**PROVISION\_API — 10.32.1.62:1239**

2021-04-14

Данный API построен для получения данных об обеспеченности населения сервисами. Некоторые моменты дублируют или предоставляют в другому виде функционал основного API Платформы. Все эндпоинты предполагают GET-запросы с указанием параметров в query.

Последние обновления направлены по большей части на то, чтобы добавить возможность по составлению произвольных запросов к базе данных (как временное решение в случае необходимости получения данных, для которых нет эндпоинта), а также на работу с файлами логов. Эти возможности отключаются передачей ключа --no\_db\_endpoints или установкой переменной окружения PROVISION\_DISABLE\_DB\_ENDPOINTS=1

После внесения изменений в схему базы данных было принято решение изменить понятие «район» на более общее «административная единица». На данный момент все эндпоинты сохранили свои интерфейсы, где используется ключ «district», но возможно их изменение в части интерфейса в виде переименования таких полей в «administrative\_unit».

Все эндпоинты, принимающие город в качестве параметра, при его отсутствии обращаются к городу по-умолчанию, установленному при запуске либо при вызове /api/reload\_data/

Кроме того, возможны переименования следующего рода для поддержания аналогии с базой:

* service\_type → city\_service\_type
* infrastructure → city\_infrastructure\_type

Список эндпоинтов с описанием:

**Разное:**

* /api — список эндпоинтов в формате HAL
* POST /api/reload\_data/ — недокументированный эндпоинт, заново загружает из базы информацию о домах, сервисах, обеспеченности и прочем. Временно принимает параметр city с именем города для загрузки данных по городу, отличному от Санкт-Петербурга.
* /api/db — получение списка таблиц в базе данных во всех схемах (доступен только в случае не-отключенных эндпоинтов базы данных)
* /api/db/{schema\_name} — получение списка таблиц в заданной схеме (доступен только в случае не-отключенных эндпоинтов базы данных)
* /api/db/{schema\_name}/{table\_name} — получение списка столбцов у заданной таблицы (доступен только в случае не-отключенных эндпоинтов базы данных)
* /api/db/?query={query}&format={csv/json/geojson/xlsx} — выполнение select-запроса в базу данных с домами, format устанавливает формат выходного файла, по-умолчанию — json. (доступен только в случае не-отключенных эндпоинтов базы данных)
* /api/logs — получение последнего файла логов (файл до 10Мб, создается только в случае не-отключенных эндпоинтов базы данных, доступен аналогично)
* /api/logs/<command> — расширенная работа с логами (только в случае не-отключенных эндпоинтов базы данных):
  + list — получение списка названий файлов логов
  + all — получение полного лога (объединение всех файлов)
  + (число) — получение файла лога под данным номером, где 0 — первый файл; также можно использовать отрицательную адресацию, аналогично Python’у (-1 — последний файл)
  + (название) — получение файла лога с названием provision\_api.(название).log

**Получение параметров значимости и интенсивности использования сервисов социальными группами (необходимость под вопросом):**

* /api/relevance/social\_groups — возвращает список социальных групп с указанием важности, если задан тип сервиса, и с указанием интенсивности использования, если заданы тип сервиса и жизненная ситуация. Без параметров возвращает просто список социальных групп, для которых есть ненулевое значение значимости и интенсивности в сочетании с любым типом сервисов и жизненной ситуации, в менее удобном формате. Параметры:
  + service\_type — тип сервиса, название или id
  + living\_situation — жизненная ситуация, название или id

Ключи в ответе:

* + social\_groups — список социальных групп, каждый из элементов содержит как минимум свойства «social\_group», «code» и «id», а также могут содержать «significance» (0.0-1.0) и «intensity» (0-10) - значимость и интенсивность использования;
* /api/relevance/city\_functions — возвращает список городских функций с указанием важности, если задана социальная группа, и с указанием интенсивности использования, если заданы социальная группа и жизненная ситуация. Без параметров возвращает просто список городских функций, для которых есть ненулевое значение значимости и интенсивности в сочетании с любой социальной группой и жизненной ситуации, в менее удобном формате. Параметры:
  + social\_group — социальная группа, название или id
  + living\_situation — жизненная ситуация, название или id

Ключи в ответе:

* + city\_functions — список городских функций, каждый из элементов содержит как минимум свойства «city\_function», «code» и «id», а также могут содержать «significance» (0.0-1.0) и «intensity» (0-10) - значимость и интенсивность использования;
* /api/relevance/service\_types — возвращает список типов сервисов, находящихся в городе, с указанием важности, если задана социальная группа, и с указанием интенсивности использования, если заданы социальная группа и жизненная ситуация. Без параметров возвращает просто список типов сервисов, для которых есть ненулевое значение значимости и интенсивности в сочетании с любой социальной группой и жизненной ситуации, в менее удобном формате. Параметры:
  + city — город, название или id
  + social\_group — социальная группа, название или id
  + living\_situation — жизненная ситуация, название или id

Ключи в ответе:

* + service\_types — список типов сервисов, каждый из элементов содержит как минимум свойства «service\_type», «code» и «count» (количество представителей сервиса в городе), «id», а также могут содержать «significance» (0.0-1.0) и «intensity» (0-10) - значимость и интенсивность использования;
* /api/relevance/living\_situations — возвращает список жизненных ситуаций с указанием важности и интенсивности использования, если заданы социальная группа и тип сервиса. Без параметров возвращает просто список жизненных ситуаций, для которых есть ненулевое зачение значимости и интенсивности в сочетании с любой социальной группой и типом сервиса, в менее удобном формате. Параметры:
  + social\_group — социальная группа, название или id
  + service\_type — тип сервиса, название или id

Ключи в ответе:

* + living\_situations — список жизненных ситуаций, каждый из элементов содержит как минимум свойства «living\_situation», «code» и «id», а также могут содержать «significance» (0.0-1.0) и «intensity» (0-10) - значимость и интенсивность использования;

**Списки социальных групп, городских сервисов, административных единиц, и т. д.:**

* /api/list/social\_groups — список (возрастных) социальных групп, имеющих ненулевые значения значимости (significance) с каким-либо типом сервиса и интенсивности использования (intensity) с какой-либо городской функцией в какой-либо жизненной ситуации. Может принимать на вход конкретные тип сервиса и/или жизненную ситуацию, чтобы возвращать только релевантные к ним социальные группы. Параметры:
  + service\_type — тип сервиса, название или id
  + living\_situation — жизненная ситуация, название или id.

Ключи в ответе:

* + social\_groups — список строк, названия социальных групп
  + social\_groups\_ids — список id социальных групп в том же порядке
* /api/list/city\_functions — список городских функций, имеющих ненулевые значения значимости (significance) с какой-либо социальной группой и интенсивности использования (intensity) с какой-либо социальной группой в какой-либо жизненной ситуации. Может принимать на вход конкретные социальную группу и/или жизненную ситуацию, чтобы возвращать только релевантные к ним социальные функции. Параметры:
  + social\_group — социальная группа, название или id
  + living\_situation — жизненная ситуация, название или id.

Ключи в ответе:

* + city\_functions — список строк, названия городских функций
  + city\_functions\_ids — список id городских функций в том же порядке
* /api/list/service\_types — список типов сервисов, имеющих ненулевые значения значимости (significance) с какой-либо социальной группой и интенсивности использования (intensity) с какой-либо социальной группой в какой-либо жизненной ситуации. Может принимать на вход конкретные социальную группу и/или жизненную ситуацию, чтобы возвращать только релевантные к ним социальные функции. Параметры:
  + social\_group — социальная группа, название или id
  + living\_situation — жизненная ситуация, название или id.

Ключи в ответе:

* + service\_types — список строк, названия типов сервисов
  + service\_types\_ids — список id городских функций в том же порядке
* /api/list/living\_situatons— список жизненных ситуаций, имеющих ненулевые значения интенсивности использования (intensity) с какой-либо социальной группой для какого-либо типа сервиса. Может принимать на вход конкретные социальную группу и/или тип сервиса, чтобы возвращать только релевантные к ним социальные функции. Параметры:
  + social\_group — социальная группа, название или id
  + service\_type — тип сервиса, название или id.

Ключи в ответе:

* + living\_situations — список строк, названия типов сервисов
  + living\_situations\_ids — список id городских функций в том же порядке
* /api/list/infrastructures — иерархический список инфраструктур, городских функций и типов сервисов. Не принимает параметров. Ключи в ответе:
  + infrastructures — список инфраструктур
    - name — имя инфраструктуры
    - functions — список городских функций, принадлежащих инфраструктуре
      * name — имя городской функции
      * service\_types — список названий типов сервисов, принадлежащих городской функции
* /api/list/municipalities — список муниципальных образований. Принимает параметр:
  + city — город, название или id

Ключи в ответе:

* + municipalities — список названий МО
  + municipalities\_ids — список id МО в том же порядке
* /api/list/districts — список административных единиц. Принимает параметр:
  + city — город, название или id
* Ключи в ответе:
  + districts — список названий административных единиц
  + districts\_ids — список id административных единиц
* /api/list/city\_hierarchy — иерархический список административных единиц, муниципальных образований и кварталов. Можно указать административную единицу, МО или квартал для получения МО внутри административной единицы (при указании административной единицы), административную единицу нахождения МО (при указании МО) или МО нахождения кваратала (при указании квартала). Кроме того, можно включить получение самих кварталов в МО. Принимает параметры:
  + city — город, название или id
  + include\_blocks — если указано, то в ответе будет возвращен слой кварталов
  + location — локация для получения образований внутри (название административной единицы, муниципального образования или id административной единицы)

Ключи в ответе:

* + districts — список административных единиц
    - full\_name — название административной единицы. Вероятно, будет переименовано в name
    - id — id административной единицы
    - population — население административной единицы
    - municipalities — список муниципальных образований внутри административной единицы
      * full\_name — название МО. Вероятно, будет переименовано в name
      * id — id МО
      * population — население МО
      * blocks — список кварталов (если был указан параметр include\_blocks)
        + id — id квартала
        + population — население кваратала

**Третья (текущая) версия обеспеченности**. Основана на разделении населения жилых домов на сервисы в зоне доступности и оценке количества свободных мест в сервисах.

* /api/provision\_v3/services — список оцененных сервисов с указанием их оценки. Можно указать тип сервиса и локацию его нахождения (административная единица или муниципальное образование). Принимает параметры:
  + city — город, название или id
  + service\_type — тип сервиса, название или id
  + location — локация нахождения сервиса, название административной единицы или МО

Ключи в ответе:

* + services — список сервисов
    - service\_id — id функционального объекта
    - service\_name — название сервиса
    - address — адрес сервиса
    - block — id квартала нахождения сервиса
    - municipality — название МО нахождения сервиса
    - district — название административной единицы нахождения сервиса
    - center — GeoJSON представление координат центра здания, в котором располагается сервис
    - houses\_in\_access — количество домов, в зону доступности которых попал сервис
    - people\_in\_access — количество жителей в домах, в зону доступности которых попал сервис
    - service\_load — количество потенциальных пользователей сервиса от жилых домов после распределения между другими сервисами того же типа, попавшими в зоны доступности данных домов. Не показательный параметр, участвует в промежуточных вычислениях.
    - needed\_capacity — количество пользователей сервиса (service\_load, умноженный на норматив по предполагаемому количеству пользователей на 1000 человек)
    - reserve\_resource — остаток свободных мест у сервиса (отрицательный запас означает дефицит)
    - provision — оценка сервиса по шкале от 0 до 10 включительно
* /api/provision\_v3/service/{service\_id} — получение данных обеспеченность по одному сервису по его id. Параметров не принимает, ключи аналогичные одному сервису из списка в предыдущем эндпоинте
* /api/provision\_v3/houses — получение оценки обеспеченности всеми типами сервисов для жилых домов. Обязательно должен быть указан хотя бы один из параметров — локация расположения домов (административная единица, МО или квартал) или тип сервиса. Принимает параметры:
  + city — город, название или id
  + location — локация расположения домов (название МО или административной единицы, или id квартала)
  + service\_type — тип сервиса, название или id
  + everything — при указании данного ключа можно выгрузить информацию по домам по всему городу. Передача ключа явным образом вызвана тем, что это достаточно большой объем данных и он может быть запрошен по ошибке.

Ключи в ответе:

* + houses — список жилых домов
    - id — id жилого дома
    - address — адрес жилого дома
    - population — население дома
    - block — id квартала нахождения жилого дома
    - municipality — название МО нахождения жилого дома
    - district — название административной единицы нахождения жилого дома
    - center — GeoJSON представления координат центра жилого дома
    - service\_types — список оценок типов сервисов для дома
      * service\_type — название типа сервиса
      * provision — обеспеченность дома типом сервиса
      * reserve\_resources — общий запас на дом по всем сервисам данного типа, попадающим в его зону доступности (отрицательное значение означает дефицит)
* /api/provision\_v3/house/{house\_id} — получение данных об обеспеченности одного жилого дома по его id. Можно запросить обеспеченность конкретным типом сервиса. Принимает параметры:
  + service\_type — тип сервиса, название или id

Ключи ответа совпадают с описанными в предыдущем эндпоинте для одного жилого дома из списка.

* /api/provision\_v3/house/{house\_id}/normative\_load — Получение данных о потенциальных пользователях типов сервисов (по нормативным величинам). Принимает параметры:
  + service\_type — тип сервиса, название или id (опционально)
  + no\_round — использовать ли округление (по-умолчанию выключено, любое значение, кроме «0», «false» и «no» отключат округление)

Ключи ответа, если был задан service\_type:

* + normative\_load — число (целое, если не установлено no\_round) – количество нормативных пользователей типа сервиса в данном доме

Ключи ответа, если не был задан service\_type:

* + normative\_load — словарь «тип сервиса» - «количество нормативных пользователей в доме».
* /api/provision\_v3/house/{house\_id}/services — получение списка сервисов, оказавшихся в зоне доступности жилого дома. Можно указать тип сервиса для получения только его представителей. Принимает параметры:
  + service\_type — тип сервиса, название или id

Ключи ответа:

* + services — список сервисов, обслуживающих дом
    - id — id функционального объекта
    - service\_type — тип сервиса (если не указан в запросе)
    - center — GeoJSON представление центра сервиса
    - name — название сервиса
    - load\_part — нагрузка, оказываемая домом на конкретный сервис. Другими словами, «количество жителей» заданного дома, пользующихся данным сервисом (в случае одного дома это одинаковое для всех сервисов одного типа число ввиду формул расчета)
    - load\_service — количество пользователей конкретного сервиса из всех домов, в зону доступности которых он попадает
* /api/provision\_v3/service/{service\_id}/houses — получение списка домов для которых заданный сервис оказался в зоне доступности. Принимает параметр:
  + city — город, название или id
* Ключи в ответе:
  + houses — список сервисов, обслуживающих дом
    - id — id дома
    - center — GeoJSON представление центра сервиса
    - load\_part — нагрузка, оказываемая на сервис конкретным домом. Другими словами, «количество жителей» заданного дома, пользующихся данным сервисом.
    - load\_house — количество пользователей конкретного сервиса из всех домов, в зону доступности которых он попадает
    - population — общее население конкретного дома
* /api/provision\_v3/service/{service\_id}/availability\_zone — получение зоны доступности данного сервиса. Не принимает параметров.

Ключи ответа:

* + geometry — geojson-представление геометрии доступности
* /api/provision\_v3/house/{house\_id}/availability\_zone — получение зоны доступности данного для дома и заданного типа сервиса. Имеет обязательный параметр:
  + service\_type (обязательный) — тип сервиса, название или id

Ключи ответа:

* + geometry — geojson-представление геометрии доступности
* /api/provision\_v3/ready — получение списка оцененных типов сервисов с количеством объектов и нормативами оценки. Принимает параметр:
  + city — город, название или id
* Ключи в ответе:
  + service\_types— список оцененных сервисов
    - service\_type — тип сервиса
    - count — количество оцененных объектов
    - radius\_meters — нормативный радиус доступности данного типа сервиса от жилого дома
    - public\_transport\_time — нормативное время доступности сервиса на общественном транспорте от жилого дома
    - max\_load — нормативная максимальная нагрузка (да, она одна на все сервисы одного типа; есть вариант комбинировать данный параметр с capacity сервиса)
    - normative — нормативное количество пользователей сервиса на 1000 человек
* /api/provision\_v3/not\_ready — получение списка сервисов, ожидающих оценки с общим количеством объектов и тех, которые еще не прошли оценку. Принимает параметр:
  + city — город, название или id
* Ключи в ответе:
  + service\_types — список неоцененных сервисов
    - service\_type — тип сервиса
    - total\_count — общее число представителей данного типа сервиса в базе
    - unevaluated — число неоцененных сервисов данного типа сервиса.
* /api/provision\_v3/prosperity/{districts,municipalities,blocks} — получение обеспеченности или благополучия по административным единицам/муниципалитетам/кварталам. Можно указать конкретную локацию (административная единица, МО, квартал), меньшую в большей (все МО в административной единицы, все кварталы в МО/административной единице) или среднее значение по (…) с помощью параметров district, municipality и block. В каждое значение можно указать конкретный экземпляр локации или значение «mean» для получения среднего. Аналогично можно поступить с социальными группами. Если указать параметр provision\_only, то социальная группа обрабатываться не будет, значимость и благополучие возвращаться не будут. Принимает параметры:
  + city — город, название или id
  + district — административная единица, название или id. «mean» для получения среднего значения по административным единицам
  + municipality — муниципальное образование, название или id. «mean» для получения среднего значения по муниципальным образованиям
  + block — id квартала. «mean» для получения средних значений по кваратлам.
  + social\_group — социальная группа. «mean» для получения средних значений по социальной группе.
  + provision\_only — если параметр присутствует и его значение не равно «0», «false», «-» или «no», то все, что касается благополучия, не будет возвращено.

Структуру ответа сложно предоставить ввиду большой вариативности. Ключи ответа:

* + service\_type — тип сервиса
  + city\_function — городская функция
  + infrastructure — тип инфраструктуры
  + services\_count — количество представителей типа сервиса в локации
  + services\_load\_mean — средняя нагрузка (необходимая емкость!) сервисов в локации
  + services\_load\_sum — суммарная нагрузка (необходимая емкость!) сервисов в локации
  + services\_reserve\_mean — средний запас по емкости сервисов в локации
  + services\_reserve\_sum — суммарный запас по емкости сервисов в локации
  + services\_evaluation — средняя оценка сервисов в локации
  + significance — значимость типа сервиса для социальной группы
  + houses\_count — количество жилых домов в локации
  + houses\_reserve\_mean — средний запас мест в сервисах по жилым домам
  + houses\_reserve\_sum — суммарный запас мест в сервисах по жилым домам
  + houses\_provision — средняя обеспеченность сервисом жителей в локации
  + prosperity — благополучие социальной группы по сервису в локации