Интеграционная карта

Интеграция BetaOnline со сторонней системой клиента с целью передачи кандидатов делится на два этапа. Первый - передача кандидата в систему клиента из системы BetaOnline и получение в ответ идентификатора этого кандидата в системе клиента. Второй - периодическое получение системой BetaOnline истории статусов переданного кандидата из системы клиента с использованием полученного идентификатора кандидата.

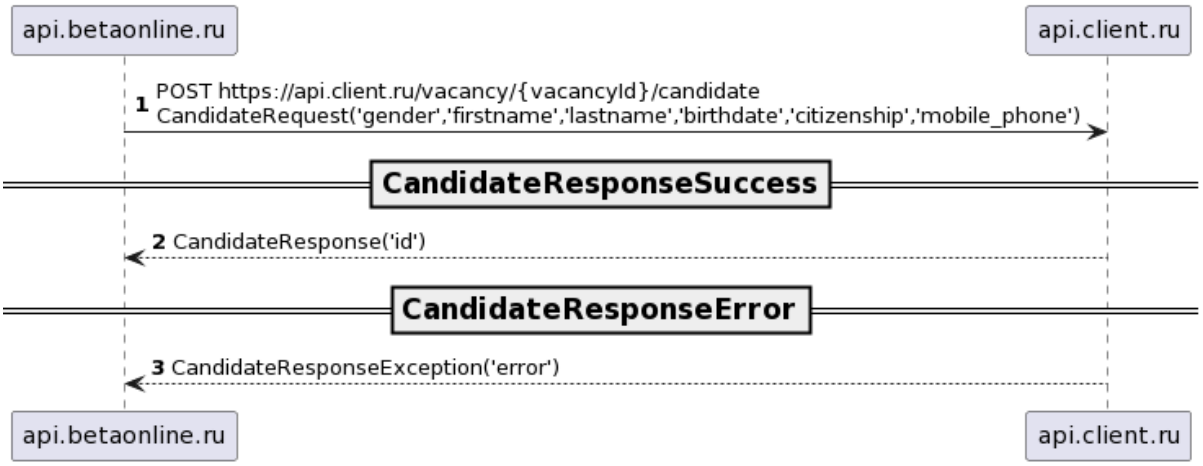
Под кандидатом понимается соискатель, откликнувшийся на вакансию клиента и прошедший проверки требований клиента, а так же дедуплицированный по требованиям клиента.

Передача кандидата

Клиент должен реализовать следующий endpoint:

POST https://api.client.ru/vacancy/{vacancyId}/candidate

Этот endpoint используется для отправки данных кандидата в систему клиента.



Пример запроса на отправку кандидата

```
curl -XPOST https://api.client.ru/vacancy/{vacancyId}/candidate -H "Content-Type: application/json" -d '
{
    "gender":"Мужской",
    "firstname":"Билл",
    "lastname":"Гейтс",
    "birthdate":"28.10.1955",
    "citizenship":"РФ",
    "mobile_phone":"+7 (000) 000-00-00"
}'
```

Пример ответа

```
{
    "id": "1fe8c9115933e31ac3ba3236"
}
```

Конечная точка для оправки кандидата

Конечная точка на которую производится отправка кандидата предоставляется системой клиента. Это может быть как uri по протоколу http, так и по любому другому протоколу удобному системе клиента. Единственными обязательными условиями к конечной точке являются наличие шифрования, аутентификации и возвращению ответов в строго согласованном формате. Например если это http то ответ со статусом 200 и телом сообщения содержащем идентификатор кандидата в системе клиента будет означать успех или 400 с явно описанной причиной ошибки.

Передаваемые данные

Система BetaOnline может предоставлять данные о кандидате которые необходимы клиенту. Возможные поля перечислены ниже.

Стандартизированные поля BetaOnline

Поле	**Кириллическое название**	**Тип**
Фамилия	lastname	Строка
Имя	firstname	Строка
Дата рождения	birthdate	Строка формата (d.m.Y) - пример 05.01.1987
Пол	gender	Строка. Два варианта (Мужской, Женский)
Гражданство	citizenship	Строка. Значения (РФ, Беларусь, Киргизия,

Казахстан, Армения, Другое)		
Номер телефона	mobile_phone	Строка формата +7 (000) 000-00-00
Наличие вод. уд.	available_drivers_license	Строка. Два варианта (Есть, Нет)
Открытые категории в. уд.	drivers_licenses_categories	Список. Значения(A, B, B1, C, D, E)
Опыт работы	work_experience	Строка. Значения (Более 1 года, Без опыта)
Адрес	address	Структуру адреса. См ниже.
Уровень образования	education_level	Список строк. Значения (Высшее, Бакалавр, Магистр,Кандидат наук, Доктор наук, Неоконченное высшее, Среднее, Среднее специальное, Курсы)
Наличие личного автомобиля	has_own_car	Строка. Два варианта (Есть, Нет)
Наличие мед.книжки	has_medical_card	Строка. Два варианта (Есть, Нет)
Наличие военного билета	has_military_id_card	Строка. Два варианта (Есть, Нет)
Судимость	criminal_record	Строка. Два варианта (Есть, Нет)

Адрес эл.почты	email	Строка. Формат addr@domain.domain
Наличие квалификационного удостоверения	has_qualification_certificate	Строка. Два варианта (Есть, Нет)
Произвольное	custom	
Страна	country	Строка. Если не указана то пустая строка
Регион	area	Строка. Если не указана то пустая строка
Населенный пункт	locality	Строка. Если не указана то пустая строка
Улица	street	Строка. Если не указана то пустая строка
Дом	house	Строка. Если не указана то пустая строка
Корпус	block	Строка. Если не указана то пустая строка
Индекс	postal_code	Строка. Если не указана то пустая строка
Широта	latitude	Строка

Долгота	longitude	Строка

Пользовательские поля

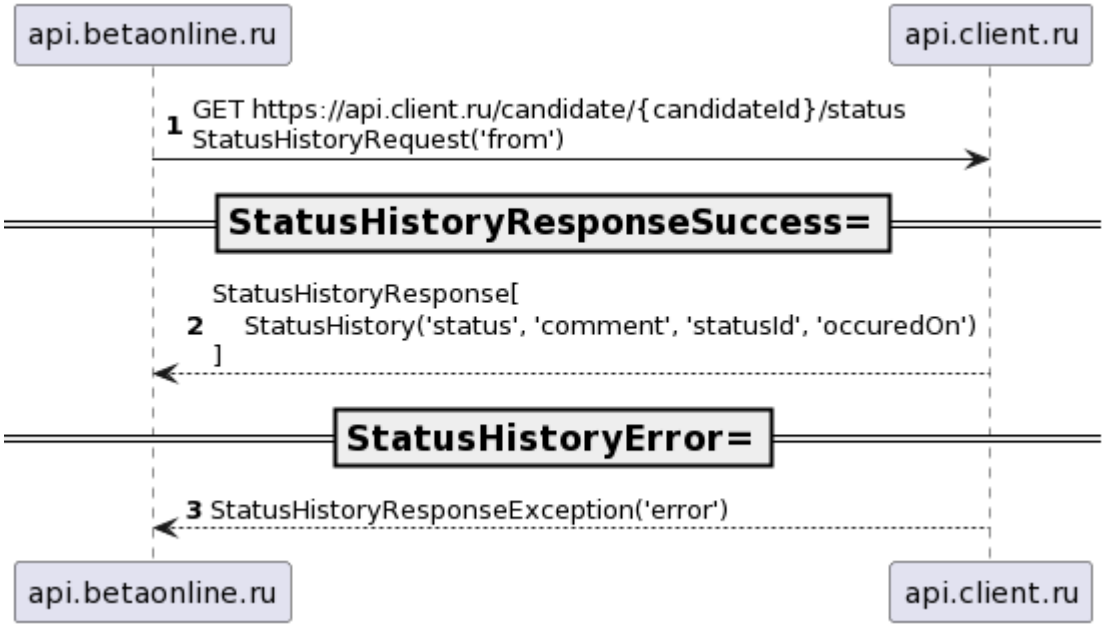
Бывают ситуации когда штатных полей не достаточно и добавляются пользовательские поля. При использовании пользовательских полей их кириллические имена транслитерируются для более удобного использования, например место работы транслитерируется в mesto-raboty. Значения этих полей бывают строковые, целочисленные, с плавающей точкой или списки с одним из этих типов.

Периодическое получение истории статусов переданного кандидата

Клиент должен реализовать следующий endpoint:

GET https://api.client.ru/candidate/{candidateId}/status

Этот endpoint используется для получения статуса кандидата.



Конечная точка для получения информации по кандидату

Конечная точка с которой производится получение информации по кандидату предоставляется системой клиента. Это может быть как uri по протоколу http, так и по любому другому протоколу удобному системе клиента. Единственными обязательными условиями к конечной точке являются наличие шифрования, аутентификации и возвращению ответов в строго согласованном формате. Например если это http то ответ со статусом 200 и телом сообщения содержащем список истории статусов кандидата в системе клиента будет означать успех или 400 с явно описанной причиной ошибки.

Пример запроса на информацию по кандидату

```
curl
https://api.client.ru/candidate/{candidateId}/status?from=2021-10-18T00:00:21+03:00
-H "Content-Type: application/json"
```

Пример ответа


```
{
  "51e76d84-b615-461e-861f-5d60554d9485": [
    {
      "status": "New",
      "comment": "",
      "statusId": 1,
      "occurredOn": "2021-10-18T00:16:21+03:00"
    },
    {
      "status": "InWork",
      "comment": "Созвон",
      "statusId": 2,
      "occurredOn": "2021-10-18T00:16:28+03:00"
    },
    {
      "status": "Rejected",
      "comment": "Хам, в черный список его!",
      "rejectionReason": "Грубо общался по телефону с интервьювером",
      "statusId": 3,
      "occurredOn": "2021-10-18T00:16:31+03:00"
    }
  ]
}
```


Справочники соответствия вакансий и статусов

Вакансий

Клиент должен реализовать следующие endpoints:

GET https://api.client.ru/vacancies

Этот endpoint используется для получения справочника вакансий.

Пример ответа

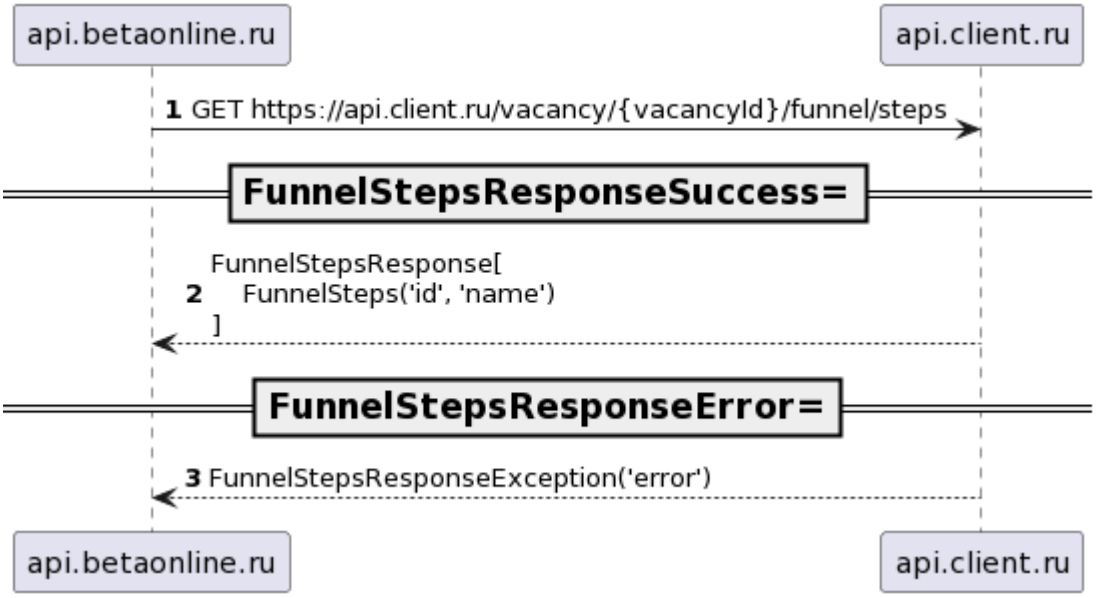

```
{
  "items": [
    {
      "id": "5fe5ceb2ce61359bf2c6781c",
      "name": "Водитель-курьер",
      "region": "Вологодская область",
      "city": "Череповец"
    },
    {
      "id": "5fe5ceb2ce61359bf2c6781c",
      "name": "Водитель-курьер",
      "region": "Вологодская область",
      "city": "Череповец"
    },
  ]
}
```


Справочник статусов воронки у вакансии

Клиент должен реализовать следующие endpoint:

GET https://api.client.ru/vacancy/{vacancyId}/funnel/steps

Этот endpoint используется для получения справочника статусов воронки у вакансии.



Пример запроса

```
curl https://API.client.ru/vacancy/11111111/funnel/steps -H "Content-Type: application/json"
```

Пример ответа

```
[
  {
    "id": 1,
    "name": "new"
  },
  {
    "id": 2,
    "name": "PhoneInterview"
  },
  {
    "id": 3,
    "name": "IncorrectNumber"
  },
  {
    "id": 4,
    "name": "Blacklist"
  }
]
```

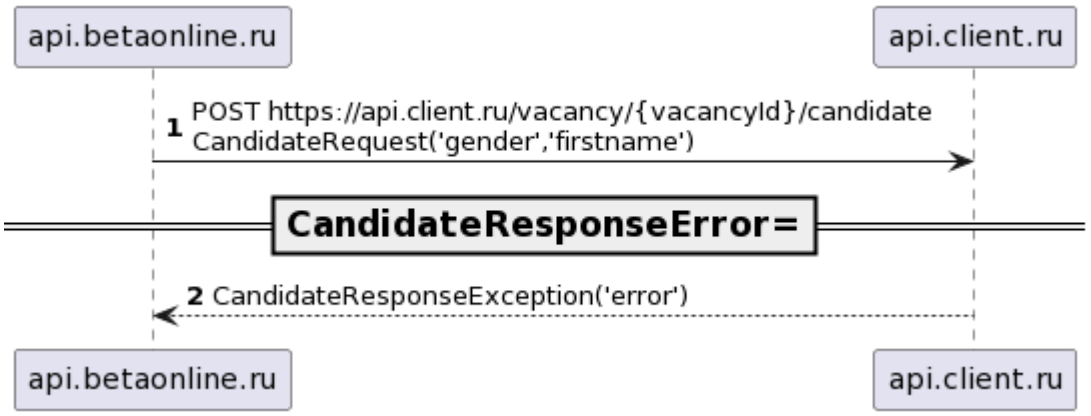
Дедупликация

Отправленный кандидат в систему клиента может оказаться дублем в системе клиента. Если такое происходит то при отправке кандидата необходимо передавать в ответ информацию о дубле.

Существует несколько возможных подходов к реализации получения информации о дубликате.

Возвращение ошибки при получении

При получении кандидата от системы BetaOnline, система клиента может ответить отличным от успеха ответом с указанием причины - duplicate. Этот подход показан на диаграмме ниже.



Пример зароса на отправку кандидата

```
curl -XPOST https://api.client.ru/vacancy/{vacancyId}/candidate -H "Content-Type: application/json" -d '
{
  "gender":"Мужской",
  "firstname":"Билл",
  "lastname":"Гейтс",
  "birthdate":"28.10.1955",
  "citizenship":"РФ",
  "mobile_phone":"+7 (000) 000-00-00"
}'
```

Ответом на такой запрос будет status code - 400 или любой другой строго описанный код ошибки.

Пример ответа

```
{
  "error": "duplicate"
}
```

Перевод кандидата в статус дубля

При получении кандидата от системы BetaOnline, система клиента может ответить успехом, а после проставить статус дубликата. Этот статус будет получен при периодическом получении статусов как описано в разделе "Периодическое получение истории статусов переданного кандидата".

Нефункциональные требования

RPS/RPM

Ко всем Endpoints должен быть обеспечен Requests Per Minute не менее 1000 (запросов в минуту)