

Vanier College
Faculty of Science and Technology
Web Services
420-511-VA

Term Project

Resources

Hamadi, Karine

Kallas, Christina

Tiongson, Devrin

October 14th, 2022

Table of Contents

Table of Contents	2
Resources	3
Resource List	3
URIs	3
CRUD operations	4
Filtering	5
Tasks	6
1- Web Framework Setup	6
2- Database Setup	6
3- Backend Creation - Resource Links	6
4- Backend Creation - Pagination	6
5- Backend Creation - Input Handling	6
6- Backend Creation - UI	7

Resources

Resource List

The resources that will be featured within our term project are listed below:

- games
- users
- properties (linked to the owned item table)
- studios
- reviews
- requests (linked to the trade requests table)
- wishlists (linked to the wishlist item table)
- gts (linked to the gts item table)

URIs

Below is a list of URIs that relate to the users resource:

- /users/{user}
- /users/{user}/wishlist
- /users/{user}/properties
- /users/{user}/gts
- /users/{user}/requests
- /users/{user}/reviews

The only URI related to the properties resource is featured below:

- /properties/{ownedid}

Another resource that only has one URI associated to it is the wishlists resource:

- /wishlists/{wishlistid}

The studios resource has a number of URIs associated to it, listed here:

- /studios/
- /studios/{studio}
- /studios/{studio}/games

The final resource that has a number of URIs to expose it is the games resource:

- /games/
- /games/{game}
- /games/{game}/reviews
- /games/{game}/boxart

CRUD operations

URI	Operations	Description
/users/{user}	create/update/delete	Points to a single user
/users/{user}/wishlist	create/update/delete	Points to an item in a user's wishlist
/users/{user}/properties	create/update	Points to all of the owned items of a user
/users/{user}/gts	create/delete	Points to all of the trade offers of a user
/users/{user}/requests	create/delete	Points to all of the trade requests of a user
/users/{user}/reviews	create	Points to all of the game reviews of a user
/properties/{ownedid}	create/update/delete	Points to a single owned item of a user
/wishlists/{wishlistid}	create/delete	Points to a single wishlist item of a user
/studios/		Points to a list of all game studios
/studios/{studio}	create/update/delete	Points to a single game studio
/studios/{studio}/games	create/update	Points to a list of games made by one studio
/games/		Points to a list of all games
/games/{game}	create/update/delete	Points to a single game
/games/{game}/reviews	create/update	Points to a list of reviews of a single game
/games/{game}/boxart	create/update	Points to the boxart of a single game

Filtering

URI	Filter	Description
/users/{user}	N/A	
/users/{user}/wishlist	gameId, gameName	Finding an item in your wishlist by gameId or gameName
/users/{user}/properties	ownedId, isForTrade	Finding the owned item through ownedId and checking what a user has for trade with isForTrade
/users/{user}/gts	gameId, gameName	Finding a trade item based on its Id or name
/users/{user}/requests	userId, gtsId	Finding requests related to a user or finding a request pertaining to a certain GTS item
/users/{user}/reviews	PosOrNeg, gameId	Finding a review based on the game it is reviewing and whether it is positive or negative
/properties/{ownedid}	isForTrade, gameId, userId	Finding a list of owned items based on whether or not it is for trade, based on the user who owns the items or based on the game ID
/wishlists/{wishlistid}	userId	Finding the items a user has listed in their wishlist
/studios/	Developer, Publisher, location	Finding all studios pertaining to a specific developer, publisher or location
/studios/{studio}	N/A	
/studios/{studio}/games	GameId, GameName	Finding a game by its gameId or Name within a specific studio

/games/	GameName, MPA Rating, Platform, GameStudioId	Finding all games and filtering them through their names, MPA rating, platform or game studio
/games/{game}	N/A	
/games/{game}/reviews	posOrNeg,	Finding all reviews of a game based on whether they are positive or negative
/games/{game}/boxart	N/A	

Tasks

1- Web Framework Setup

- Creation of the base project structure
Requirements: Apache web services
Assigned to: Karine Hamadi
Dependencies: N/A
Priority Level: 1

2- Database Setup

- Creation of a MySQL database, where relevant service data will be read from
Requirements: PHPmyAdmin
Assigned to: Christina Kallas
Dependencies: 1
Priority Level: 2

3- Backend Creation - Resource Links

- Creation of URIs to our resources within the database
Requirements: Slim
Assigned to: Devrin Tiongson
Dependencies: 2
Priority Level: 3

4- Pagination

- Pagination of resources onto a web page
Requirements: Slim, php, bootstrap
Assigned to: Christina Kallas
Dependencies: 3
Priority Level: 5

5- Input Handling

- Handling user input (eg. account creation...)
Requirements: Slim, php, bootstrap
Assigned to: Karine Hamadi
Dependencies: 2
Priority Level: 4

6- Frontend UI

- Creation of a MySQL database, where relevant service data will be read from
Requirements: Slim, php, bootstrap, html+css
Assigned to: Devrin Tiongson, Christina Kallas
Dependencies: 3, 4, 5
Priority Level: 6