

PROZE 20Z PROJEKT - SOKOBAN

Etap IV

Zespół: Mateusz Brawański, Maciej Kaczkowski

Spis treści

Podstawowe założenia.....	2
Protokół jest niezależny od ścieżki.....	2
Parametry w ścieżce	2
Parametry zapytania	2
Opis XML	2
API HTTP.....	2
GET /configuration.....	2
Przykład.....	2
GET /levels/meta/:pack	3
Przykład.....	3
GET /levels/data/:pack/:level	4
Przykład.....	4
GET /scoreboard/:pack	4
Przykład.....	5
Przykład.....	6
GET /scoreboard/:pack/:level	6
Przykład.....	7
Przykład.....	7
POST /scoreboard/:pack/:level.....	7
Przykład.....	7
Błędy	8
HTTP 500	8
HTTP 404	8
HTTP 400	8

Podstawowe założenia

Protokół został oparty o protokół HTTP, ze względu na dostępne w Javie narzędzia do obsługi tego protokołu, jak również dużą łatwość w uruchomieniu takiego serwera. Połączenia HTTP daje się łatwo przekazywać oraz debugować, co również ułatwia diagnostykę protokołu gry, a w dodatku umożliwia zabezpieczenie połączenia poprzez zastosowanie protokołu TLS. Biorąc to pod uwagę, protokół opisany został w ramach możliwości oraz funkcji protokołu HTTP.

Protokół jest niezależny od ścieżki

Wszystkie opisane poniżej endpointy HTTP mogą mieć dowolny prefix ścieżki, tzn. że na serwerze `http://example.org/` endpoint `GET /test` może być dostępny jako `http://example.org/test`, lub `http://example.org/some/long/path/with/many/levels/test`. Wymagana jest wtedy odpowiednia konfiguracja aplikacji klienckiej.

Parametry w ścieżce

Fragmenty ścieżki rozpoczynające się od dwukropków są parametrami. Oznacza to, że ten fragment jest zastępowany wartością. Przykładowo, `/hello/:name` może zostać wywołane jako `/hello/world`, co oznacza że wywołany został endpoint `/hello/:name`, gdzie `name` ma wartość `world`.

Parametry zapytania

Parametry zapytania (tzw. query string) są opisywane oddzielnie od ścieżki.

Opis XML

Przy opisie struktury XML, prefix `@` oznacza atrybut.

API HTTP

GET /configuration

Zwraca konfigurację aplikacji w formacie XML. Dokument ten ma następującą strukturę:

- `configuration` – definiuje konfigurację
 - `level-pack` – identyfikator paczki poziomów na której gracze będą grali
 - `max-lives` – maksymalna liczba żyć, którą gracz może dysponować
 - `start-lives` – liczba żyć, z którą gracz rozpoczyna grę
 - `life-recovery-threshold` – ilość poziomów, które gracz musi ukończyć bez resetowania pod rząd, żeby odzyskać część utraconych żyć
 - `life-recovery-count` – ilość żyć które gracz odzyskuje po przejściu określonej liczby poziomów pod rząd bez resetowania
 - `timers-active` – czy limity czasowe są aktywne
 - `active-powerup` – włączone power-upy, ten element występuje wielokrotnie, raz dla każdego power-upa.

Przykład

`GET /configuration`

Odpowiedź (HTTP 200)

```
<configuration>
  <level-pack>pack0</level-pack>
  <max-lives>5</max-lives>
  <start-lives>5</start-lives>
  <life-recovery-threshold>3</life-recovery-threshold>
  <life-recovery-count>1</life-recovery-count>
  <timers-active>true</timers-active>
```

```
<active-powerup>STRENGTH</active-powerup>
<active-powerup>GHOST</active-powerup>
<active-powerup>PULL</active-powerup>
</configuration>
```

GET /levels/meta/:pack

Parametr	Typ	Opis
pack	tekst	Identyfikator paczki poziomów, którą chce się uzyskać.

Zwraca metadane oraz definicje poziomów zawartych w paczce poziomów o ID odpowiadającemu parametrowi pack, w formacie XML. Dokument ma następującą strukturę:

- level-pack – definiuje paczkę poziomów
 - @id – identyfikator paczki
 - @name – nazwa paczki
 - level – definiuje poziom, występuje wielokrotnie, raz dla każdego zawartego poziomu
 - ordinal – numer poziomu, począwszy od 0
 - name – nazwa poziomu
 - bonus-time – liczba sekund, poniżej której ukończenie poziomu skutkuje bonusem
 - penalty-time – liczba sekund, przekroczenie której przy rozwiązywaniu poziomu skutkuje karą
 - fail-time – liczba sekund, po której poziom jest resetowany, a gracz traci życie
 - definition – plik, zawierający definicję poziomu

Przykład

GET /levels/meta/pack0

Odpowiedź (HTTP 200)

```
<level-pack id="pack0" name="Poziomy podstawowe">
  <level>
    <ordinal>0</ordinal>
    <name>Poziom 1</name>
    <bonus-time>30</bonus-time>
    <penalty-time>90</penalty-time>
    <fail-time>150</fail-time>
    <definition>LEVEL_0.txt</definition>
  </level>
  <level>
    <ordinal>1</ordinal>
    <name>Poziom 2</name>
    <bonus-time>30</bonus-time>
    <penalty-time>90</penalty-time>
    <fail-time>150</fail-time>
    <definition>LEVEL_1.txt</definition>
  </level>
  <level>
    <ordinal>2</ordinal>
    <name>Poziom 3</name>
    <bonus-time>30</bonus-time>
    <penalty-time>90</penalty-time>
    <fail-time>150</fail-time>
```

```

        <definition>LEVEL_2.txt</definition>
</level>
<level>
    <ordinal>3</ordinal>
    <name>Poziom 4</name>
    <bonus-time>30</bonus-time>
    <penalty-time>90</penalty-time>
    <fail-time>150</fail-time>
    <definition>LEVEL_3.txt</definition>
</level>
</level-pack>

```

GET /levels/data/:pack/:level

Parametr	Typ	Opis
pack	tekst	Identyfikator paczki poziomów, którą chce się uzyskać.
Level	liczba	Numer poziomu, którego dane chce się uzyskać, zaczynając od 0.

Zwraca definicję poziomu, czyli tekstowy opis rozłożenia elementów poziomu. W definicji tej, poszczególne znaki opisują różne elementy:

- # – ściana, nie do przejścia
- _ – podłoga, można po niej chodzić
- X – miejsce, w którym należy ustawić skrzynię, można po nim chodzić
- S – pozycja startowa gracza
- P – skrzynia
- G – pole aktywujące power-up GHOST, można po nim chodzić
- U – pole aktywujące power-up PULL, można po nim chodzić
- T – pole aktywujące power-up STRENGTH, można po nim chodzić

Przykład

GET /levels/data/pack0/0

Odpowiedź (HTTP 200)

```

#####
###_##
#XSP_##
###_PX##
#X##P_##
#_#_X_##
#PPXPPX#
#___X___#
#####

```

GET /scoreboard/:pack

Parametr	Typ	Opis
pack	tekst	Identyfikator paczki poziomów, której tabelę chce się uzyskać.

Parametry zapytania:

Parametr	Typ	Opis
player	tekst	Nazwa gracza, którego wyniki chce się uzyskać. Wielkość liter nie ma znaczenia.

Zwraca tabelę wszystkich najlepszych wyników dla danej paczki poziomów, w formacie XML. Można opcjonalnie podać nazwę gracza, wtedy zwrócone zostaną tylko wyniki tego gracza. Struktura tego dokumentu przedstawia się następująco:

- **scoreboard** – definiuje tabelę wyników
 - **level-pack** – identyfikator paczki poziomów, odpowiadającej tej tabeli
 - **entry** – wpis w tabeli, element występuje wielokrotnie, po raz na każdy wpis
 - **player** – nazwa gracza, który dokonał tego wpisu
 - **level** – numer poziomu, któremu odpowiada wynik
 - **score** – uzyskany na danym poziomie przez gracza wynik

Przykład

GET /scoreboard/pack0

Odpowiedź (HTTP 200)

```
<scoreboard>
  <level-pack>pack0</level-pack>
  <entry>
    <player>EMZI</player>
    <level>0</level>
    <score>50</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>1</level>
    <score>150</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>2</level>
    <score>60</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>3</level>
    <score>120</score>
  </entry>
  <entry>
    <player>JOHN</player>
    <level>0</level>
    <score>60</score>
  </entry>
  <entry>
    <player>JOHN</player>
    <level>0</level>
```

```
        <score>200</score>
    </entry>
    <entry>
        <player>JOHN</player>
        <level>0</level>
        <score>90</score>
    </entry>
</scoreboard>
```

Przykład

GET /scoreboard/pack0?player=EMZI

Odpowiedź (HTTP 200)

```
<scoreboard>
  <level-pack>pack0</level-pack>
  <entry>
    <player>EMZI</player>
    <level>0</level>
    <score>50</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>1</level>
    <score>150</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>2</level>
    <score>60</score>
  </entry>
  <entry>
    <player>EMZI</player>
    <level>3</level>
    <score>120</score>
  </entry>
</scoreboard>
```

GET /scoreboard/:pack/:level

Parametr	Typ	Opis
pack	tekst	Identyfikator paczki poziomów, której tabelę chce się uzyskać.
level	liczba	Numer poziomu, którego tabelę chce się uzyskać, zaczynając od 0.

Parametry zapytania:

Parametr	Typ	Opis
player	tekst	Nazwa gracza, którego wyniki chce się uzyskać. Wielkość liter nie ma znaczenia.

Zwraca tabelę wszystkich najlepszych wyników dla konkretnego poziomu, w formacie XML. Można opcjonalnie podać nazwę gracza, wtedy zwrócone zostaną tylko wyniki tego gracza. Struktura dokumentu jest identyczna jak w poprzednim przypadku.

Przykład

GET /scoreboard/pack0/0

Odpowiedź (HTTP 200)

```
<scoreboard>
  <level-pack>pack0</level-pack>
  <entry>
    <player>EMZI</player>
    <level>0</level>
    <score>50</score>
  </entry>
  <entry>
    <player>JOHN</player>
    <level>0</level>
    <score>60</score>
  </entry>
</scoreboard>
```

Przykład

GET /scoreboard/pack0/0?player=EMZI

Odpowiedź (HTTP 200)

```
<scoreboard>
  <level-pack>pack0</level-pack>
  <entry>
    <player>EMZI</player>
    <level>0</level>
    <score>50</score>
  </entry>
</scoreboard>
```

POST /scoreboard/:pack/:level

Parametr	Typ	Opis
pack	tekst	Identyfikator paczki poziomów, do tabeli której wynik będzie zapisany.
level	liczba	Numer poziomu, do tabeli którego wynik będzie zapisany, zaczynając od 0.

Parametry zapytania:

Parametr	Typ	Opis
Player	tekst	Nazwa gracza, którego wynik chce się zapisać. Wielkość liter nie ma znaczenia. Parametr jest wymagany.

Zapisuje nowy wynik do tabeli. Wynik jest zapisywany tylko jeżeli jest niższy. W przypadku pomyślnego zapisania, nie jest zwracana żadna zawartość.

Przykład

POST /scoreboard/pack0/0?player=EMZI

48

Odpowiedź (HTTP 204)

Błędy

HTTP 500

Wystąpił błąd w aplikacji serwera. Generalnie powtórzenie zapytania nie zmieni rezultatu.

HTTP 404

Podany endpoint jest nieprawidłowy, albo podana paczka poziomów lub poziom nie istnieje.

HTTP 400

Wysłane zostały niepoprawne dane. Należy skorygować błąd i ponowić żądanie.