DOKUMENTACJA PROJEKTU – NAUKA SZYBKIEGO PISANIA NA KLAWIATURZE

DOMINIK WIŚNIEWSKI | INFORMATYKA STOSOWANA V SEMESTR | INDEKS 113131

MICHAEL SZYCHULSKI | INFORMATYKA STOSOWANA V SEMESTR | INDEKS 113115

## Opis dostępnego API obsługiwanego przez backend w C#:

|  |  |  |
| --- | --- | --- |
| Składania zapytania: https://{localhost}:{port}/{model}/{x} Preambuła : https://{localhost}:{port}/ | Opis | Przykładowy wynik |
| Typ metody: GET  /Accounts | Pobranie wszystkich dostępnych kont | {  {  "id": 1,  "username": "testowy",  "password": "test",  "last\_logged": "2021-01-12T18:30:10.47843",  "created": "2021-01-12T18:30:10.480785"  },  {  "id": 2,  "username": "pierwszy",  "password": "TEST",  "last\_logged": "2021-01-12T18:58:11.251023",  "created": "2021-01-12T18:58:11.2531"  },  } |
| Typ metody: GET  /Accounts/find?id={id} | Pobranie konta po jego numerze id | {  "id": 1,  "username": "testowy",  "password": "test",  "last\_logged": "2021-01-12T18:30:10.47843",  "created": "2021-01-12T18:30:10.480785"  } |
| Typ metody: POST  /Accounts/login  Parametry JSON w body:  Username, password | Sprawdzenie czy dany użytkownik istnieje i czy jego hasło jest poprawne | Response 200 OK  Response 404 Not Found |
| Typ metody: POST  /Accounts/add  Parametry JSON w body:  Username, password | Dodanie nowe konta użytkownika do serwisu | Response 201 Created  Response 404 Not Found |
| Typ metody: DELETE  /Accounts/delete?id={id} | Usunięcie gracza o danym id | Response 200 OK Response 404 Not Found |
| Typ metody: PUT  /Accounts/update?id={id}  Parametry JSON w body:  Username, password | Aktualizacja danych użytkownika o danym id, zmiana jego nazwy i/lub hasła | Response 200 OK Response 404 Not Found |
| Typ metody: GET  /Games | Pobranie listy wszystkich dostępnych obiektów gier wraz z powiązanymi obiektami teksów, który zawiera dodatkowe informacje | {  "id": 1,  "gameName": "testGame",  "textToWrite": {  "id": 1,  "text": "testowy\_tekst\_do\_gry\_bo\_tak\_trzeba",  "source": "ja",  "wordCount": 1,  "topSpeed": 0,  "averageSpeed": 0  },  "wordCount": 0,  "minWordspeed": 20,  "maxMistakes": 0,  "difficulty": 0  },  {  "id": 7,  "gameName": "tryme7",  "textToWrite": {  "id": 2,  "text": "to jest tekst ktory musimy wrzucic do bazy danych ze spacjami.",  "source": "moja glowa",  "wordCount": 11,  "topSpeed": 0,  "averageSpeed": 0  },  "wordCount": 11,  "minWordspeed": 60,  "maxMistakes": 0,  "difficulty": 2  }  } |
| Typ metody: GET  /games/find?id={id} | Pobranie gry z powiązanym tekstem o podanym id | {  "id": 2,  "gameName": "tryme",  "textToWrite": {  "id": 2,  "text": "to jest tekst ktory musimy wrzucic do bazy danych ze spacjami.",  "source": "moja glowa",  "wordCount": 11,  "topSpeed": 0,  "averageSpeed": 0  },  "wordCount": 0,  "minWordspeed": 20,  "maxMistakes": 1,  "difficulty": 0  } |
| Typ metody: POST  /Games/add  Parametry JSON w body:  gameName, textToWrite, diff | Dodanie obiektu gry z jej danymi w body requestu, gdzie gameName to string z nazwą gry, textToWrite to int z id powiązanego tekstu i diff to trudność gry jako string, który zawiera się w enumie | Response 200 OK  Response 404 Not Found |
| Typ metody: PUT  /Games/update?id={id}  Parametry JSON w body:  gameName, textToWrite, diff | Aktualizacja obiektu gry o danym id z nowymi parametrami podanymi w body requestu, opisanymi w przypadku powyżej | Response 200 OK  Response 404 Not Found |
| Typ metody: GET  /results | Pobranie wszystkich wyników i powiązanych z nimi gier dla wszystkich użytkowników | {  "id": 10,  "game": {  "id": 2,  "gameName": "tryme",  "textToWrite": null,  "wordCount": 0,  "minWordspeed": 20,  "maxMistakes": 1,  "difficulty": 0  },  "wordSpeed": 87.93,  "account": {  "id": 3,  "username": "drugi",  "password": "test",  "last\_logged": "2021-01-12T18:59:22.412457",  "created": "2021-01-12T18:59:22.412477"  },  "finish\_date": "2021-01-17T01:13:20.739607",  "isPassed": true,  "mistakes": 0  },  {  "id": 11,  "game": {  "id": 8,  "gameName": "Lorem welcome",  "textToWrite": null,  "wordCount": 0,  "minWordspeed": 20,  "maxMistakes": 6,  "difficulty": 0  },  "wordSpeed": 44.58,  "account": {  "id": 3,  "username": "drugi",  "password": "test",  "last\_logged": "2021-01-12T18:59:22.412457",  "created": "2021-01-12T18:59:22.412477"  },  "finish\_date": "2021-01-17T01:38:25.282066",  "isPassed": true,  "mistakes": 0  } |
| Typ metody: GET  /Results/player?accId={Id} | Pobranie wszystkich wyników gier dla użytkownika o danym id | {  "id": 2,  "game": null,  "wordSpeed": 10,  "account": {  "id": 3,  "username": "drugi",  "password": "test",  "last\_logged": "2021-01-12T18:59:22.412457",  "created": "2021-01-12T18:59:22.412477"  },  "finish\_date": "2021-01-16T17:23:37.327277",  "isPassed": false,  "mistakes": 1  },  {  "id": 3,  "game": null,  "wordSpeed": 107.33,  "account": {  "id": 3,  "username": "drugi",  "password": "test",  "last\_logged": "2021-01-12T18:59:22.412457",  "created": "2021-01-12T18:59:22.412477"  },  "finish\_date": "2021-01-16T18:04:18.098837",  "isPassed": true,  "mistakes": 0  }, |
| Typ metody: GET  results/player/criteria  Parametry JSON w body: | Pobranie wszystkich wyników dla gracza o podanym id | {JSON} |
| Typ metody: GET  results/player/{playerId}/{isPassed} | Pobranie wszystkich wyników, które zostały zaliczone | {JSON} |
| Typ metody: GET  /results/player/{accId}/{date1}/{date2} | Pobranie wyników dla danego użytkownika, których data zakończenia gry znajduje się w przedziale <date1;date2> | Response 200 OK / Response 404 Not found |
| Typ metody: POST  /games/add/{gameName}/{textID}/{diff} | Utworzenie obiektu gry dla danego tekstu i podanej trudności | {JSON} / Response 404 Not found (?) {empty JSON} |
| Typ metody: POST  /writingText/add/{text}/{source} | Dodanie nowego obiektu tekstu do nauki pisania | Response 201 Created |
| Typ metody: PUT  /writingText/updateSpeeds/{id} | Aktualizacja TopSpeed i AvgSpeed dla WritingText o podanym ID | Response 201 Created / 200 OK (?) |
|  |  | Response 200 OK |
|  |  |  |