



Information Retrieval

Assignment 1:


IMDB Spider

Due Date: Thursday, 17.11., 23:59


Patrick Schäfer (patrick.schaefer@hu-berlin.de)


Marc Bux (buxmarcn@informatik.hu-berlin.de)




IMDb: Internet Movie Database

All



[Unbegrenzter Film- und Seriengenuss mit Prime Instant Video](#)
Jetzt 30 Tage testen




[FULL CAST AND CREW](#) | [TRIVIA](#) | [USER REVIEWS](#) | [IMDbPro](#) | [MORE](#)  [SHARE](#)


 **Avatar - Aufbruch nach Pandora (2009)**  **7.9** 10
898.519  [Rate This](#)

Avatar (original title)
12 | 2h 42min | Action, Adventure, Fantasy | 17 December 2009 (Germany)




3:36 | Trailer

 [Watch Now](#)
From EUR2.99 (SD) on Amazon Video




A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.

Director: [James Cameron](#)
Writer: [James Cameron](#)
Stars: [Sam Worthington](#), [Zoe Saldana](#), [Sigourney Weaver](#) | [See full cast & crew »](#)

 **Metascore**
From [metacritic.com](#)

Reviews
3.063 user | [720 critic](#)

 **Popularity**
332 (★ 25)

Cast

[Edit](#)

Cast overview, first billed only:

	Sam Worthington	...	Jake Sully
	Zoe Saldana	...	Neytiri (as Zoë Saldana)
	Sigourney Weaver	...	Dr. Grace Augustine
	Stephen Lang	...	Colonel Miles Quaritch
	Michelle Rodriguez	...	Trudy Chacón
	Giovanni Ribisi	...	Parker Selfridge
	Joel David Moore	...	Norm Spellman
	CCH Pounder	...	Moat
	Wes Studi	...	Eytukan
	Laz Alonso	...	Tsu'tey
	Dileep Rao	...	Dr. Max Patel
	Matt Gerald	...	Corporal Lyle Wainfleet
	Sean Anthony Moran	...	Private Fike
	Jason Whyte	...	Cryo Vault Med Tech
	Scott Lawrence	...	Venture Star Crew Chief

[See full cast »](#)

Storyline

[Edit](#)

When his brother is killed in a robbery, paraplegic Marine Jake Sully decides to take his place in a mission on the distant world of Pandora. There he learns of greedy corporate figurehead Parker Selfridge's intentions of driving off the native humanoid "Na'vi" in order to mine for the

Assignment

- Task:
Given a list of 500 movies, answer queries on movies.
- Problem:
IMDB data is human-readable, but semi-structured.
- Idea:
 - We “scrap”* data from each movie on IMDB.
 - Then, we perform queries on the scrapped data.

* Data scrapping is a technique in which a computer program extracts data from human-readable output coming from another program.

Concrete tasks

1. Implement a JAVA program that reads a list of 500 movie titles from a JSON file.
2. For each movie title, perform a web search on IMDB and retrieve movie's URL.
3. For each movie, extract metadata (actors, budget, description) from movie's URL and store to a JSON file.
4. Implement queries on movies' metadata.

1. Read Movie Titles from JSON File

- Read movie titles from a JSON* file "movies.json":

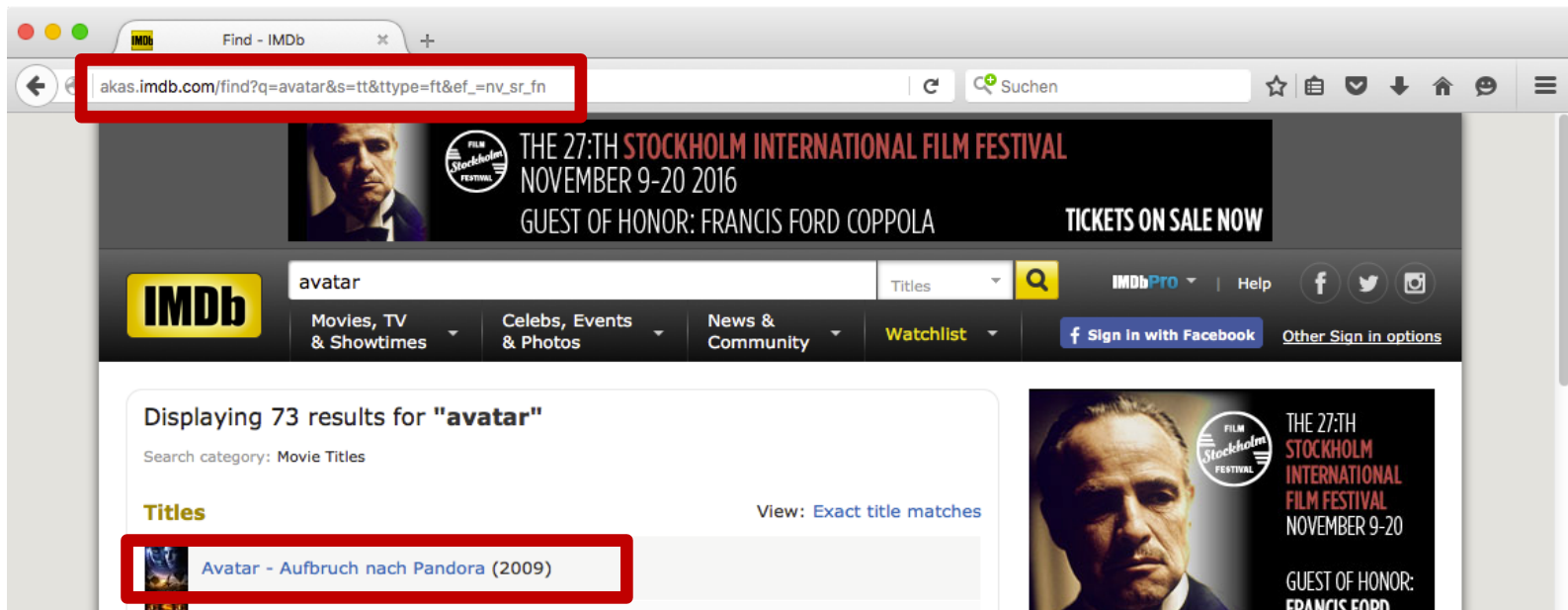
```
[  
    {"movie_name": "Avatar"},  
    {"movie_name": "Star Wars VII: The Force Awakens"},  
    ...  
]
```

- You can use any JAVA library for parsing JSON files.
 - Reference implementation: Oracle's JSONP (<https://jsonp.java.net/>).
 - JSON.simple (<https://github.com/fangyidong/json-simple>).
 - GSON (<https://github.com/google/gson>).
 - Jackson Project (<https://github.com/FasterXML/jackson>).

* JSON is a common syntax for storing and exchanging data. JSON is an alternative to XML:
http://www.w3schools.com/js/js_json_intro.asp

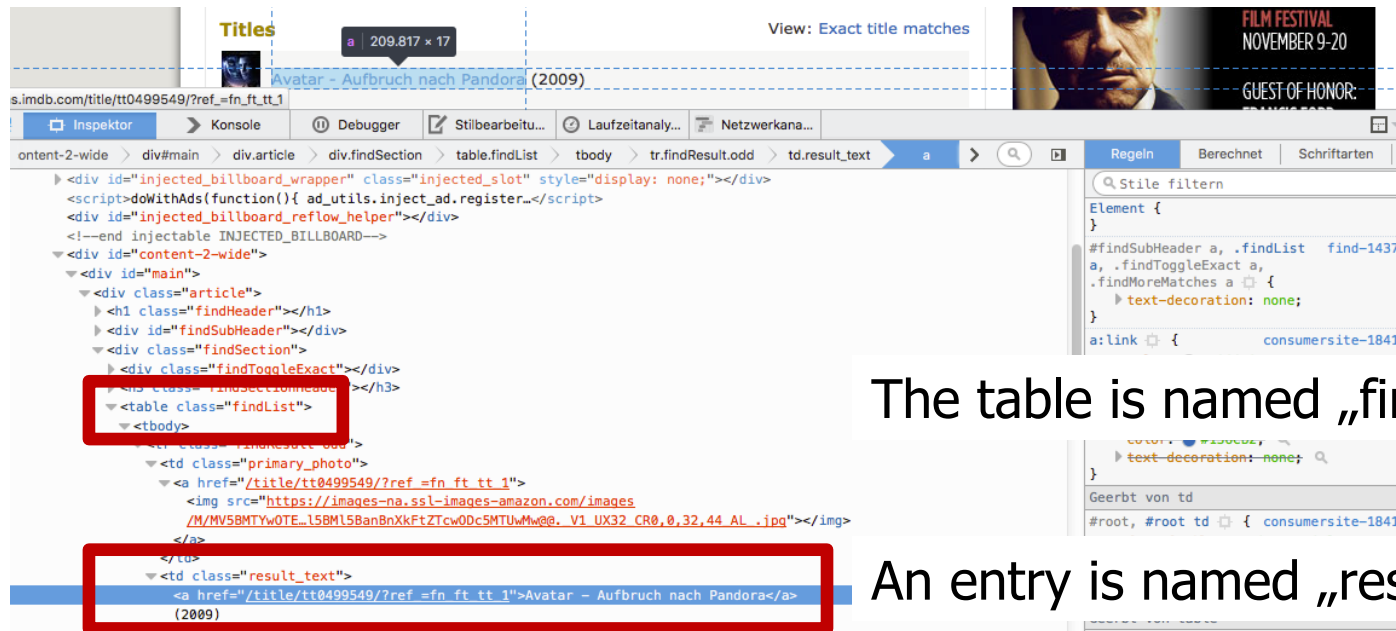
2. Perform a Web Search on IMDB

- Implement IMDBSpider.java that opens the URL:
`http://akas.imdb.com/find?q=<MOVIE>&s=tt&ttype=ft`
- From the results, extract the first element and its URL.
- Use URL encoding of movie titles.



2. Perform a Web Search on IMDB

- You have to parse the html file to extract the URL.
- You can use any method. You could use XPATH and html cleaner:
 - <http://htmlcleaner.sourceforge.net>
 - <http://htmlcleaner.sourceforge.net/doc/org/htmlcleaner/XPather.html>



The screenshot shows the IMDb page for the movie "Avatar - Aufbruch nach Pandora" (2009). The browser's developer tools are open, displaying the HTML structure. A red box highlights the `<table class="findList">` element, and another red box highlights the `<td class="result_text">` entry. The right sidebar shows the "Stile filtern" (Filter styles) panel with a search for "findList".

The table is named „findList“

An entry is named „result_text“

2. Perform a Web Search on IMDB

- XPATH is a syntax for navigating parts of an XML document.
- Has a directory-path-like syntax.
- ```
<table class= "list" >
 <tr>
 <TD class = "result">Avatar</TD>
 </tr>
</AAA>
```
- XPATH:  

```
/table[@class='list']//td[@class='result']/text()
```

=> Avatar



# 3. Extract Metadata from Movie's URL

The image displays three browser screenshots of the IMDb page for the movie "Avatar" (2009). The first screenshot shows the movie's main page, including the title "Avatar - Aufbruch nach Pandora (2009)", a 7.9 rating, and a synopsis. The second screenshot shows the "Cast" section, listing actors and their roles. The third screenshot shows the "Storyline" section, providing a detailed plot summary.

**Avatar - Aufbruch nach Pandora (2009)**

7.9  
12 24.42m Action, Adventure, Fantasy 17 December 2009 (Germany)

**Director:** James Cameron  
**Writer:** James Cameron  
**Stars:** Sam Worthington, Zoe Saldana, Sigourney Weaver

**Metascore:** 83  
**Reviews:** 3,063 user 720 critic  
**Popularity:** 332

**Cast**

Actor	Role
Sam Worthington	Jake Sully
Zoe Saldana	Neytiri (as Zoe Saldana)
Sigourney Weaver	Dr. Grace Augustine
Stephen Lang	Colonel Miles Quaritch
Michelle Rodriguez	Trudy Chacón
Giovanni Ribisi	Parker Selfridge
Joel David Moore	Norm Spellman
CCH Pounder	Moat
Wes Studi	Eytukan
Laz Alonso	Tsu'tey
Dileep Rao	Dr. Max Patel
Matt Gerald	Corporal Lyle Wainfleet
Sean Anthony Moran	Private Pike
Jason Whyte	Cryo Vault Med Tech
Scott Lawrence	Venture Star Crew Chief

**Technical Specs**

**Runtime:** 162 min | 171 min (special edition) | 178 min (extended cut)  
**Sound Mix:** Dolby Digital | DTS | SDDS | Sonics-DDP (IMAX version)  
**Color:** Color  
**Aspect Ratio:** 1.78 : 1

**Did You Know?**

**Trivia**  
Grace Augustine was originally named Shipley in earlier drafts.

**Goofs**  
In the colonel's robot there is a rear-view mirror. When we see the robot from the outside, it is very close to his head. But when we see shots from inside the robot, there is plenty of room around his head and we don't see the mirror.

**Quotes**  
[first lines]  
Jake Sully: [Narrating] When I was lying in the V.A. hospital with a big hole blown through the middle of my life, I started having these dreams of flying. I was free. But sooner or later, you always have to wake up.

**Crazy Credits**  
The initial end credits soar over the world of Pandora.

**Connections**  
Referenced in *La última película* (2013)

**Soundtracks**  
I See You (Theme from Avatar)  
Performed by Leona Lewis  
Music by James Horner and Simon Franglen  
Lyrics by Simon Franglen, Kuk Harrell, and James Horner  
Produced by Simon Franglen and James Horner  
Leona Lewis performs courtesy of Syco Music

**Storyline**

When his brother is killed in a robbery, paraplegic Marine Jake Sully decides to take his place in a mission on the distant world of Pandora. There he learns of greedy corporate figurehead Parker Selfridge's intentions of driving off the native humanoid "Na'vi" in order to mine for the precious material scattered throughout their rich woodland. In exchange for the spinal surgery that will fix his legs, Jake gathers intel for the cooperating military unit spearheaded by gung-ho Colonel Quaritch, while simultaneously attempting to infiltrate the Na'vi people with the use of an "avatar" identity. While Jake begins to bond with the native tribe and quickly falls in love with the beautiful alien Neytiri, the restless Colonel moves forward with his ruthless extermination tactics, forcing the soldier to take a stand - and fight back in an epic battle for the fate of Pandora.

### 3. Extract Metadata from Movie's URL

---

- Extract the following information from each movie and store it to a separate JSON file:  
url, title, year, genreList, countryList, description, budget, gross, ratingValue, ratingCount, duration, castList, characterList, directorList.
- Treat each attribute as a String and list names refer to JSON lists. Stick to exactly these names!
- Refer to example\_movie\_avatar.json for an example.

```
[{ "url":"http://akas.imdb.com/title/tt0499549/?ref_=fn_ft_tt_1",
 "title":"Avatar - Aufbruch nach Pandora (2009)",
 "year":"2009",
 "genreList":["Action","Adventure", "Fantasy", "Sci-Fi"], ...
 }
]
```

## 4. Easy Queries

---

- You have to correctly implement (at least) **three** basic queries out of:
  1. All-rounder: Determine all movies in which the director stars as an actor (cast). Return the top ten matches sorted by decreasing IMDB rating.
  2. Under the radar: Determine the top ten US-American movies until (including) 2015 that have made the biggest loss despite an IMDB score above (excluding) 8.0, based on at least 1,000 votes. Here, loss is defined as budget minus gross.
  3. The pillars of storytelling: Determine all movies that contain both (sub-)strings "kill" and "love" in their lowercase description (`String.toLowerCase()`). Sort the results by the number of appearances of these strings and return the top ten matches.
  4. The red planet: Determine all movies of the Sci-Fi genre that mention "Mars" in their description (case-aware!). List all found movies in ascending order of publication (year).
  5. Colossal failure: Determine all US-American movies with a duration beyond 2 hours, a budget beyond 1 million and an IMDB rating below 5.0. Sort results by ascending IMDB rating.

## 4. Harder Queries (Aggregation & Join)

---

- You have to correctly implement (at least) **two** queries out of:
  6. Uncreative writers: Determine the ten most frequent character names of all times ordered by frequency of occurrence. Filter any name containing "himself", "doctor", and "herself" from the result.
  7. Workhorse: Provide a ranked list of the top ten most active actors (cast), i.e., those actors which have starred in most movies.
  8. Must see: List the best rated movie of each year starting from 1990 until (including) 2010 with more than 10,000 ratings. Order the movies by increasing year.
  9. Rotten Tomatoes: List the worst rated movie of each year starting from 1990 till (including) 2010 with an IMCB score larger than 0. Order the movies by increasing year.
  10. Magic Couples: Determine those couples that feature together in the most movies. I.e., Adam Sandler and Allen Covert feature together in multiple movies; Report the top 10 pairs of actors and sort the result by the number of movies.

## 4. Optional: Custom Queries

---

- Come up with a fancy custom query.
- Give a text description of the query, the implementation and the result.

# Caveats

---

- Crawler:
  - You must implement the JAVA class `IMDBSpider.java`, which reads the movie titles from a JSON file and stores each movie into a separate JSON file.
- Queries:
  - You must implement five queries in `IMDBQueries.java`.
  - Optional Custom Query: You can implement one fancy custom query. Give a description of the query, source code and the result.
  - A query counts as implemented if it is correct. So, implement more than five to be sure.
- **Your two jars must be ready to run on GRUENAU2 (Java 8)**

# Competition

---

- Queries should not only be correct but as fast as possible.
- While you have 500 movies, we will execute your queries with 5000+ movies.
- Evaluation:
  - A correctly implemented query.
  - Bonus for faster implementation.
- We determine the best 5 teams.

# Deliverables

---

- **By Thursday, 17.11., 23:59 (midnight)**
  - Two-and-a-half weeks
- Send zipped assignment by **mail** to [buxmarcn@informatik.hu-berlin.de](mailto:buxmarcn@informatik.hu-berlin.de) and [patrick.schaefer@hu-berlin.de](mailto:patrick.schaefer@hu-berlin.de)
- Submission: JAVA source codes, libraries, and **two** executable JARs
  - IMDBSpider must be callable with  
`java -jar IMDBSpider.jar movies.json <moviesDir>`
  - IMDBQueries must be callable with  
`java -jar IMDBQueries.jar <moviesDir>`



# Presentation of Solutions

---

- You are be able to pick when and what you'd like to present (first-come-first-served):

Monday:

[https://dudle.inf.tu-dresden.de/inforet\\_ue1/](https://dudle.inf.tu-dresden.de/inforet_ue1/)

Tuesday:

<https://dudle.inf.tu-dresden.de/w2hvbhdi/>

- Presentation has to be given on 21.11./22.11..

# Next Week (Attendance Optional)

---

- Q/A session for assignment 1 (as every week).
- Live coding session. Getting you started with:
  - Eclipse,
  - JSON parsing,
  - Opening URLs, and
  - XPATH,
  - Executable Jars.