ReactPlay Restfuldb API

POST /login

Related View restviews.login

Description

Users can log in by submit their username and password. If the user already logged in, return their information.

Parameters

Name	Field Type	Restriction
username	Char	Not null
password	Char	Not null

Responses

Code	Status	Situation	Data	Description
100	CONTINUE	Logged in user	status (logged-in-user), desc, userinfo (userid, username, email, role)	
200	OK	Anonymous user	status (**success**), desc, userinfo (userid, username, email, role)	Log in successfully.
400	BAD REQUEST	Anonymous user	status (failure), desc	
401	UNAUTHORIZED			

POST /register

Related View restviews.register

Description

Users can register by submit the username, password, email, and role. If the user already logged in or register successfully, return their information.

Parameters

Name	Field Type	Restriction
username	Char	Not null
password	Char	Not null
email	Char	
role	Char	UserPlayer / UserDeveloper

Responses

Code	Status	Situation	Data	Description
100	CONTINUE	Logged in user	status (logged-in-user), desc, userinfo (userid, username, email, role)	
201	CREATED	Anonymous user	status (**success**), desc, userinfo (userid, username, email, role)	Register successfully.

400	BAD REQUEST	Anonymous user	status (failure), desc	

POST /logout

Related View restviews.logout

Description

Users are able to logout.

Parameters

Nothing needs to input.

Responses

Code	Status	Situation	Data	Description
200	OK		status (**success**), desc	Log out successfully.
400	BAD REQUEST	Unknown error	status (failure), desc	

GET /users

Related View restviews.all_users

Description

Only the super user can view the list of all users, other users have no permission.

Parameters

Nothing needs to input.

Responses

Code	Status	Situation	Data	Description
200	OK	Super user	status (**success**), desc, userlist	Get list for all usernames.
401	UNAUTHORIZED	Other users	status (failure), desc	No permission.

GET /users/[userid]

Related View restviews.user_info

Description

User can view their own information, but not allowed to view other people’s account detail.

Parameters

Name	Type	Restriction
userid	*Integer*	*RESTful. Input by url, not by request.*

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user	status (**success**), desc, userinfo (userid, username, email, owned_games (gameheader (gameid, gamename, author), purchase_date, score), role)	Get their own user information
400	BAD REQUEST	Unknown user	status (failure), desc	User doesn't exist
401	UNAUTHO RIZED	Logged in user	status (failure), desc	Does not request for their own account. No permission.

GET /users/[userid]/games/[gameid]

Related View restviews.usergames

Description

Logged in user can view their own game states, and it's only for the game they owned. Not allowed to view other people's states. Super user has the highest authority, who can access to everyone's state.

Parameters

Name	Type	Restriction
userid	*Integer*	*RESTful. Input by url, not by request.*
gameid	*Integer*	*RESTful. Input by url, not by request.*

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user or super user	status (**success**), desc, usergame_info (userid, gameid, purchase_date, score, state)	Get all UserGame information.
400	BAD REQUEST	Unknown error	status (failure), desc	
401	UNAUTHO RIZED	Anonymous user	status (failure), desc	No permission.
		Logged in user	status (failure), desc	Does not own the game, or does not request for their own account. No permission.

POST /users/[userid]/games/[gameid]

Related View restviews.usergames

Description

User can modify their own game states, and it's only for the game they owned. Not allowed to modify other people's states. Super user has the highest authority, who can access to everyone's state.

Parameters

Name	Field Type	Restriction
score	Integer	Default 0
states	Text	

** No need to input all of them*

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user or super user	status (**success**), desc	Create/Update UserGame information
400	BAD REQUEST	Unknown error	status (failure), desc	
401	UNAUTHORIZED	Anonymous user	status (failure), desc	No permission.
		Logged in user	status (failure), desc	Does not own the game, or does not request for their own account. No permission.

GET /genres

Related View restviews.all_genres

Description

Show the list of all genre's id and name.

Parameters

Nothing needs to input.

Responses

Code	Status	Situation	Data	Description
200	OK		status (**success**), desc, genrelist (genreid, name)	Get list for all genres (id, name).
400	BAD REQUEST	Unknown error	status (failure), desc	

GET /games

Related View restviews.all_games

Description

Show the list of all game’s information to logged in user. Anonymous user has no permission.

Parameters

Nothing needs to input.

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user	status (**success**), desc, gamelist (gameid, name, author, price, description)	Get all game information.
401	UNAUTHORIZED	Anonymous user	status (failure), desc	No permission.

POST /games

Related View restviews.all_games

Description

Show the list of game information which filtered by game genres.

Parameters

Name	Type	Restriction
genre_ids	list	

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user	status (**success**), desc, gamelist (gameid, name, author, price, description)	Get all game information by genre filter.
401	UNAUTHORIZED	Anonymous user	status (failure), desc	No permission.

GET /games/[gameid]

Related View restviews.game_detail

Description

Show the game list. Anonymous user and the logged in user who doesn’t own the game cannot get the game URL. Logged in users and game author can access all game information.

Parameters

Name	Type	Restriction
gameid	*Integer*	*RESTful. Input by url, not by request.*

Responses

Code	Status	Situation	Data	Description
200	OK	Anonymous	status (**success**), desc,	Get game detail except

		user	game_detail (gameid, name, author, price, description)	url.
		Logged in user - Author	status (**success**), desc, game_detail (gameid, name, author, price, description, url)	Get all game details.
		Logged in user - Player	status (**success**), desc, game_detail (gameid, name, author, price, description, url)	Own the game. Get all game details.
			status (**success**), desc, game_detail (gameid, name, author, price, description)	Does not own the game. Get game detail except url.
400	BAD REQUEST	Unknown error	status (failure), desc	

POST /games/[gameid]

Related View restviews.game_detail

Description

Only the author of the game can update the game information.

Parameters

Name	Field Type	Restriction
name	Char	Max length 255, unique
price	Positive Integer	Default 0
description	Text	
url	URL	
gameid	*Integer*	*RESTful. Input by url, not by request.*

** No need to input all of them*

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user - Author	status (**success**), desc, game_detail (gameid, name, author, price, description)	Update the game detail.
401	UNAUTHORIZED	Other users	status (failure), desc	No permission.

GET /games/[gameid]/analytic

Related View restviews.game_analytic

Description

Collect relevant information from all related UserGames.

Parameters

Name	Type	Restriction
gameid	*Integer*	*RESTful. Input by url, not by request.*

Responses

Code	Status	Situation	Data	Description
200	OK	Logged in user	status (**success**), desc, info (score / purchase_date, purchase_price, score)	Will return (purchase_date, purchase_price, score) if the user is the author of the game or is super user.
401	UNAUTHORIZED	Anonymous user	status (failure), desc	No permission.

Appendix

Models.py

	Fields	Attributes
User	username, email, password, groups, is_staff, is_active, is_superuser, last_login	is_authenticated, is_anonymous
Group	name, permissions	
GameGenre	name	
Game	name, author, genre, price, description, url	set_price
UserGame	user, game, purchase_date, purchase_price, score, state	set_score, set_state

GameGenre	
name	Char, **unique**

Game	
name	Char, **unique**
author	ForeignKey (User / developedgames)
genre	ForeignKey (GameGenre / developedgames)
price	PositiveInteger (0)
description	Text
url	URL

UserGame	
user	ForeignKey (User / usergames)
game	ForeignKey (Game / usergames)
purchase_date	Date (now)
purchase_price	Integer (0)
score	Integer (0)
state	Text

Urls.py

	method	related views
users/	GET	all_users
users/[userid]/	GET	user_info
users/[userid]/games/[gameid]	GET/POST	usergames
genres/	GET	all_genres
games/	GET/POST	all_games
games/[gameid]/	GET/POST	game_detail

games/[gameid]/analytic/	GET	game_analytic
login	POST	login
register	POST	register
logout	POST	logout

Views.py

See *api.xlsx* sheet 3.

Created information

Group	
name	**permissions**
UserPlayer	add_usergame, change_usergame
UserDeveloper	add_game, change_game

User				
id	**name**	**password**	**email**	**role**
1	super	debugpass	nobody@nowhere.com	SuperUser
2	player	debugpass	nobody@nowhere.com	UserPlayer
3	developer	debugpass	nobody@nowhere.com	UserDeveloper

GameGenre	
id	**name**
1	Platform
2	Shooter
3	Fighting
4	Adventure
5	Survival

Game						
id	**name**	**author**	**genre**	**price**	**description**	**url**
1	test game	developer	genre	10	test description	http://notvalid.com
2	test game 2	developer	genre	20	test description 2	http://notvalid2.com

UserGame						
id	**user**	**game**	**purchase_date**	**purchase_price**	**score**	**state**
1	player	game	now	0	1337	