

Semestrální projekt z Multimediálních systémů (ver. 2022/Po 17:00)

Vytvoření projektu [0,5 b]

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

NASTAVENÍ	HODNOTA
PROJECT	Maven Project
LANGUAGE	Java
SPRING BOOT	2.7.5. (nebo hodnotu, která bude automaticky vybrána)
GROUP	mds
ARTIFACT NAME	project
DESCRIPTION	Semestralni projekt
PACKAGE NAME	mds.project
PACKAGING	Jar
JAVA	11

V rámci výběru závislostí (Dependencies) vyberte **Spring Web** a **Thymeleaf**.

Vygenerovaný projekt stáhněte a pokračujte v IDE IntelliJ IDEA. Správně vytvořený projekt [0,5 b], pokud však bude špatně, tzn. že se balíček nejmenuje „mds.project“ uvnitř kterého je main třída s názvem ProjectApplication.java, budou body strženy... [-1 b]

V elearningu u zadání se dále nachází složka se statickými soubory, které můžete potřebovat pro vypracování projektu. Do projektu si je tedy správně nainportujte (soubory by se měli nacházet ve správných složkách).

Všechnu logiku webového serveru umísťujte do nové třídy **WebController.java** (nezapomeňte na anotaci @Controller). Případné metody **handlerů** nebo **servletů** do tříd jim určeným.

V celém projektu dodržujte adresy, názvy metod a šablon podle zadání. Chybné názvy budou penalizovány.

Cesty k souborům a adresám [0,5 b]

Vytvořte si třídu **FilePaths**, která bude obsahovat veškeré potřebné cesty k souborům a URL adresám. Definované statické proměnné pak budete využívat ve zbytku projektu. Žádná kompletní cesta k souborům nebude definovaná na jiném místě v projektu. Bude obsahovat následující statické cesty: [0,5 b]

- VIDEO_FROM_URL = "http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/BigBuckBunny.mp4"
- DASH_DIRECTORY = "D:\\MDS\\files\\project\\streams\\"

Složka streams obsahuje složky s MPEG-DASH streamy (uvnitř složek je vždy manifest.mpd).

V učebně jsou soubory předpřipravené. Na vlastním PC si musíte stáhnout a vytvořit. Viz pokyny v elearningu.

K cestám můžete přistupovat přes import, nebo následovně: FilePaths.DASH_DIRECTORY;

Úvodní stránka [2 b]

Ve třídě **WebController** vytvořte metodu **index**, která naslouchá na adrese „/index“ a bude využívat šablonu **index.html**. V šabloně nastavte **title** na „Project – VAŠE_ID“. Na stránce bude jako první element **h1**, uvnitř kterého napíšete své jméno a příjmení. Dále změňte barvu textu tohoto elementu na **oranžovou** prostřednictvím souboru **myStyle.css** (který si správně nalinkujete ze složky css). [1 b]

Dále bude na stránce formulář, který bude volat metodou **POST** adresu „/video“ a bude předávat následující parametry: [0,5 b]

- URL adresa videa (text input, jméno parametru **url**)
- Výška videa (text input, jméno parametru **height**)

Dále na stránku přidejte tlačítko nebo jiný odkaz, který se bude jmenovat **Videa** a bude směřovat na adresu „/videocollection“. (Adresy, na které tlačítka směřují vytvoříte dále v zadání) [0,5 b]

Zobrazení videa z URL [1 b]

Ve třídě **WebController** vytvořte metodu **video**, která naslouchá na adrese „/video“ a přijímá parametry z formuláře na stránce index (String url, String height) a přijímá pouze požadavky typu **POST**. V metodě vyřešte výchozí hodnoty zadaných parametrů. Pokud uživatel nezadá adresu videa, tak použijte statickou proměnnou **VIDEO_FROM_URL** (z třídy FilePaths). Nezadaná výška videa bude ve výchozím stavu **400px**.

Url i výška se přes Model předají do šablony **video.html**, kde se aplikují na element video. [1 b]

Další URL, kterou můžete otestovat funkčnost: [Online video](#).

Resource component [0,5 b]

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**. [0,5 b]

Metoda odpovídá třídě MyResourceHttpRequestHandler ze cvičení. Nezapomeňte nalinkovat ve třídě WebController, stačí využít @Autowired přímo u definice proměnné.

Získávání souborů z cesty (byterange) [1 b]

Ve třídě **WebController** vytvořte metodu s názvem **streaming()**. Tato metoda bude naslouchat na adrese „/dash/{stream}/{file}“, kde **{stream}** značí proměnnou složku adresy (@PathVariable), kde bude umístěn dotazovaný soubor **{file}**. Metoda bude poskytovat data adaptivního streamu typu MPEG-DASH. Celá metoda bude využívat globální proměnnou **DASH_DIRECTORY**, která definuje kořenovou složku, ve které se nachází složky ({stream}) s jednotlivými adaptivními streamy typu MPEG-DASH. Metoda bude také reagovat pouze na požadavek typu **GET**. [1 b]

Metoda nepotřebuje žádný switch oproti metodě ze cvičení. Pouze nastavit správně soubor z předaných parametrů a statické cesty DASH_DIRECTORY.

Přehrání adaptivního streamu [1 b]

Ve třídě **WebController** vytvořte metodu **player()**, která naslouchá na adrese „/player/{stream}“. Metoda bude využívat šablonu **player.html**, kde implementujete přehrávač schopný přehrát adaptivní stream typu MPEG-DASH z adresy **/dash/{stream}/manifest.mpd**. [1 b]

Pro přehrání adaptivního streamu využijte knihovnu Dash.js. Správně ji tedy nalinkujete v šabloně.

Pro přehrání v elementu video lze použít th:src="@{'/dash/'+'\${stream}'+'/manifest.mpd'}"

Vyhledání souborů [1,5 b]

Vytvořte novou třídu **VideoLibrary**, která bude rozšiřovat třídu **ArrayList<String>**. Uvnitř této třídy naplňte tento list názvy složek, ve kterých se nachází soubory s MPEG-DASH streamy. K nalezení souborů využijte statickou cestu **DASH_DIRECTORY**, kde tuto cestu využijte v metodě **discoverFiles**. Tuto metodu využijte přímo v konstruktoru třídy **VideoLibrary** a za pomoci cyklu naplňte list (this.add) názvy složek. [1,5 b]

Metoda discoverFiles je v podstatě shodná s metodou ze cvičení s galerií. Pouhý rozdíl je v hledané koncovce.

*Zaměřte se pouze na vyhledání souborů, které končí na **mpd**. Pokud bude problém s třídou SimplePathVisitor, můžete využít SimpleFileVisitor, nebo musíte doplnit knihovnu (Apache Commons IO).*

*Do listu vkládejte složku, ve které se nachází soubor.mpd. Zjištění názvu složky z nalezeného souboru provedete pomocí: **file.getParentFile().getName()**. (Platí pouze, pokud je file datového typu File)*

Zobrazení kolekce videí [1 b]

Ve třídě **WebController** vytvořte metodu **videoCollection()**, která naslouchá na adrese „/videocollection“ a využívá šablonu **videocollection.html**. V této metodě využijte/vytvořte objekt typu VideoLibrary, se kterým budete dále pracovat v šabloně.

V šabloně zobrazíte jednotlivé položky z předaného seznamu. Každá tato položka bude směřovat odkazem na adresu „/player/{stream}“, kde stream značí název složky, ve které se nachází soubor manifest.mpd (zároveň se jedná o každou jednu položku z vytvořeného listu). [1 b]

Po splnění všech předchozích kroků bude stránka /videocollection vypadat například takto:

Videos
[Stream: Big Buck Bunny](#)
[Stream: Sintel](#)
[Stream: Tears_of_Steel](#)

Změna kvality streamu [1 b]

Upravte šablonu player.html (zkopírujte jako novou šablonu s názvem **playerControls.html** a upravte volání šablony v metodě player), tak aby dokázala přehrávat MPEG-DASH streamy a přehrávač umožňoval změnit kvalitu daného streamu. Viz obrázek a knihovna Dash.js (a přiložené statické soubory). [1 b]

Funkčnost můžete otestovat pomocí tohoto streamu: [Stream](#)

Soubory potřebné pro plnou implementaci přehrávače jsou umístěny v sekci zadání projektu.

Doporučení: Pokud se vám nepodaří tento úkol dokončit, vraťte volání šablony v metodě **player** do původního stavu tak, aby docházelo k zobrazení šablony player.html.

Pokud bude změna kvality funkční či nikoliv, nechte v projektu obě šablony. (z částečného řešení se třeba naleznou nějaké desetinové body.)

Předání proměnné do JavaScriptu: `url = 'http://localhost:8080/dash/[[${name}]]/manifest.mpd';`

Po splnění všech předchozích kroků bude stránka vypadat následovně:



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

Konec zadání

Možné problémy

Proměnné v šabloně jsou bez hodnot.

Zkontrolujte si název proměnné v šabloně s názvem, který do šablony předáváte. Název proměnné v kódu je první String parametr volání metody `model.addAttribute("nazevProměnné", hodnota)`.

Mám nedefinovanou cestu, ale v prohlížeči mám chybu 404.

Zkontrolujte si zadanou adresu v definici metody, šabloně a prohlížeči. Cesty jsou case sensitive.

Stránka nenačítá konkrétní část šablony.

Zkontrolujte si DevTools (F12), jaký HTML kód stránka dostala. Odtud můžete zkoušet otevírat i použité odkazy v novém panelu (např. zdroj videa, odkaz na stránku atd.)

PS: Kontrolujte si konzoli v IntelliJ, zda se zde nenachází nějaký error. Stačí vyscrollovat nahoru do první mezery a zde je nejčastěji přímo popsán daný problém. (často se jedná o chybějící znak v šabloně).

Před odevzdáním projektu

Změňte cesty k souborům, tak aby seděly se zadáním.

Souhrn bodování

Body	Úloha
0,5 b	Vytvoření projektu
0,5 b	Třída obsahující cesty k souborům
1 b	Úvodní stránka, včetně jména a nalinkování css souborů
0,5 b	Formulář pro předání URL do šablony video.html
0,5 b	Tlačítko pro přesměrování na stránku /videocollection
1 b	Šablona video.html, která dokáže přehrát video ze zadané adresy, případně výchozí URL
0,5 b	Třída ProjectResourceComponent
1 b	Metoda streaming pro předávání videa skrze HttpResourceComponent
1 b	Přehrání streamu pomocí Dash.js, šablona player.html
1,5 b	Třída pro vyhledání souborů/složek, uložení do Listu stringů
1 b	Stránka /videoCollection
1 b	Změna kvality v šabloně player.Controls.html
10 b	Celkem

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 0 b!

Obsah souboru zip bude **složka main**, která se v projektu nachází ve složce **src**.

Struktura bude následující:

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