

Анализ Sabre Geo Search API

Основная информация

API Endpoint: <https://api.cert.platform.sabre.com/v2/geo/search> **Метод:** POST

Версия: v2

Что это такое?

Geo Search V2 API идентифицирует аэропорты, отели и места аренды автомобилей, которые находятся в заданной географической области поиска с использованием радиуса поиска.

Типы поиска

API поддерживает три различных типа разрешения местоположения:

1. Код аэропорта или многоаэропортового города (MAC)
2. Гео код (широта и долгота)
3. Название местоположения (например, название города или населенного пункта)

Возможности поиска

Поиск по Гео коду (широта и долгота)

- Поиск в определенном радиусе

Поиск по RefPoint

Различные типы опорных точек: - Поиск по коду аэропорта: RefPointType = 6, ValueContext = 'CODE' - Поиск по названию города/места: RefPointType = 5, ValueContext = 'NAME' - Поиск по ID полигона: RefPointType = 37, ValueContext = 'CODE' - Поиск по коду отеля: RefPointType = 11, ValueContext = 'CODE' - Поиск по POI: RefPointType = 16, ValueContext = 'NAME'

Поиск по адресу

- Предоставление полной информации об адресе (улица, город, почтовый индекс, страна, штат/провинция, код страны и т.д.)

Фильтры

API предоставляет два типа фильтров GeoAttributes: - **CHAIN** - фильтр по сети (например, EN Filter) - **LOCALAREA** - фильтр по местной области (например, Stuttgart greater area) - Максимум пять цепочек/местных областей или цепочек и местных областей можно задать как фильтры

Параметры

- **Radius parameter** - позволяет радиус поиска до 200 миль или километров
- Дополнительные фильтры CHAIN и LOCALAREA для критериев GeoSearch

Применение

API позволяет клиентам искать отели на основе: - Кода аэропорта/POI/Адреса/ Названия города/Гео кода (широта и долгота) - Предоставляет краткое описание отелей, автомобилей или аэропортов, расположенных в параметрах поиска

Структура запроса GeoSearchRequest

Основные параметры:

GeoRef (обязательный) - содержит параметры для поиска: - **Category** (string) - тип локации для возврата - Возможные значения: HOTEL, CAR, AIR - Пример: "HOTEL"

- **Radius** (number) - максимальное расстояние от опорной точки
- Пример: 1
- **Direction** (string) - направление поиска
- Возможные значения: N, S, E, W, NE, NW, SE, SW
- Пример: "N"
- **UOM** (string) - единица измерения для радиуса

- Возможные значения: MI (мили), KM (километры)
- Пример: "KM"

Варианты поиска:

1. По GeoCode (координатам): - **Latitude** (number, обязательный) - широта -

Пример: 32.877416 - **Longitude** (number, обязательный) - долгота - Пример: -96.959879

2. По AirportCode: - **AirportCode** (string) - трехбуквенный код аэропорта IATA -

Паттерн: [A-Z0-9]{3} - Пример: "DFW"

3. По Address: - **Address** (string) - точный адрес - Длина: 0-50 символов - Пример:

"Hidden Ridge Street Irving City"

4. По POI (Point of Interest): - **POI** (string) - точка интереса - Длина: 0-20 символов -

Пример: "Eiffel Tower"

5. По HotelCode: - **HotelCode** (string) - уникальный номер отеля - Паттерн: [0-9]{1,} -

Пример: "100066952"

6. По RefPoint (опорная точка): - **RefPoint** - содержит параметры для поиска по

опорной точке - **Value** (string, обязательный) - значение опорной точки - Длина:

3-50 символов - Пример: "park" - **ValueContext** (string, обязательный) - тип

информации - Возможные значения: NAME, CODE - Пример: "NAME"

Структура ответа GeoSearchResponse

Основные компоненты ответа:

GeoSearchRS - корневой элемент ответа, содержит:

ApplicationResults - результаты приложения: - **Success** - информация об успешном

выполнении - **TimeStamp** (string) - временная метка ответа - Пример: "2024-10-"

GeoSearchResults - результаты поиска: - **Radius** (number) - радиус поиска - Пример:

1 - **UOM** (string) - единица измерения - Возможные значения: MI, KM - Пример: "KM" -

Category (string) - категория локаций - Возможные значения: HOTEL, CAR, AIR -

Пример: "HOTEL" - **Latitude** (number) - географическая широта центра поиска -

Пример: 32.877416 - **Longitude** (number) - географическая долгота центра поиска -

Пример: -96.959879 - **MaxSearchResults** (number) - максимальное количество

результатов - Пример: 18 - **Offset** (number) - смещение от центра поиска - Пример: 1

GeoSearchResult - массив найденных локаций: - **minItems**: 1 - **maxItems**: 50 -
Каждый элемент содержит: - **Distance** (number) - расстояние от центра поиска -
Direction (string) - направление от центра поиска

Дополнительные параметры RefPoint:

- **RefPointType** (string) - тип опорной точки
- Возможные значения: 6, 7, 11, 16, 18, 37
- 5 - CITY (город)
- 6 - AIRPORT (аэропорт)
- 7 - RAIL_STATION (железнодорожная станция)
- 11 - HOTEL (отель)
- 16 - ATTRACTION (достопримечательность)
- 18 - CAR_RENTAL_LOCATION (место аренды автомобилей)
- Пример: "6"
- **StateProv** (string) - название штата или провинции
- Длина: 2-50 символов
- Пример: "TX"
- **CountryCode** (string) - код страны
- Длина: 2-50 символов
- Пример: "US"
- **PostalCode** (string) - почтовый индекс
- Длина: 2-50 символов
- Пример: "75038"
- **CityName** (string) - название города
- Длина: 2-50 символов
- Пример: "Irvine"

GeoAttributes (фильтры):

- **Attributes** - массив атрибутов для фильтрации (0-10 элементов)
- **Name** (string, обязательный) - название атрибута
 - Пример: "CHAIN"

- **Value** (string, обязательный) - значение атрибута
 - Пример: "НС"

Аутентификация Sabre API

Методы аутентификации:

1. OAuth Token Create /v3 - основной метод для Stateless Sabre REST APIs: - EPR
username 'EPR-PCC-AA' - EPR Password
- clientId - clientSecret

2. OAuth Token Create /v2 - альтернативный метод для тестирования: - EPR
username 'EPR-PCC-AA' - EPR Password

Типы токенов:

Session Tokens: - Используются для поддержания сессии с Sabre APIs - Требуют аутентификации и создания сессии с 15-минутным таймаутом неактивности - Лимит 50 сессий на клиента

Sessionless Tokens: - Используются без поддержания сессии - Действуют семь дней - Не подвержены влиянию неактивности или выходных - Подходят для высоконагруженных и конкурентных запросов

Необходимые учетные данные:

PCC/iPCC (Pseudo City Code): - Буквенно-цифровой идентификатор для туристического агентства - Контролирует функции, которые может выполнять агентство в системе Sabre

EPR (Employee Profile Record): - Индивидуальный профиль сотрудника в туристическом агентстве - Контролирует функции, которые может выполнять сотрудник - Каждый EPR связан с PCC/iPCC

ClientId и ClientSecret (опционально): - Учетные данные для идентификации приложения - Уникальная подпись, генерируемая Sabre для внутренних или внешних клиентских приложений - Для получения обратитесь к менеджеру аккаунта

Получение тестовых учетных данных:

Для REST APIs: - Можно зарегистрироваться для получения Dev Studio аккаунта - Позволяет тестировать подмножество REST APIs в сертификационной среде

Для SOAP APIs: - Необходимо связаться с Sabre для получения учетных данных -
Можно получить доступ к менеджеру аккаунта Sabre для тестирования
подмножества APIs

Важные замечания:

- Некоторые REST APIs могут требовать наличие Sabre-провизионированного аккаунта
- Для тестирования API, требующих PCC на Dev Studio, некоторые поддерживают DEVCENTER
- Тестовые имя пользователя и пароль можно найти в соответствующем разделе документации

Результаты тестирования

Выполненные тесты

1. Тест аутентификации: - URL: <https://api.cert.platform.sabre.com/v3/auth/token> -
Метод: POST - Результат: Ожидаемая ошибка 401 "invalid_client" при
использовании демонстрационных учетных данных - Статус: ✓ Структура запроса
корректна

2. Структура запроса к Geo Search API: - URL: <https://api.cert.platform.sabre.com/v2/geo/search> - Метод: POST - Заголовки: Authorization: Bearer {access_token},
Content-Type: application/json

Примеры тестовых запросов

Поиск по координатам (Москва):

```
{
  "GeoRef": {
    "Radius": 10,
    "UOM": "KM",
    "Category": "HOTEL",
    "GeoCode": {
      "Latitude": 55.7558,
      "Longitude": 37.6176
    }
  }
}
```

Поиск по коду аэропорта:

```
{
  "GeoRef": {
    "Radius": 10,
    "UOM": "KM",
    "Category": "HOTEL",
    "AirportCode": "SV0"
  }
}
```

Поиск по названию города:

```
{
  "GeoRef": {
    "Radius": 10,
    "UOM": "KM",
    "Category": "HOTEL",
    "RefPoint": {
      "Value": "Moscow",
      "ValueContext": "NAME",
      "RefPointType": "5"
    }
  }
}
```

Анализ примера ответа

Структура успешного ответа:

```
{
  "GeoSearchRS": {
    "ApplicationResults": {
      "Success": {
        "TimeStamp": "2024-06-14T15:17:54.123Z"
      }
    },
    "GeoSearchResults": {
      "Radius": 10,
      "UOM": "KM",
      "Category": "HOTEL",
      "Latitude": 55.7558,
      "Longitude": 37.6176,
      "MaxSearchResults": 50,
      "Offset": 0,
      "GeoSearchResult": [...]
    }
  }
}
```

Анализ найденных локаций: - Общее количество результатов: до 50 локаций - Информация о расстоянии и направлении от центра поиска - Детальные адреса и координаты каждой локации - Информация о сетях отелей и рейтингах - Коды отелей для дальнейшего использования в других API

Ключевые особенности API

Преимущества: 1. **Гибкость поиска** - поддержка множества типов поиска (координаты, коды, адреса, POI) 2. **Детальная информация** - полные адреса, координаты, информация о сетях 3. **Фильтрация** - возможность фильтрации по сетям отелей и местным областям 4. **Точность** - расстояние и направление от центра поиска 5. **Масштабируемость** - поддержка до 50 результатов в одном запросе

Ограничения: 1. **Аутентификация** - требуются реальные учетные данные Sabre 2. **Коммерческое использование** - необходим договор с Sabre 3. **Лимиты** - ограничения на количество запросов 4. **Сложность настройки** - требуется получение Client ID и Client Secret

Практические рекомендации

Для разработчиков: 1. Зарегистрироваться на <https://developer.sabre.com> для получения тестовых учетных данных 2. Использовать OAuth Token Create /v2 для начального тестирования 3. Реализовать обработку ошибок аутентификации и API 4. Кэшировать токены доступа (действуют 7 дней для sessionless) 5. Оптимизировать запросы по радиусу поиска

Для бизнеса: 1. API подходит для туристических агентств и платформ бронирования 2. Интеграция позволяет предоставлять актуальную информацию о локациях 3. Возможность создания карт с отелями и достопримечательностями 4. Поддержка мультиязычности и международных стандартов

Сценарии использования

1. Туристические приложения: - Поиск отелей рядом с аэропортом - Поиск достопримечательностей в радиусе от отеля - Планирование маршрутов с учетом расположения объектов

2. Корпоративные решения: - Поиск отелей для командировок - Анализ доступности транспорта - Оптимизация расположения офисов

3. Аналитические системы: - Анализ плотности отелей в регионах - Исследование туристической инфраструктуры - Сравнение предложений по географическим зонам

Заключение

Sabre Geo Search API представляет собой мощный инструмент для поиска и анализа географических данных в туристической индустрии. API обеспечивает:

- **Высокую точность** поиска с поддержкой различных типов запросов
- **Богатую функциональность** с детальной информацией о локациях
- **Гибкость интеграции** с существующими системами
- **Масштабируемость** для обработки больших объемов запросов

Основные выводы: 1. API имеет хорошо структурированную документацию и понятные примеры 2. Требуется коммерческое партнерство с Sabre для продуктивного использования 3. Предоставляет качественные данные для туристических приложений 4. Поддерживает современные стандарты REST API и OAuth аутентификации

Рекомендации для внедрения: 1. Начать с тестовой интеграции через Dev Studio 2. Оценить объемы запросов и выбрать подходящий тарифный план 3. Реализовать надежную систему кэширования и обработки ошибок 4. Рассмотреть интеграцию с другими API Sabre для расширения функциональности

Данный анализ показывает, что Sabre Geo Search API является профессиональным решением для работы с географическими данными в туристической сфере, обеспечивающим высокое качество данных и широкие возможности интеграции.