

# ROBOGAMES

## Stopwatch protocol Server

### Abstrakt

Následující text popisuje spuštění a ovládání grafického uživatelského rozhraní aplikace pro podporu testování během vývoje automatických stopek připojitelných k aplikaci pro organizaci zápasů robogames.

### Úvod

Za účelem organizování zápasů v rámci her Robogames je u určitých disciplín zapotřebí odečítat čas jednotlivých úkolů. Pro tento účel je vyvíjeno zařízení pro přesné časové odpočty – automatické stopky. Z důvodu efektivní komunikace mezi aplikací pro organizování her a automatickými stopkami je vyžit protokol WebSocket, který umožňuje obousměrnou komunikaci mezi serverem (aplikací pro organizování her) a klientem (automatickými stopkami). Jelikož jsou tyto projekty vyvíjeny nezávisle na sobě, je definován protokol pro vzájemnou komunikaci.

Pomocí této aplikace lze manuálně simulovat chování serveru a může sloužit jako podpůrný prostředek při vývoji automatických stopek.

### Spuštění programu

Aplikace je napsána v jazyce Java (openjdk verze 8), pro spuštění je zapotřebí mít nainstalované JRE 8 (Java Runtime Environment). Spuštění je možné pomocí skriptu StopwatchServer.bat (v prostředí MS Windows), nebo StopwatchServer (v prostředí GNU/Linux, případně MacOS), které se nacházejí v podadresáři „bin“.

## Popis rozhraní

Aplikace se ovládá pomocí grafického uživatelského rozhraní (viz. obrázek). V horní části aplikace se nachází dvě záložky, které umožňují zobrazení buďto ovládací prvky, nebo globální log.

- |             |  |
|-------------|--|
| hostname    | - doménové jméno, nebo IP adresa pro spuštění serveru  |
| port        | - TCP port na kterém bude server poslouchat  |
| context     | - cesta určující kontext aplikace  |
| Bind        | - tlačítko po stisknutí spustí server  |
| Unbind      | - tlačítko po stisknutí zastaví server   |
| Client URI  | - adresa pro navázání spojení klienta (zapotřebí doplnit vhodné {ID} klienta)  |
| Client list | - seznam připojených klientů, zvolenému klientovi budou odesílány zprávy   |
| Client log  | - log záznam pro zvoleného klienta v listu klientů   |
| start       | - odešle zprávu „start“ s hodnotou zadanou v níže zobrazeném vstupním poli   |
| reset       | - odešle zprávu „reset“ zvolenému klientovi  |
| stop        | - odešle zprávu „stop“ zvolenému klientovi   |
| ping        | - odešle zprávu PING zvolenému klientovi, očekávaná odpověď je: PONG<br>(tyto dvě zprávy jsou definované protokolem WebSocket) |
| close       | - uzavře spojení se zvoleným klientem  |
| send        | - odešle zprávu zadanou vlevo umístěném vstupním poli (label xxx)  |

## Připojení k serveru

Připojení k serveru je možné pomocí klientské WebSocket knihovny. Lze k tomu využít i standardní internetový prohlížeč (např. Mozilla Firefox), ve kterém po spuštění konzole (v menu vybrat položku „Web Developer“ a zvolit možnost „Web Console“, nebo zmáčknout kombinaci kláves Ctrl+Shift+K) pomocí jazyka javascript připojit k serveru pomocí příkazů:

```
var client = new WebSocket("ws://localhost:8025/robogames/stopwatch?id=123")
client.onmessage = function(value) {alert(`Received: ${value.data}`)}
```

Předchozí řádky umožňují zobrazení zprávy přijaté ze serveru pomocí alert dialogu. Pomocí následujícího příkazu můžete odeslat zprávu na server:

```
client.send("started")
```

Pokud klient při připojení k serveru neuvede parametr „id“, nebo hodnota tohoto parametru bude prázdný řetězec, bude spojení ze strany serveru zamítnuto a uzavřeno.

V případě, že se k serveru bude snažit připojit klient se stejným „id“ jako jiný aktuálně připojený klient, bude tento nový klient odmítnut a spojení uzavřeno.