phone": "+420737545676"  
}
```

## › Reprezentace dat

- XML
- JSON
- Různé binární
- ...

## › REST

- Datově orientovaný
- Svázán s HTTP protokolem
  - Metody: GET, POST, PUT, DELETE
  - HTTP Status codes



SMARTBEAR  
SwaggerHub

SIGN UP

LOG IN

Servers

https://virtserver.swaggerhub.com/jonaash/Eshop/1.0.0

SwaggerHub API Auto Mocking

Authorize

USERS

Operations about user

POST /users Create a new user (user registration)

GET /users List users

GET /users/{userId} Get user by id

PUT /users/{userId} Update user

DELETE /users/{userId} Delete user

POST /users/login Logs user into the system

POST /users/logout Logs out current logged in user

items

Everything about items

orders

Access to store orders

GET

/users

List users

This can only be done by admin.

Parameters

Try it out

| Name                           | Description                                                          |
|--------------------------------|----------------------------------------------------------------------|
| email<br>string<br>(query)     | Email used for filter<br>email - Email used for filter               |
| firstName<br>string<br>(query) | First name used for filter<br>firstName - First name used for filter |
| lastName<br>string<br>(query)  | Last name used for filter<br>lastName - Last name used for filter    |

Responses

| Code | Description          | Links    |
|------|----------------------|----------|
| 200  | successful operation | No links |

# OpenAPI

- › Specifikace RESTful API
- › Lze generovat z kódu REST endpointů

- › Generování client API

- Příklad TypeScript client:

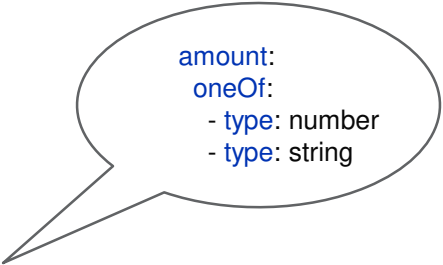
```
return this.validateApi.bankAccountValidateUsingGET({ bic, iban, stat, statId })
```

- Příklad Java client

- Problémy

- OpenAPI 3.0 může obsahovat v více typů pro jeden atribut
        - Java (nástroj pro generování Java client) nepodporuje
        - ➔ Hack – konverze na verzi 2.0 a zpět (odstraní nekompatibilní definice)
      - Kód volání se nám nelíbí a nevyhovuje nám
        - ➔ Generujeme jen model

```
/**  
 * Request parameters for bankAccountValidateUsingGET operation ...  
 * @interface ValidateApiBankAccountValidateUsingGETRequest  
 */  
export interface ValidateApiBankAccountValidateUsingGETRequest {  
  /**  
   * BIC kód banky  
   * @type {string}  
   * @memberof ValidateApiBankAccountValidateUsingGET  
   */  
  readonly bic?: string  
  
  /**  
   * Číslo účtu v IBAN. Pokud je mimo IBAN zónu, nebude záznam ...  
   * @type {string}  
   * @memberof ValidateApiBankAccountValidateUsingGET  
   */  
  readonly iban?: string  
}
```



```
amount:  
  oneOf:  
    - type: number  
    - type: string
```

# Přihlášení uživatele



- › Jak to funguje
  - Uživatel zadá jméno a heslo
  - Frontend odešle na server a server vrátí token
  - Následující volání se ověřují tokenem
- › Příklad 1: Jakou HTTP metodu použijeme pro získání tokenu?
  - GET nebo POST?

# Přihlášení uživatele



- › Jak to funguje
  - Uživatel zadá jméno a heslo
  - Frontend odešle na server a server vrátí token
  - Následující volání se ověřují tokenem
- › Příklad 1: Jakou HTTP metodu použijeme pro získání tokenu?
  - POST
- › Příklad 2: Kam v prohlížeči bezpečně uložíme vrácený token?
  - Local storage
  - Sesion storage
  - Cookie
  - Statická proměnná
  - IndexedDB



# HTTP kódy

- › Přijdeme na stránku dostupnou pouze přihlášenému uživateli
  - Dostaneme odpověď HTTP 401 Unauthorized
- › Typy statusů
  - 1xx: Informational
  - 2xx: Success
  - 3xx: Redirection
  - 4xx: Client Error
  - 5xx: Server Error



**Steve Losh** @stevelosh · 28. 8. 2013

HTTP status ranges in a nutshell:

1xx: hold on

2xx: here you go

3xx: go away

4xx: you fucked up

5xx: I fucked up

# Komunikace – zabezpečení

## › HTTPS



Your connection is not private

Attackers might be trying to steal your information from **domain.com** (for example, passwords, messages, or credit cards). [Learn more](#)

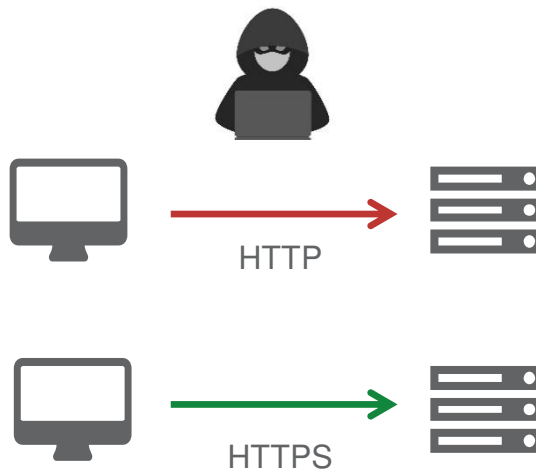
NET::ERR\_CERT\_COMMON\_NAME\_INVALID

Hide advanced

Back to safety

## › Certifikáty

- Už jste si někdy vytvářeli certifikát?



## Certifikáty – typické scénáře

- › Potřebujeme vytvořit certifikát pro testovací server
  - Jak certifikát vygenerovat?
  - Jak ho nastavit pro server?
- › Volám službu třetí strany, která ale nemá důvěryhodný certifikát
  - Co je důvěryhodný certifikát?
  - Co je certifikační autorita?
  - Jak přidám certifikát mezi důvěryhodné?
- › Nejde provolat službu a logy indikují, že problém je v SSL
  - Jak zapnu logování SSL komunikace?
  - SSL handshake, domluva na algoritmu šifrování

# A další

- › Autentizace a autorizace
  - Ověřování a ukládání hesel
  - Session, session cookie
  - Basic Auth, Bearer token, API key
  - OAuth, OpenID, JWT
- › Bezpečnost
  - Šifrování vs. hashování
  - Symetrická vs. asymetrická kryptografie
  - Zranitelnosti
    - XSS, CSRF, injections (SQL a jiné)
    - OWASP Top 10





# Materiály

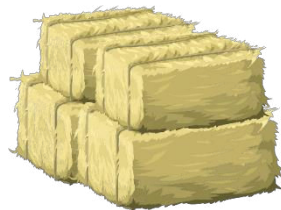
- › NSWI145 Webové služby
- › NSWI163 Úvod do middleware
- › NSWI090 Počítačové sítě
  - Jiří Peterka: Počítačové sítě, verze 4.0 (<http://www.earchiv.cz/l226/>)
- › NSWI089 a NSWI071 Ochrana informací 1 a 2
- › <https://owasp.org/> - Open Web Application Security Project®



Podpora v produkcii

# Ladění výkonu

- › Co když je aplikace pomalá?
- › Hledáme problémové místo
  - Máme **logy**
  - **Monitorujeme** výkon



**TABLESCAN**

- › Možná jen tabulka v databázi potřebuje (lepší) index?
- › Jak zjistit, jak dlouho trvá dotaz?
- › Jak se databáze rozhodne, jaký použije index?
- › Jak zvolit co nejlepší index?



**INDEX SCAN**

# Exekuční plán

- › Teorie: Selekce, projekce, řazení – výpočet ceny dotazu

```
select * from trade t join mismatched_field f
  on t.id= f.trade_id
where f.status = 'RESOLVED'
```

- › Typický příkaz:

- explain plan

- › Zjistí:

- Jak se dotaz vykonává
- Používá indexy?

|    |                                                     |                                 |                  |      |       |                    |  |
|----|-----------------------------------------------------|---------------------------------|------------------|------|-------|--------------------|--|
| 1  | Plan hash value: 608591764                          |                                 |                  |      |       |                    |  |
| 2  |                                                     |                                 |                  |      |       |                    |  |
| 3  | -----                                               |                                 |                  |      |       |                    |  |
| 4  | Id                                                  | Operation                       | Name             | Rows | Bytes | Cost (%CPU)   Time |  |
| 5  | -----                                               |                                 |                  |      |       |                    |  |
| 6  | 0                                                   | SELECT STATEMENT                |                  | 20   | 5660  | 75 (0)   00:00:01  |  |
| 7  | * 1                                                 | HASH JOIN                       |                  | 20   | 5660  | 75 (0)   00:00:01  |  |
| 8  | * 2                                                 | TABLE ACCESS FULL               | MISMATCHED_FIELD | 20   | 1720  | 68 (0)   00:00:01  |  |
| 9  | 3                                                   | TABLE ACCESS FULL               | TRADE            | 618  | 118K  | 7 (0)   00:00:01   |  |
| 10 | -----                                               |                                 |                  |      |       |                    |  |
| 11 |                                                     |                                 |                  |      |       |                    |  |
| 12 | Predicate Information (identified by operation id): |                                 |                  |      |       |                    |  |
| 13 | -----                                               |                                 |                  |      |       |                    |  |
| 14 |                                                     |                                 |                  |      |       |                    |  |
| 15 | 1 -                                                 | access("T"."ID"="F"."TRADE_ID") |                  |      |       |                    |  |
| 16 | 2 -                                                 | filter("F"."STATUS"='RESOLVED') |                  |      |       |                    |  |

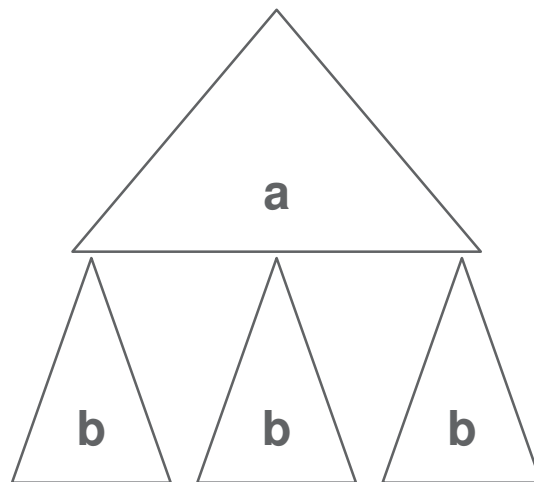
# Indexy

## › Příklad

- create index ... t (a, b)
- Které dotazy mohou použít tento index?
  - A. WHERE a = :a
  - B. WHERE b = :b
  - C. WHERE a = :a AND b = :b

## › Materiály:

- [use-the-index-luke.com](http://use-the-index-luke.com)
- = (kniha SQL Performance Explained)



# Paměť

- › Memory leak
  - Obecně špatné hospodaření s pamětí
    - Neoptimální UI Framework
      - Velké sessions (2 MB)
      - Session i bez přihlášení
  - Problémy (DDoS útoky)
  - Monitoring
  - Memory dump
- › Garbage collection
  - Logování
  - Analýza
  - „Quick and dirty“/dočasné řešení

# Vlákna

## › Thread leak

- Vlákna se nevrací do poolu + spotřebovávají čas procesoru
- Chyba frameworku
  - Server push pomocí Timeru s vláknem
  - Vlákno pro každou komponentu a návštěvníka
- Thread dump
  - Analýza – porovnání několik snímků po sobě

## › Deadlock

- Coffmanovy podmínky
- Aplikace vs. DB
- Řešení: Odstranění cyklického čekání

# Monitoring (příklad) – Java Melody

› Monitorovací stránka webové aplikace

› Snadná konfigurace

› Obsah

– Systémové prostředky

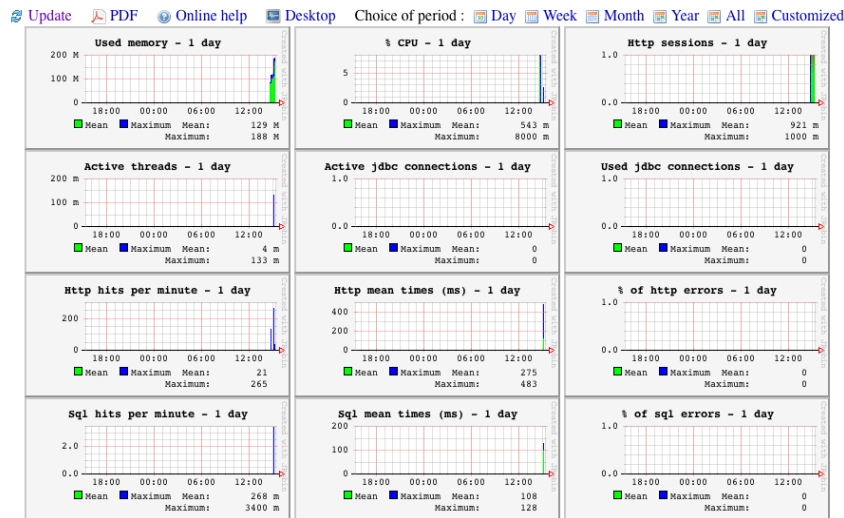
– Statistiky

• HTTP requestů

• SQL dotazů

– Paměť, vlákna

– Cache



| Požadavek                                             | % kumulativního času | Hity | Průměrný čas (ms) | Max. čas (ms) | Standardní odchylka | % využití CPU času | Průměrný čas cpu (ms) | Znamenaná přidělená Kb | % systémové chyby | Průměrná velikost (Kb) | Průměrné hity sql | Průměrný čas sql (ms) |
|-------------------------------------------------------|----------------------|------|-------------------|---------------|---------------------|--------------------|-----------------------|------------------------|-------------------|------------------------|-------------------|-----------------------|
| /appli/JS GET                                         | 78                   | 4    | 164               | 246           | 86                  | 3                  | 1                     | 8                      | 0,00              | 368                    | 0                 | 0                     |
| /appli/AuthenticationService.connexion GWT-RPC        | 10                   | 4    | 21                | 45            | 18                  | 33                 | 9                     | 198                    | 0,00              | 0                      | 3                 | 1                     |
| /appli/PersonneService.findAll GWT-RPC                | 3                    | 5    | 5                 | 7             | 1                   | 21                 | 4                     | 406                    | 0,00              | 0                      | 1                 | 0                     |
| /appli/AuthenticationService.rafraichirDroits GWT-RPC | 3                    | 4    | 6                 | 9             | 1                   | 22                 | 6                     | 251                    | 100,00            | 0                      | 0                 | 0                     |
| /referentiel/GradeService.findAll GWT-RPC             | 1                    | 6    | 1                 | 5             | 1                   | 8                  | 1                     | 68                     | 0,00              | 0                      | 0                 | 0                     |
| /appli/PersonneService.findById GWT-RPC               | 0                    | 1    | 6                 | 6             | 0                   | 5                  | 6                     | 160                    | 0,00              | 0                      | 2                 | 0                     |
| / GET                                                 | 0                    | 5    | 1                 | 1             | 0                   | 2                  | 0                     | 6                      | 0,00              | 1                      | 0                 | 0                     |



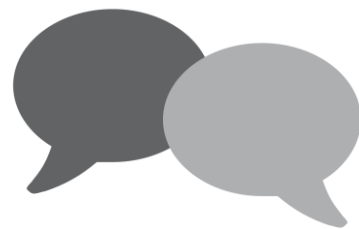
## A další

- › Kvalitní kód – NPRG043 Doporučené postupy v programování
- › Design – NPRG024 Návrhové vzory
- › Nástroje
  - NSWI154 Nástroje pro vývoj software
  - NSWI126 Pokročilé nástroje pro vývoj a monitorování software
- › Operations
  - Správa serverů, databází
  - Cloudy
    - NSWI150 – Virtualizace a cloud computing
    - NSWX152 – Vývoj cloudových aplikací

# Shrnutí

- › Příklady, s čím se může Full Stack vývojář setkat
- › Inspirace k rozšiřování znalostí
- › Širší rozhled se hodí
  - Znalost konceptů usnadňuje učení nových technologií
  - Znalost konceptů usnadňuje řešení problémů





## **Diskuze**

# Děkujeme za pozornost

## PROFINIT

NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.

Tychonova 2, 160 00 Praha 6 | Telefon + 420 224 316 016



Web  
[www.profinit.eu](http://www.profinit.eu)



LinkedIn  
[linkedin.com/company/profinit](https://linkedin.com/company/profinit)



Twitter  
[twitter.com/Profinit\\_EU](https://twitter.com/Profinit_EU)



Facebook  
[facebook.com/Profinit.EU](https://facebook.com/Profinit.EU)



Youtube  
Profinit EU