

```
@app.route('/register', methods=['POST'])
def register():
    parser = reqparse.RequestParser()
    parser.add_argument("login", required=True)
    parser.add_argument("password", required=True)
    parser.add_argument("name", required=True)
    args = parser.parse_args()
    session = db_session.create_session()
    check_user = session.query(User).filter(User.login == args['login'].strip()).first()
    if check_user:
        return jsonify({'result': 'FAIL', 'message': 'user with this login already exists'})
    user = User(
        name=args['name'],
        login=args['login'].strip(),
        password=generate_password_hash(args['password'])
    )
    session.add(user)
    session.commit()
    return jsonify({'result': 'OK'})
```

#метакомпетенции

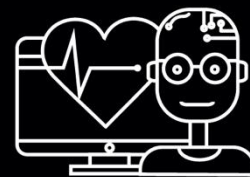
#MachineLivesMatters

#машины тоже чувствуют

#эмоциональный интеллект

#онлайн неприговор

#metavkd



```
@app.route('/login', methods=['GET'])
def login():
    parser = reqparse.RequestParser()
    parser.add_argument("login", required=True)
    parser.add_argument("password", required=True)
    args = parser.parse_args()
    session = db_session.create_session()
    user = session.query(User).filter(User.login == args['login'].strip()).first()
    if user and user.check_password(args['password']):
        return jsonify({'result': 'OK', 'message': {'api_key': user.generate_api_key()}})
    return jsonify({'result': 'FAIL', 'message': 'invalid login or password'})
```

#кружковое движение

#вызовы онлайн

#фронт будущего

#мемы стабильнее нашего кода

#ориентированность на результат

#MetaChallenge

#metateleport

#коммуникабельность

#практики будущего

```
@app.route('/vote_for_habit/<int:habit_id>', methods=['PUT'])
def vote_for_habit(habit_id):
    parser = reqparse.RequestParser()
    parser.add_argument("vote_type", required=True, choices=("positive", "negative"))
    parser.add_argument("api_key", required=True)
    args = parser.parse_args()
    session = db_session.create_session()
    habit = session.query(Habit).get(habit_id)
    user = User.get_by_api(args['api_key'])
    if user is None:
        abort(403, message='Invalid api key')
    if habit.user_id == user.id:
        return jsonify({'result': 'FAIL', 'message': 'cant vote on self habit'})
    if habit.voted_users is not None:
        already_voted = habit.voted_users.split(',')
        if str(user.id) in already_voted:
            return jsonify({'result': 'FAIL', 'message': 'already voted on this habit'})
        else:
            habit.voted_users += f', {user.id}'
    else:
        habit.voted_users = str(user.id)
    if args['vote_type'] == 'positive':
        habit.reputation += 1
    else:
        habit.reputation -= 1
    habit.votes += 1
    session.commit()
    return jsonify({'result': 'OK'})
```

```
@app.route('/habit_completed/<int:habit_id>', methods=['PUT'])
def habit_completed(habit_id):
    parser = reqparse.RequestParser()
    parser.add_argument("api_key", required=True)
    session = db_session.create_session()
    args = parser.parse_args()
    # weekday_today = str(datetime.datetime.today().weekday())
    habit = session.query(Habit).filter(Habit.id == habit_id).first()
    user = User.get_by_api(args['api_key'])
    if habit is None:
        abort(404, 'habit not found')
    if user is None:
        abort(403, 'Invalid api key')
    if habit.user_api_key != args['api_key']:
        abort(403, 'Invalid api key')
    # if weekday_today not in habit.weekdays:
    #     return jsonify({'result': 'FAIL', 'message': 'This habit is not scheduled for today'})
    user.change_data(rating=user.rating + habit.rating/len(habit.weekdays.split(',')), habit.votes, habit.reputation)
    return jsonify({'result': 'OK'})
```

# БЛАГОДАРНОСТЬ

Благодарим

Григория Сенченюк

за активное участие в проведении онлайн-мероприятия «Meta-Challenge»,  
которое прошло с 3 мая по 17 июня 2020 года.

Президент

Ассоциации участников технологических кружков

А.И. Федосеев