

CIS550 Milestone 4 Backend API Introduction

We divided the backend API into four parts, login page, main page, profile page, and game page for us to work together. And this is also the basic frontend structure we will follow on the client side. Below is the basic introduction for these:

Login Page

app.get('/login/:userid/getpassword', getpassword)
return the password given a user_id

Main Page

app.get('/mainpage', mainpage)
return random 10 games in mainpage
app.get('/mainpage/name/:gamename', getGameByname);
search game by accurate game name
app.get('/mainpage/date/:date', getGameByReleaseDate);
search game after a certain release date
app.get('/mainpage/lan/:language', getGameByLanguage);
search game by language
app.get('/mainpage/genre/:genre', getGameByGenre);
search game by genre
app.get('/mainpage/pos_rating/:positive_ratings', getGameByPositiveRatings);
search game sorted by positive_rating amounts
app.get('/mainpage/allsort', getGameByGenreLanguageReleaseDatePositiveRatings);
search game by genre, language, release date and sorted by positive rating
app.get('/allgenres', getAllGenres);
return all genres in the dataset
app.get('/mainpage/review_count', getGameByReviewsCount);
return games sorted by review_count
app.get('/mainpage/gsort/:genre', getGameByGenreRatingReviewCount);
return games in a certain genre, sorted by rating and review amount

Profile Page

app.get('/profile/:userid/owngame', getowngame)

return at most 10 games user owned

app.get('/profile/:userid/userrecommend', userrecommend);

recommend friends for user base on the game user reviewed, the type of game user played, and the location of the user.

app.get('/profile/:userid/gamerecommend', gamerecommend);

recommend games for the user, the algorithm first computes the game user reviewed to find a user list that may hold a similar taste of games. Then based on the user list, the algorithm found the game list that the user may like except the user owned. After filter the game type user liked, the final game list sort by the combination of positive and game review amounts.

Game Page

app.get('/game/:app_id', getGameInfo)

return the game description features.

app.get('/game/:app_id', getGameReview);

return the recommended reviews of the game.

app.get('/game/:app_id/gamerecommend', gameRecommend);

return at most of 10 similar games for user based on the current game type, rating and num of reviews.