

# IB053 Webová služba “seznam úkolů”

Zadávací dokumentace projektu

Název projektu:	IB053 Webová služba “seznam úkolů”
Účel projektu:	zápočtový příklad
Verze dokumentu:	1.0
Datum:	23.3.2017
Autor:	Petr Steinmetz
Zadavatel:	Petr Steinmetz
Řešitel:	studenti IB053

# Obsah

[Obsah](#)

[Cíl projektu](#)

[Úvod projektu](#)

[Součásti řešení projektu](#)

[Analytická část \(pro vedoucího týmu\)](#)

[Vývojová část \(programování\)](#)

[Minimální zadání](#)

## Cíl projektu

Cílem projektu je vytvoření aplikace - serveru poskytující webovou službu sloužící k realizaci seznamu úkolů v týmu. Součástí je i jednoduchý klient. Data aplikace budou uložena v databázi SQLite. Webová služba je typu RESTful, data předávaná oběma směry jsou v JSON.

## Úvod projektu

Projekt vytvoří (až) tříčlenný tým, jedna osoba má na starosti datovou vrstvu, druhá komunikační vrstvu a třetí vytvoří jednoduchého klienta (může být i v Javascriptu) pomocí kterého se aplikace otestuje. Jeden ze členů je vedoucím týmu, který upřesní detaily zadání, koordinuje práce a řídí testování. Aplikace nemusí mít administrační rozhraní a rozlišovat oprávnění uživatelů.

Server je možné vytvořit v libovolném programovacím jazyku. Pokud ale půjde o jiný jazyk než Java, C/C++, C# nebo Python, je nutno výběr konzultovat se zadavatelem.

Předání díla spočívá v předvedení funkčnosti, předání zdrojových textů mailem a obhájení zdrojových textů před zadavatelem (bude velmi šťouravý) :-)

## Součásti řešení projektu

Projekt se skládá z části analytické a vývojové.

### Analytická část (pro vedoucího týmu)

- upřesnit datový model
- upřesnit rozhraní webové služby (tj. jaké položky se budou jednotlivými službami předávat)
- rozdělit aplikaci na nezávislé moduly a sepsat zadání podle kterého bude možné jednotlivé moduly naprogramovat
- najít dostupné knihovny pro zvolený programovací jazyk, které usnadní vývoj
- sepsat strategii testování aplikace

Výsledkem této části jsou dokumenty. Dokumentaci každého modulu a plán postupu testování je potřeba v průběhu řešení zaslat zadavateli k odsouhlasení.

### Vývojová část (programování)

- programování modulů dle rozdělení mezi jednotlivé členy týmu
- testování, koordinace

Úkolem vedoucího týmu je

- sledovat a mít přehled o průběhu práce jednotlivých řešitelů

- určit časový harmonogram a zajistit jeho dodržení
- přebírat a kontrolovat dokumenty s návrhem modulů
- přebírat a testovat dodané moduly

Předání hotového řešení je věcí všech řešitelů.

## Minimální zadání

Datový model musí obsahovat minimálně tyto entity:

- uživatel (login, heslo, plné jméno)
- úkol (ID, uživatel, pořadové číslo v rámci uživatele, odhadovaná doba plnění, nepovinná reference na jiný úkol, jehož splnění je nutnou podmínkou splnění tohoto úkolu - ideálně by to měl být seznam úkolů, minimálně však jeden)

Přitom úkol je přiřazen právě jednomu uživateli. ID úkolu je jednoznačný identifikátor úkolu v celém systému (GUID, číslo apod.)

Webová služba nemusí evidovat seznam přihlášených uživatelů a tím ani nemusí držet každému uživateli session - tj. jeho pracovní data uložená na serveru. Přihlášení pouze prověří správnost loginu a hesla. Každá další služba potom dostane kromě dat ke zpracování i znovu login a heslo. Logout tedy není také nutný.

Webové služby poskytují tato volání

- login - IN: login a heslo, OUT: true/false - zda je login funkční
- getTaskList - IN: login a heslo, login uživatele, jehož seznam úkolů se žádá, OUT: seznam úkolů
- addTask: IN: login a heslo, úkol (všechny údaje včetně řešitele a případné reference na jiný úkol) OUT: true/false - povedlo se/nepovedlo se + ID přidělené novému úkolu
- changeTask: IN: login a heslo, položky úkolu, které mají být změněny, OUT: true/false - povedlo se/nepovedlo se
- deleteTask: IN: login a heslo, úkol (stačí ID), OUT: true/false - povedlo se/nepovedlo se
- getTotalTime: IN: login a heslo, login uživatele, jehož se dotaz týká, OUT: doba plnění všech úkolů

Pokud se v addTask nebo v changeTask uvede pořadové číslo úkolu (v rámci uživatele) takové, které již existuje u jiného úkolu, dojde k jeho předřazení a posunutí dalších úkolů dál v pořadí. Pokud se v changeTask uvede jiný uživatel, chová se služba jako deleteTask a addTask (ale se zachováním ID)