

Quiz/Assignment 4 --- C&C Views & REST Connectors

Objective

Create a component and connector view and a deployment view to address a statement of concern. Implement a REST connector prototype and parse returned JSON. Work in groups using chosen programming language.

Scenario

You were just given a contract to develop a new system: The Movie Writers Guild desires a system to store and annotate movie listings (e.g., poster images, descriptions, cast, etc) with blog posts and critic reviews to help users better understand and appreciate the art of movie making. The system will have mobile and web interfaces for registered MWG members to post reviews and other related content. It will also contain mobile and web interfaces for public users to register and interact with the content provided (e.g., post comments). Movie information will be obtained from a third party repository, The Movie Database. The system will store all movie data retrieved from the movie database, registered user information, and content provided by MWG members. The system will be cloud-based (i.e., hosted by a separate service provider [AWS, Microsoft Cloud]) and maintained by the MWG after deployment. The cloud-based system must have separate web server (e.g., Nginx), application server (e.g., Apache Tomcat, Node.js), and database as they hope to duplicate the system with a sister organization, The TV Writers Guild. The Movie Database is accessible using the REST framework and will return JSON which will be the primary source for the site's third-party content. The MWG wants you to develop an initial proof-of-concept to access and search relevant information from The Movie Database.

Requirements

Create Views

- Client-Server / Service-Oriented Architecture Component and Connector View for the Planned System
 - Indicate all required component types and connector types (protocols) used for the full system implementation
- Create a Deployment View
 - Indicate all required component types and connector types (protocols) used for the full system implementation
- Provide a brief overview for each view with design choice rationale

Implement a REST Connector to test functionality (retrieval and search)

- Use specified programming language to access The Movie Database for “Top Rated”, “Now Playing”, and “Upcoming” movie data
- API Key used to send request to TMDB: 9d3c8941bb5a7d5abef3326b3cd2cab8
- Create a client that allows the user to request 1 of the 3 categories of content
- Provide a search feature to search for matching strings (case ignored) within the movie title or overview

Constraints

- Programming language:
 - Python, Java, javascript, Ruby, Go, C++, C#/.Net, Swift, Groovy, Scala, Kotlin (others with prior approval)

Sample Prototype Workflow

- User launches TMDB REST client
- Client provides the user with a choice of content (Top Rated, Now Playing, Upcoming)

- User selects content type
- System returns complete list of movie titles, release date, and poster images (displayed for GUI clients)
- User can enter a string to search movie titles and descriptions
- System Returns all movies (title, release date, poster, and description / overview) where the title or overview matches the search

Lessons learned

- Discuss lessons learned from using REST / JSON and TMDB API
 - Challenges, programming language choice (simple or complicated)
- Identify five other public REST APIs and List (web site link, documentation link, sample API request [url])

Submission

- Submit source code files as single compressed zip file
- Submit PDF file containing Views, overview/rationale, lessons learned, and list of other REST APIs
- Instructions to execute code and links for any required frameworks / libraries

NOTES

- The Movie Database
 - API Documentation - <https://www.themoviedb.org/documentation/api>
 - Sample request:
https://api.themoviedb.org/3/movie/now_playing?api_key=9d3c8941bb5a7d5abef3326b3cd2cab8
- Help / Learning Resources
 - Beginners Guide to REST - <https://goo.gl/fXnNJZ>
 - JSON overview - <https://goo.gl/FMcy9I>
 - Helpful to install the JSONView Chrome Extension to view the returned JSON more easily
 - <https://goo.gl/IoR7pI>

Grading

- Counts for 2 Assignment / Quiz Grades
- Criteria: consistency between views, views contain clear indication of component and connector types with a key, all required functionality shown in the views, working sample application, presentation of lessons learned and other supporting info
- **Due Tuesday, October 25 @ 3:30pm**