

Вводные данные, макет

Необходимо реализовать функционал регистрации, авторизации и заполнения профиля на платформе, а также выполнить верстку страниц. Frontend должен быть исполнен на React.

Страницы должны корректно отображаться в разрешении 1920 и 1360

Ссылка на макет Figma: <https://www.figma.com/file/9lraHzePI4lCHP7teGKipC/test?node-id=9%3A745>

Регистрация

API для взаимодействия с бекендом по шагам:

Шаг 1: Запрос на регистрацию:

```
POST https://test.it-planet.org/sso/signup
В теле запроса:

{
  "username": "mail@gmail.com",
  "password": "12345",
  "mailing": true
}
```

Ответ от сервера:

```
{
  "id": "1"
}
```

где id - идентификатор созданного пользователя. Также после выполнения данного запроса на указанный email будет отправлен код подтверждения регистрации, который понадобится на следующем шаге. При желании можно использовать сервисы, предоставляющие временные почтовые ящики, типа этого: <https://temp-mail.io>

Шаг 2: Подтверждение email:

```
POST https://test.it-planet.org/sso/signup/confirm
В теле запроса:

{"userId": 1, "confirmationCode": "865-857"}
```

где:

userId - id пользователя, полученный на шаге 1,

confirmationCode - код из письма, отправленного на email.

В случае успеха придет пустой ответ от сервера со статусом 200.

Шаг 3: Завершающий. Инициализация профиля.

Шаг 3.1: Для ФЛ:

В теле запроса:

}

Авторизация

Необходимо реализовать функционал авторизации на платформе, а также выполнить верстку страниц авторизации.

Для авторизации необходимо отправить запрос:

}

В случае успеха, придет следующий ответ:

}

token - JWT Token - необходимо сохранить в localStorage, чтобы в дальнейшем использовать в запросах (в заголовках), требующих авторизации.

Пример передачи заголовка:

Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJ.....

Пример передачи заголовка curl:

UqaANFhUdTNHdobgc3ZyBOTKtXGLa5kLRuQrsV6fUa-tVLIvfI69NorrFBv-SNpHRWTaUApp_XUVog'

Профиль - личные данные

Необходимо реализовать раздел “Личные данные”.

Важно: для всех действий с профилем требуется предварительная авторизация, т.е. в заголовке Authorization требуется передавать JWT токен в виде: Bearer token_here

Обновление информации в профиле

Для обновления информации необходимо отправить следующий запрос:

```
PUT: https://test.it-planet.org/user/profile/personal
```

```
{
  "firstname": "Jon",
  "lastname": "Snow",
  "patronymic": "",
  "sex": 1,
  "birthDate": "1905-06-21",
  "personStatusId": 1,
  "languageCode": "RU",
  "location": {
    "languageCode": "RU",
    "country": {
      "code": "US",
      "currencyCodes": [
        "USD"
      ],
      "name": "США",
      "wikiDataId": "Q30"
    },
    "region": {
      "countryCode": "US",
      "fipsCode": "42",
      "isoCode": "PA",
      "name": "Пенсильвания",
      "wikiDataId": "Q1400"
    },
    "city": {
      "id": 124815,
      "wikiDataId": "Q1134176",
      "type": "CITY",
      "city": "Калифорния",
      "name": "Калифорния",
      "regionCode": "PA",
      "countryCode": "Q30"
    }
  }
}
```

Где:

firstname - имя

lastname - фамилия

patronymic - отчество

sex - пол (0 - женский, 1 - мужской)

birthDate - дата рождения в формате YYYY-mm-dd

personStatusId - статус пользователя (студент/школьник и т.п.) равный id из словаря статусов (описан ниже),

location - структура, составленная из справочника получения стран/городов/областей (методы работы со справочником описаны ниже)

В случае успеха придет следующий ответ (**Важно:** в ответе будут поля, которые пока не описаны в задаче, просьба пока не обращать на них внимание):

```
{
  "createdAt": "2021-12-20T12:47:02.684+00:00",
  "updatedAt": "2021-12-27T13:57:51.587+00:00",
  "id": 3,
  "userId": 2,
  "login": "jsnow",
  "firstname": "Jon",
  "lastname": "Snow",
  "patronymic": "",
  "personStatus": 1,
  "sex": 1,
  "birthDate": "1905-06-21T00:00:00.000+00:00",
  "countryWikiDataId": "wiki1",
  "regionWikiDataId": "wiki1",
  "cityWikiDataId": "wiki1",
  "location": {
    "languageCode": "RU",
    "country": {
      "code": "US",
      "currencyCodes": [
        "USD"
      ],
      "name": "США",
      "wikiDataId": "Q30"
    },
    "region": {
      "countryCode": "US",
      "fipsCode": "42",
      "isoCode": "PA",
      "name": "Пенсильвания",
      "wikiDataId": "Q1400"
    },
    "city": {
      "id": 124815,
      "wikiDataId": "Q1134176",
      "type": "CITY",
      "city": "Калифорния",
      "name": "Калифорния",
      "regionCode": "PA",
      "countryCode": "Q30"
    }
  }
}
```

Получение профиля

Для получения профиля необходимо выполнить запрос:

```
GET https://test.it-planet.org/user/profile/personal
```

Получение стран

Для получения списка стран необходимо отправить следующий запрос:

```
POST https://test.it-planet.org/location/country
{
  "name": "Py",
  "languageCode": "RU"
}
```

Где:

name - название или часть названия страны

languageCode - код языка ответа (пока будем использовать только RU)

В случае успеха придет следующий ответ:

```
{
  "data": [
    {
      "code": "RO",
      "currencyCodes": [
        "RON"
      ],
      "name": "Румыния",
      "wikiDataId": "Q218"
    },
    {
      "code": "RW",
      "currencyCodes": [
        "RWF"
      ],
      "name": "Руанда",
      "wikiDataId": "Q1037"
    }
  ]
}
```

Где:

code - код страны

currencyCodes - коды валют страны

name - название страны

wikiDataId - идентификатор страны

Получение областей

Для получения областей необходимо отправить следующий запрос:

```
POST https://test.it-planet.org/location/region

{
  "name": "BO",
  "languageCode": "RU",
  "countryId": "Q159"
}
```

Где:

name - полное или частичное название области

languageCode - код языка ответа (пока используем только RU)

countryId - countryWikiDataId из ответа в шаге "Получение страны"

В случае успеха придет следующий ответ:

```
{
  "data": [
    {
      "countryCode": "RU",
      "fipsCode": "85",
      "isoCode": "VLG",
      "name": "Вологодская область",
      "wikiDataId": "Q2015"
    },
    {
      "countryCode": "RU",
      "fipsCode": "84",
      "isoCode": "VGG",
      "name": "Волгоградская область",
      "wikiDataId": "Q3819"
    },
    {
      "countryCode": "RU",
      "fipsCode": "86",
      "isoCode": "VOR",
      "name": "Воронежская область",
      "wikiDataId": "Q3447"
    }
  ]
}
```

Получение городов

Для получения городов необходимо отправить следующий запрос:

```
POST https://test.it-planet.org/location/city

{
  "name": "Воро",
  "languageCode": "RU",
  "countryId": "Q159",
  "regionCode": "VOR"
}
```

Где:

name - полное или частичное наименование города

regionCode - isoCode региона, из ответа по регионам

languageCode - код языка ответа (пока используем только RU)

countryId - countryWikiDataId из ответа в шаге "Получение страны"

В случае успеха, придет следующий ответ:

```
{
  "data": [
    {
      "id": 3471163,
      "wikiDataId": "Q4125245",
      "type": "CITY",
      "city": "Воробьёвка",
      "name": "Воробьёвка",
      "regionCode": "VOR",
      "countryCode": "Q159"
    },
    {
      "id": 163883,
      "wikiDataId": "Q4016457",
      "type": "ADM2",
      "city": "Воробьёвский район",
      "name": "Воробьёвский район",
      "regionCode": "VOR",
      "countryCode": "Q159"
    },
    {
      "id": 100102,
      "wikiDataId": "Q3426",
      "type": "CITY",
      "city": "Воронеж",
      "name": "Воронеж",
      "regionCode": "VOR",
      "countryCode": "Q159"
    },
    {
      "id": 100891,
      "wikiDataId": "Q4125792",
      "type": "CITY",
      "city": "Воронцовка",
      "name": "Воронцовка",
      "regionCode": "VOR",
      "countryCode": "Q159"
    }
  ]
}
```

Получение списка статусов (школьник, студент и т.п.)

Для получения списка статусов необходимо отправить следующий запрос:

```
GET https://test.it-planet.org/user/dict/person-status
```

В случае успеха, придет следующий ответ:

Навыки

Необходимо реализовать функционал создания/получения навыков.

1. Создание навыка

Для создания навыка необходимо выполнить:

```
POST https://test.it-planet.org/user/skill

{
  "name": "тестирование" // название навыка
}
```

В случае успеха придет респонс код 200. После этого, навык добавится в массив skills профиля пользователя

2. Получение навыков (в профиле)

Все созданные пользователем навыки добавляются в профиль, в поле skills, которое будет выглядеть так:


```

...
"skills": [
  {
    "createdAt": "2022-01-18T08:13:28.890+00:00",
    "updatedAt": "2022-01-18T08:13:28.890+00:00",
    "id": 1,
    "parentId": null,
    "isNative": false,
    "name": "Программирование"
  },
  {
    "createdAt": "2022-01-18T08:13:39.350+00:00",
    "updatedAt": "2022-01-18T08:13:39.350+00:00",
    "id": 2,
    "parentId": null,
    "isNative": false,
    "name": "Дизайн"
  },
  {
    "createdAt": "2022-01-18T08:13:56.179+00:00",
    "updatedAt": "2022-01-18T08:13:56.179+00:00",
    "id": 3,
    "parentId": null,
    "isNative": false,
    "name": "Рытье ям"
  },
  {
    "createdAt": "2022-01-18T08:14:14.704+00:00",
    "updatedAt": "2022-01-18T08:14:14.704+00:00",
    "id": 4,
    "parentId": null,
    "isNative": false,
    "name": "Вождение"
  },
  {
    "createdAt": "2022-01-18T08:14:22.574+00:00",
    "updatedAt": "2022-01-18T08:14:22.574+00:00",
    "id": 5,
    "parentId": null,
    "isNative": false,
    "name": "Проектирование"
  },
  {
    "createdAt": "2022-01-18T08:14:42.069+00:00",
    "updatedAt": "2022-01-18T08:14:42.069+00:00",
    "id": 6,
    "parentId": null,
    "isNative": false,
    "name": "Просто чтоб находилось по про*"
  }
]
...

```

3. Удаление навыка

Для удаления навыка необходимо отправить запрос:

```
DELETE https://test.it-planet.org/user/skill/{id} # id - айдишник скилла из профиля пользователя
```

В случае успеха придет респонс код 200

4. Подсказки по навыкам

Важно: не все добавленные навыки будут выводиться в подсказках, т.к. сначала должны пройти модерацию. В настоящее время для теста можно использовать названия навыков из примера ниже.

Для получения подсказок по навыкам необходимо выполнить запрос:

```
GET https://test.it-planet.org/user/skill?name=про # name – полное или частичное название
```

В случае, если навык найден, придет следующий ответ:

```
{
  "skills": [
    {
      "createdAt": "2022-01-18T08:13:28.890+00:00",
      "updatedAt": "2022-01-18T08:16:27.416+00:00",
      "id": 1,
      "parentId": null,
      "isNative": true,
      "name": "Программирование"
    },
    {
      "createdAt": "2022-01-18T08:14:22.574+00:00",
      "updatedAt": "2022-01-18T08:17:42.067+00:00",
      "id": 5,
      "parentId": null,
      "isNative": true,
      "name": "Проектирование"
    },
    {
      "createdAt": "2022-01-18T08:14:42.069+00:00",
      "updatedAt": "2022-01-18T08:17:55.058+00:00",
      "id": 6,
      "parentId": null,
      "isNative": true,
      "name": "Просто чтоб находилось по про*"
    }
  ]
}
```

Алгоритм добавления найденного скилла аналогичен созданию нового, т.е. передается имя как в п.1