

Semestrální projekt z Multimediálních systémů (ver. 2020/B)

Vytvoření projektu

Připravte si pomocí inicializátoru projektu (<https://start.spring.io/>) projekt s následujícími parametry:

NASTAVENÍ	HODNOTA
PROJECT	Gradle Project
LANGUAGE	Java
SPRING BOOT	2.3.5. (nebo hodnotu, která bude automaticky vybrána)
GROUP	cz.vutbr.feec.utko.bpcmds
ARTIFACT	projekt
NAME	projekt
DESCRIPTION	Semestrální projekt
PACKAGE NAME	cz.vutbr.feec.utko.bpcmds.projekt
PACKAGING	Jar
JAVA	11

V rámci výběru závislostí (Dependencies) vyberte:

- **Spring Web**
- **Thymeleaf**

Vygenerovaný projekt stáhněte a pokračujte v IDE IntelliJ IDEA.

Zadání vypracování projektu

Všechnu logiku webového serveru umísťujte do nové třídy **WebController.java**. Případné metody **handlerů** nebo **servletů** do tříd jim určených.

Implementujte kód v jazyce Java, který rozšíří funkcionalitu serveru o podporu poskytování adaptivního streamu. Implementaci můžete omezit pouze pro adaptivní stream typu MPEG-DASH.

Vytvořte třídu komponenty s názvem **ProjectResourceComponent**, která bude rozšiřovat třídu **ResourceHttpRequestHandler**. V této třídě vytvořte metodu **getResource()**, která bude rozšiřovat pomocí anotace **@Override** stejnojmennou metodu z třídy **ResourceHttpRequestHandler**. **(1b)**

V třídě **WebController.java** vytvořte metodu s názvem **streaming()**. Tato metoda bude naslouchat na adresách „/dash/{file}“, kde {file} značí proměnnou složku adresy, kde bude umístěn dotazovaný soubor. Celá metoda bude poskytovat data adaptivního streamu typu MPEG-DASH. Celá metoda bude využívat globální proměnnou **DASH_DIRECTORY**, ve které bude definována cesta ke kořenové složce adaptivního streamu typu MPEG-DASH. Metoda bude také reagovat pouze na požadavek typu **GET**. **(1b)**

Dále vytvořte metodu **index()**, naslouchající na adrese „/index“ (<http://localhost:8080/index>), která bude poskytovat stejnojmennou šablonu (**index.html**). V šabloně bude umístěn element s přehrávačem, který bude přehrávat lokální stream uložený v počítači.

V šabloně bude „napevno“ definován zdroj adaptivního streamu:

<http://localhost:8080/dash/manifest.mpd>

V šabloně využijte JS knihovnu Dash.JS, která je pro přehrávání adaptivního streamu MPEG-DASH nezbytná. Pro knihovnu vytvořte samostatnou složku s názvem „js“. **(1b)**

Přehrávač bude obsahovat volbu přehrávaného profilu. (viz domácí úkol č. 3) Soubory potřebné pro plnou implementaci přehrávače jsou umístěny v sekci zadání projektu. **(1b)**

Po splnění všech předchozích kroků bude stránka vypadat následovně:



U přehrávače je důležité mít zobrazené profily videa. Nezobrazení profilů audia **není** bráno jako chyba. Přehrávač **nebude** po načtení stránky automaticky spouštět přehrávání. **Pro testování využijte pouze soubory z elearningu!**

V projektu dále implementujte funkcionalitu, která bude vytvářet seznam dostupných videí.

Začněte vytvořením třídy **Video**, která bude obsahovat informace o zdroji adaptivního streamu. Třída bude obsahovat objekt typu **String** s názvem **URL**, ve kterém bude uložena URL adresa adaptivního streamu. Druhý objekt v třídě **Video** bude také typu **String** a bude nazván **name**. Objekt **name** bude obsahovat název konkrétního streamu. V třídě **budou přítomny tzv. gettery a settery** objektů. Oba přítomné objekty budou **privátní**. K hodnotám přistupujte skrze „**getter**“. Objekt nastavujte pomocí konstruktoru. **(0,5b)**

Dále vytvořte metodu s názvem **addVideo()**, která bude naslouchat na adrese „**/addvideo**“. Metoda bude poskytovat stejnojmennou šablonu **addvideo.html**. V šabloně se bude nacházet formulář. Formulář bude obsahovat dva prvky typu **input type=“text”** a prvek typu **button** s textem „**Odeslat**“. Nad každým prvkem typu **input** bude umístěn prvek typu **label** pro popis konkrétního inputu. **(2b)**

Formulář bude vypadat následovně:

Adresa:

Název:

Odeslat

Přičemž do pole „Adresa“ uživatel vloží URL adresu videa a do pole „Název“ uživatel vloží název videa. Při kliknutí na tlačítko „Odeslat“ budou hodnoty předány stránce na adrese „/videolibrary“ pomocí metody **POST**.

Pro stránku se seznamem dostupných videí (/videolibrary) vytvořte metodu **videoLibrary()**. Metoda bude obsahovat argumenty **String url** a **String name**, pro předání hodnot z formuláře. Metoda bude volat stejnojmennou šablonu **videolibrary.html**.

Stránka /videolibrary bude naslouchat jak metodě **POST**, tak metoda **GET**. Bude tedy možné stránku otevřít i v prohlížeči bez předání jakýchkoliv dat (v tomto případě je myšleno předání dat argumentům **url** a **name**).

Při zavolání stránky (ať už z formuláře, nebo i přímo v prohlížeči) bude stránka vracet seznam **ul** s prvky **li**. Seznam bude obsahovat aktuální (včetně nově přidané položky při volání z formuláře) položky přítomné v objektu typu **ArrayList**, který bude obsahovat objekty **Video**. Při prvním spuštění nebude seznam obsahovat žádný objekt a stránka /videolibrary bude prázdná. V případě postupného přidávání skrze stránku „/addvideo“ bude seznam zobrazovat položky z objektu typu **ArrayList**. Položky v seznamu na stránce budou odkazy, kdy text odkazu bude název videa. Při kliknutí na odkaz bude uživatel přesměrován na další stránku na adrese „/player“. (2,5b)

Stránka /videolibrary obsahující dvě položky v seznamu bude vypadat následovně:

- [BBC Test](#)
- [Elephant](#)

Při vyplňování může nastat, že jedna z položek nebude vyplněna. V případě nevyplnění položky „Název“ se na stránce „/videolibrary“ zobrazí hláška:

Nebylo vyplněno žádné jméno!

- [BBC Test](#)
- [Elephant](#)

S chybovou hláškou:

Nebylo vyplněno žádné jméno!

Při nevyplnění adresy se zobrazí stránka:

Nebyla vyplněna žádná adresa!

- [BBC Test](#)
- [Elephant](#)

S chybovou hláškou:

Nebyla vyplněna žádná adresa!

Jako poslední vytvořte metodu **player()**, která bude naslouchat na adrese „/player“. Metoda bude předávat obsah šablony s názvem **player.html**.

Na vygenerované stránce bude umístěn přehrávač schopný přehrát adaptivní stream MPEG-DASH včetně možnosti přepnout profil streamu.

Celá stránka bude volána pouze pomocí odkazu ze seznamu na stránce **/videolibrary**. **(1b)**

Konec zadání

Možné problémy a jejich teoretické řešení

- 1) Pokud vytváříte projekt na systému Windows užívejte v cestách zpětná lomítka (\), u systému na bázi Unixu (Linux nebo i macOS) používejte klasická lomítka (/). **Před odevzdáním však cesty upravte pro použití na Windows.** Děkuji.
- 2) Vkládání dat z modelu MVC do proměnné v JavaScriptu:

```
url = '[$${video.getName()}]';
```

Hodnocení projektu

Projekt bude hodnocen jak po celkové funkční stránce, tak i po dílčích úkolech. Celkově je možné z projektu získat 10b. Body jsou v projektu rozloženy následovně:

1. část

- 1b – vytvoření komponenty **ProjectResourceComponent** a její funkčnost
- 1b – metoda **streaming()**, její základní funkčnost definovaná na první straně
- 1b – metoda **index()** a její funkčnost ověřená výslednou stránkou v prohlížeči
- 1b – možnost výběru profilu videa streamu

2. část

- 0,5b – vytvoření třídy **Video** pro ukládání informací o streamu
- 2b – vytvoření metody **addVideo** formulářem pro přidání informací o videu
- 2,5b – vytvoření metody **videoLibrary** generující webovou stránku z šablony se seznamem přidáných streamů
- 1b – vytvoření metody **player** generující webovou stránku z šablony s přehrávačem streamu získaného ze stránky **videolibrary** (přehrávač bude obsahovat možnost přepínání profilu videa)

Celkově 10b

Odevzdání projektu

Projekt odevzdejte v souboru **.zip**. Jiné formáty na elearning nelze nahrát. Soubor zip bude pojmenován podle **VAŠEHO ID!**

Bude-li odevzdaný soubor pojmenován jinak, je automaticky hodnocen 0b!

Obsah souboru zip bude složka **main**, která se v projektu nachází ve složce **src** (stejně jako domácí úkoly).

Struktura bude následující:

- <ID>.zip – main – java – balíček se soubory tříd
 - templates – složky se soubory šablon a dalšími potřebnými soubory pro běh šablon