Спецификация интерфейсов.

\*- обязательное поле

Игра

1. **/game/ (GET)**
   1. Бэк получает:
      1. search – поиск по никнеймам пользователей
      2. status – статус игры (created//finished//in\_progress)
   2. На фронт уходит:
      1. id – id записи
      2. user\_1
         1. id – id пользователя 1
         2. username – никнейм пользователя 1
      3. user\_2
         1. id – id пользователя 2
         2. username – никнейм пользователя 2
      4. user\_1\_turn – идентификатор очереди хода пользователя 1
      5. user\_1\_win – идентификатор победы пользователя 1
      6. user\_1\_points – количество очков пользователя 1
      7. user\_2\_points – количество очков пользователя 2
      8. start\_at – дата и время начала игры
      9. finish\_at – дата и время конца игры
      10. status – статус игры (created//finished//in\_progress)
      11. moves – json поле, хранящее записи ходов игры после ее окончания
2. **/game/ (POST)**
   1. Бэк получает:
      1. user\_1 \* – id пользователя 1
      2. user\_2 \* – id пользователя 2
      3. user\_1\_turn – идентификатор очереди хода пользователя 1
      4. user\_1\_win – идентификатор победы пользователя 1
      5. user\_1\_points – количество очков пользователя 1
      6. user\_2\_points – количество очков пользователя 2
      7. start\_at – дата и время начала игры
      8. finish\_at – дата и время конца игры
      9. status – статус игры (created//finished//in\_progress)
      10. moves – json поле, хранящее записи ходов игры после ее окончания
   2. На фронт уходит:
      1. id – id записи
      2. user\_1 – id пользователя 1
      3. user\_2 – id пользователя 2
      4. user\_1\_turn – идентификатор очереди хода пользователя 1
      5. user\_1\_win – идентификатор победы пользователя 1
      6. user\_1\_points – количество очков пользователя 1
      7. user\_2\_points – количество очков пользователя 2
      8. start\_at – дата и время начала игры
      9. finish\_at – дата и время конца игры
      10. status – статус игры (created//finished//in\_progress)
      11. moves – json поле, хранящее записи ходов игры после ее окончания
3. **/game/{id}/ (GET)**
   1. Бэк получает:
      1. id \* -
   2. На фронт уходит:
      1. id \* – id записи
      2. user\_1 \* – id пользователя 1
      3. user\_2 \* – id пользователя 2
      4. user\_1\_turn – идентификатор очереди хода пользователя 1
      5. user\_1\_win – идентификатор победы пользователя 1
      6. user\_1\_points – количество очков пользователя 1
      7. user\_2\_points – количество очков пользователя 2
      8. start\_at – дата и время начала игры
      9. finish\_at – дата и время конца игры
      10. status – статус игры (created//finished//in\_progress)
      11. moves – json поле, хранящее записи ходов игры после ее окончания
4. **/game/{id}/ (PATCH)**
   1. Бэк получает:
      1. user\_1 – id пользователя 1
      2. user\_2 – id пользователя 2
      3. user\_1\_turn – идентификатор очереди хода пользователя 1
      4. user\_1\_win – идентификатор победы пользователя 1
      5. user\_1\_points – количество очков пользователя 1
      6. user\_2\_points –количество очков пользователя 2
      7. start\_at – дата и время начала игры
      8. finish\_at – дата и время конца игры
      9. status – статус игры (created//finished//in\_progress)
      10. moves – json поле, хранящее записи ходов игры после ее окончания
   2. На фронт уходит:
      1. id \* – id записи
      2. user\_1 \* – id пользователя 1
      3. user\_2 \* – id пользователя 2
      4. user\_1\_turn – идентификатор очереди хода пользователя 1
      5. user\_1\_win – идентификатор победы пользователя 1
      6. user\_1\_points – количество очков пользователя 1
      7. user\_2\_points –количество очков пользователя 2
      8. start\_at – дата и время начала игры
      9. finish\_at – дата и время конца игры
      10. status – статус игры (created//finished//in\_progress)
      11. moves – json поле, хранящее записи ходов игры после ее окончания
5. **/game/{id}/ (DELETE)**
   1. Передается id игры и все данные игры удаляются

Ход

1. **/move/ (GET)**
   1. Бэк получает:
      1. checker\_id – id шашки
      2. game – id игры
      3. user – никнейм пользователя
   2. На фронт уходит:
      1. id \* – id записи
      2. game \* – id игры, в которой был сделан ход
      3. user \* – id игрока, сделавшего ход
      4. checker\_id \* – id шашки, которой сходили
      5. new\_positions – координаты «пути» шашки относительно игрового поля в JSON формате
      6. is\_white – флаг цвета шашки
      7. is\_king – флаг дамки
2. **/move/ (POST)**
   1. Бэк получает:
      1. id \* – id записи
      2. game \* – id игры, в которой был сделан ход
      3. user \* – id игрока, сделавшего ход
      4. checker\_id \* – id шашки, которой сходили
      5. new\_positions – координаты «пути» шашки относительно игрового поля в JSON формате
      6. is\_white – флаг цвета шашки
      7. is\_king – флаг дамки
   2. На фронт уходит:
      1. id \* – id записи
      2. game \* – id игры, в которой был сделан ход
      3. user \* – id игрока, сделавшего ход
      4. checker\_id \* – id шашки, которой сходили
      5. new\_positions – координаты «пути» шашки относительно игрового поля в JSON формате
      6. is\_white – флаг цвета шашки
      7. is\_king – флаг дамки
3. **/move/{id}/ (GET)**
   1. Бэк получает:
      1. id \* - id записи
   2. На фронт уходит:
      1. id \* – id записи
      2. game \* – id игры, в которой был сделан ход
      3. user \* – id игрока, сделавшего ход
      4. checker\_id \* – id шашки, которой сходили
      5. new\_positions – координаты «пути» шашки относительно игрового поля
      6. is\_white – флаг цвета шашки
      7. is\_king – флаг дамки

Профиль

1. **/profile/ (GET)**
   1. Бэк получает:
      1. search – поиск по никнеймам пользователей
   2. На фронт уходит:
      1. id \* - id записи
      2. username \* - никнейм пользователя
      3. wins – количество побед
      4. games – количество сыгранных игр
      5. rating – рейтинг пользователя
2. **/profile/{id}/ (GET)**
   1. Бэк получает:
      1. id \* - id записи
   2. На фронт уходит:
      1. id \* - id записи
      2. username \* - никнейм пользователя
      3. wins – количество побед
      4. games – количество сыгранных игр
      5. rating – рейтинг пользователя
3. **/profile/{id}/ (PATCH)**
   1. Бэк получает:
      1. id \* - id записи
   2. На фронт уходит:
      1. wins – количество побед
      2. games – количество сыгранных игр
      3. rating – рейтинг пользователя

Пользователь

1. **/user/ (GET)** – вывод информации о пользователе, совершившем запрос
   1. На фронт уходит:
      1. id \* – id записи
      2. email \* – адрес эл. почты пользователя
      3. username \* – никнейм пользователя
      4. profile \*
         1. wins \* – количество побед
         2. games \* – количество сыгранных игр
         3. rating \* – рейтинг игрока
2. **/user/ (PUT)**
   1. Бэк получает:
      1. email \* – адрес эл. почты пользователя
      2. username \* – никнейм пользователя
      3. password \* – пароль пользователя
   2. На фронт уходит:
      1. id \* - id записи
      2. email \* – адрес эл. почты пользователя
      3. username \* – никнейм пользователя
      4. profile \*
         1. wins \* – количество побед
         2. games \* – количество сыгранных игр
         3. rating \* – рейтинг игрока
3. **/user/ (PATCH)**
   1. Бэк получает:
      1. email \* – адрес эл. почты пользователя
      2. username \* – никнейм пользователя
      3. password \* – пароль пользователя
   2. На фронт уходит:
      1. id \* - id записи
      2. email \* – адрес эл. почты пользователя
      3. username \* – никнейм пользователя
      4. profile \*
         1. wins \* – количество побед
         2. games \* – количество сыгранных игр
         3. rating \* – рейтинг игрока
4. **/user/ (DELETE)**
   1. Передается id пользователя, и все его данные удаляются
5. **/user/login/ (POST)**
   1. Бэк получает:
      1. email \* – адрес эл. почты пользователя
      2. password \* – пароль пользователя
   2. На фронт уходит:
      1. id \* - id пользователя
      2. email \* – адрес эл. почты пользователя
      3. username \* – никнейм пользователя
      4. access\_token \* – токен доступа
      5. refresh\_token \* – токен обновления токена доступа
6. **/user/logout/ (POST)**
   1. Бэк получает refresh токен и заносит его в черный список. Это означает, что больше ни одному пользователю он присвоен не будет
7. **/user/refresh-token/ (POST)**
   1. На бэк поступает refresh токен, после чего бэк возвращает новый сгенерированный access токен
8. **/user/registration/ (POST)**
   1. Бэк получает:
      1. email \* – адрес эл. почты пользователя
      2. username \* – никнейм пользователя
      3. password \* – пароль пользователя
      4. password2 \* – дублирование пароля пользователя (как подтверждение правильности ввода)
   2. На фронт уходит:
      1. id \* - id пользователя
      2. email \* – адрес эл. почты пользователя
      3. username \* – никнейм пользователя