HTTP nástěnka

Dokumentace

Filip Jeřábek (xjerab24)

VUT FIT 2019

ISA

# Obsah

[1 Obsah 1](#_Toc24481707)

[2 Obecný popis 2](#_Toc24481708)

[2.1 Úvod 2](#_Toc24481709)

[2.2 Klient 2](#_Toc24481710)

[2.3 Server 2](#_Toc24481711)

[2.4 Zpracování argumentů a volání programu 2](#_Toc24481712)

[2.4.1 Klient 2](#_Toc24481713)

[2.4.2 Server 3](#_Toc24481714)

[2.5 Komunikace serveru a klienta 4](#_Toc24481715)

[2.5.1 Request klienta 4](#_Toc24481716)

[2.5.2 Response serveru 4](#_Toc24481717)

[2.6 Výstup klienta 4](#_Toc24481718)

[3 Užitečné odkazy 5](#_Toc24481719)

[4 Závěr 6](#_Toc24481720)

# Obecný popis

Aplikace implementuje stranu klienta a serveru pro správu nástěnky pomocí HTTP API. API umožňuje klientovi prohlížet, přidávat, upravovat a mazat jak příspěvky na nástěnkách, tak i nástěnky samotné. Nástěnka obsahuje seřazený seznam textových příspěvků.

## Úvod

Pro tvorbu aplikace byl zvolen programovací jazyk C++. Kód je tvořen v IDE CLion od společnosti JetBrains na operačním systému Ubuntu 18.04. Ke správě verzí byl zvolen verzovací systém git.

## Klient

Klient dle zadaných argumentů odesílá požadavky pro správu nástěnky na server. Jednotlivé způsoby volaní jsou uvedeny v kapitole TODO. Klientská aplikace má návratovou hodnotu **0**, pokud se daný požadavek na serveru povedl a hodnotu **-1**, pokud server vrátil chybovou návratovou hodnotu.

## Server

Server čeká na navázání spojení a příjem požadavku od klienta. Tento požadavek si poté zpracuje a odešle na něj příslušnou odpověď. Argumenty akceptující server jsou popsány v kapitole TODO. Implementace

## Zpracování argumentů a volání programu

### Klient

Program nejprve pomocí funkce checkHelp() zkontroluje, zda byl zadán přepínač *–h* a případně vypíše nápovědu a ukončí se.

Pro zpracování argumentů je použita doporučená rodina getopt(), konkrétně getopt\_long\_only(). Funkce pro načtení argumentů je pojmenována loadArgs().

Ta nejprve zavolá funkci checkArgCount(). Tato funkce kontroluje, zda je zadán počet argumentů v rozsahu min max.

Poté načte do struktury myArgs argumenty s přepínači a ve struktuře zanechá informace o tom, které přepínače byly načteny. Dále probíhá validace zadání povinných argumentů.

Dále je zavolána funkce loadCommand() která pomocí regexů v argumentech vyhledává jeden z předem očekávaných příkazů pro správu http nástěnky. Pokud nalezne shodu, pak do struktury myArgs vloží typ a hodnoty příkazu, jako je například „board add jméno“.

#### Přijímané argumenty

Způsob volání klienta:

./isaclient -H <host> -p <port> <command>

kde

* <host> je adresa hosta, na kterou se bude požadavek odesílat
* <port> je port, na kterém bude probíhat komunikace
* <command> je:
* boards - Vráti seznam dostupných nástěnek, jedna na řádek.
* board add <name> - Vytvoří novou prázdnou nástěnku s názvem name.
* board delete <name> - Smaže nástěnku name a veškerý její obsah.
* board list <name> - Zobrazí obsah nástěnky name.
* item add <name> <content> - Vloží nový příspěvek do nástěnky name. Příspěvek je vložený na konec seznamu.
* item delete <name> <id> - Změní obsah příspěvku číslo id v nástěnce name.
* item update <name> <id> <content> - Smaže příspevek číslo id z nástěnky name.

kde:

* <name> je řetězec znaků a-z A-Z a 0-9
* <id> je číselná hodnota v rozsahu 0-65535. Hodnota 65535 byla zvolena jako maximální hodnota typu integer. Předpokládá se, že jedna nástěnka nebude mít více než 65535 příspěvků.

Příklad volání pak bude:

./isaclient –H localhost –p 8080 GET

### Server

Program nejprve pomocí funkce checkHelp() zkontroluje, zda byl zadán přepínač *–h* a případně vypíše nápovědu a ukončí se.

Poté pomocí funkce checkArgCount() ověří počet zadaných argumentů. V tomto případě 2. Přepínač –p a jeho hodnotu.

Jelikož program neočekává žádné další argumenty, kontroluje, zda byl spuštěn s výhradně jen přepínačem –p a jeho hodnotou. Tato hodnota udává číslo portu, na kterém bude aplikace naslouchat. Pokud je zadáno neplatný řetězec (abc), nebo port mimo rozsah (max 65535), pak program nastaví port na hodnotu 65535 a informuje o tom uživatele. Hodnota 65535 byla zvolena jako maximální hodnota typu integer. Předpokládá se, že jedna nástěnka nebude mít více než 65535 příspěvků.

#### Přijímané argumenty

Server přijímá pouze přepínač –h, nebo přepínač –p a k němu jeho hodnotu. Ostatní argumenty povedou k chybě.

Příklad volání pak bude:

./isaserver –p 8080

## Komunikace serveru a klienta

Server a klient spolu komunikují přes API za pomocí jednoduchých http requestů pomocí síťové knihovny BSD sockets. Velikost bufferu byla nastavena na 2048, což pro naši aplikaci představuje optimální velikosti. Pro příspěvky zadávané jako argument je dostatečně velký a naopak ne zbytečně velký.

### Request klienta

Request se skládá ve funkci createRequest(), kde se pomocí příkazu switch vybere příslušná funkce pro naplnění requestu. Funkce nese podobný název, jako požadovaný příkaz (cmdBoards, cmdBoardAdd..).

Request se skládá pouze z nezbytných hlaviček:

* metoda, příkaz api, verze http 1.1
* host
* content-type
* content-length

Takto vytvořený request se poté přidá do hlavní funkce main a odešlě se serveru.

Překlad argumentů programu na API vypadá následovně:

**boards** - GET /boards

**board add <name>** - POST /boards/<name>

**board delete <name>** - DELETE /boards/<name>

**board list <name>** - GET /board/<name>

**item add <name> <content>** - POST /board/<name>

**item delete <name> <id>** - DELETE /board/<name>/<id>

**item update <name> <id>** **<content>** - PUT /board/<name>/<id>

### Response serveru

Server zpracuje a podobně jako klient pomocí funkce getCommand() a createResponse() zpracuje požadavek, provede příslušné operace nad nástěnkami, vytvoří odpověď a odešle ho.

## Výstup klienta

Klient vrací návratovou hodnotu 0 v případě, že operace, která mu byla zadána, proběhla na serveru úspěšně.

Pokud nastala chyba, jak na straně klienta, tak na straně serveru, pak vrací hodnotu -1.

# Užitečné odkazy

http request

[1] <https://www.tutorialspoint.com/http/http_requests.htm>

http response

[2] <https://www.tutorialspoint.com/http/http_responses.htm>

regexy pro zpracovávání požadavku a odpovědi

[3] <http://www.cplusplus.com/reference/regex/smatch/>

# Závěr

Program je plně funkční pro ipv4. Pro Ipv6 není kompletně implementován.