Инструкция по запуску web сервиса.

Перейти по ссылке на сайт GitHub.com. Нажать на зеленую кнопку со стрелочкой и выбрать «Download ZIP». Начнется загрузка архива. Далее перейти в папку с загруженным архивом и использую архиватор, извлечь в любое удобно место папку из архива.

Если у вас есть visual studio вы можете самостоятельно собрать проект. Если нет, то внутри архива есть еще один с название build.zip. Его тоже нужно извлечь в удобное место.   
  
В папке build (или папке проекта если собираете сами) найти файл под названием «JsonDownloaderService.exe». И запустить его простым двойным нажатием мыши или клавишей Enter.

Отроется черное консольное окно, в котором будет указан адрес по которому будет доступен нужный нам контроллер с его точками доступа.  
  
Скопируйте адрес отсюда <https://localhost:5001> , <http://localhost:5000> или из консоли. Какой-то из адресов будет недоступен, попробуйте другой. По умолчанию откроется пустая страница с несколькими строчками, в которых указаны точки доступа к контроллеру.   
  
Можно выбрать любую нужную точку доступа и использовать ее для выполнения запросов к API. Например, если у вас есть точка доступа «api/SpaceXEvent/events», пользователь может отправить Get-запрос на этот URL получения JSON-массива, который находится в базе данных. Сейчас можно закрыть программу нажав «ctrl + c» или на крестик.

Далее нужно создать базу данных. Будем создавать на пример Postgres. У вас должен установлен и работать локальный сервер и быть доступен консольный клиент psql или pgAdmin панель.   
  
Если работать через панель, то там нужно выбрать любой сервер и базу данных, далее выбрать слева вверху инструмент query tool. В отрывшемся окне можно выполнять запросы. После первого запроса на создание базы, нужно выбрать ее в списке слева и подключится к ней. Все остальные запросы будут выполнять действия внутри ее.  
  
Через консоль (Sql shell) нужно будет ввести значения сервера. По умолчанию (сервер - localhost, для база данных - postgres, порт - 5432, в качестве пользователя – postresб пароль, который вы задавали во время установки, часто – root. При вводе пароля символы не отображаются).

В любом тестовом редакторе можно открыть файл под названием «ScriptCreateDbAndTable.sql» в папке проекта по пути(«\JsonDownloaderService\JsonDownloaderService»). Или Использовать команды из этого документа.  
  
Теперь нужно ввести команды по очереди. Ниже те же команды, что и в файле «ScriptCreateDbAndTable.sql».  
  
***CREATE database spacexeventsdb;***   
  
«spacexeventsdb» - вместо этого имени может быть любое другое. Эта команда создает базу данных. В ответ должна быть надпись successfully.  
  
Далее подключимся к этой базе данных для осуществления с ней взаимодействия. Для этого применяется команда \c (сокращение от connect), после которой указывается имя базы данных. Внимательно смотрим на сторону слеша.

***\c spacexeventsdb***

Затем создадим в этой базе данных таблицы с помощью следующих команд по очереди:  
  
 ***CREATE TABLE links ( id bigint GENERATED BY DEFAULT AS IDENTITY, article text NOT NULL, CONSTRAINT "PK\_links" PRIMARY KEY (id) );***  
  
 ***CREATE TABLE space\_x\_event ( id bigint GENERATED BY DEFAULT AS IDENTITY, title text NOT NULL, event\_date\_utc timestamp with time zone NOT NULL, event\_date\_unix bigint NOT NULL, details text NOT NULL, "LinksId" bigint NOT NULL, CONSTRAINT "PK\_space\_x\_event" PRIMARY KEY (id), CONSTRAINT "FK\_space\_x\_event\_links\_LinksId" FOREIGN KEY ("LinksId") REFERENCES links (id) ON DELETE CASCADE );***  
  
Вставим тестовые данные:  
  
***INSERT INTO links (article) VALUES ('https://example.com');  
  
INSERT INTO space\_x\_event (title, event\_date\_utc, event\_date\_unix, details, "LinksId") VALUES ('TestTitle', '2023-07-14T19:19:02.031727Z', 1231231231, 'DetailTest', (SELECT id FROM links WHERE article = 'https://example.com'));***  
  
Настройка базы готова. Остался еще 1 нюанс. В тай же папке где лежит файл запуска программы «JsonDownloaderService.exe», нужно найти файл «appsettings.json». Его можно отрыть любым текстовым редактором.   
В нем есть строчка «"Default": "Server=localhost;Port=5432;Database=test10;User Id=postgres;Password=root;"». Это настройки базы данных, порт, логин и пароль указываем такой, как вы указали при настройках сервера postgres. Так же обязательно нужно поменять значение «***Database= spacexeventsdb;».*** Если вы указа при создании другую базу данных (в самом начале, первой командой), то необходимо после символа равенства (=) указать это название. Файл сохраняем.   
  
Вот и все. Переходим к Api сервису.   
  
Вновь запустим «JsonDownloaderService.exe». Для теста выполним Get запрос на по адресу <https://localhost:5001/api/SpaceXEvent/events> . В ответ должна быть json строка, в которой указаны разные данные в том числе что-то вроде «"title": "TestTitle",». Это означает, что все работает.  
  
Теперь в рамках задания в контроллере есть Post метод, который подключается к Api, получает строку, обрабатывает ее и сохраняет в базу данных. Для того что бы воспользоваться этим end point, нужно ввести открыть Postman или curl.  
Воспользуемся Postman. Создаем новый **Post** запрос вводи адрес [***https://localhost:5001/api/SpaceXEvent//download***](https://localhost:5001/api/SpaceXEvent//download) ***.***  
Нажимаем Send. В ответ придёт ответ с заголовком похожим на «"title": "Falcon reaches Earth orbit"». Теперь создаем новый **Get** запрос по адресу <https://localhost:5001/api/SpaceXEvent/events> .

Если вы выполнили все команды из этого описания, то будет выведено 3 объекта, которые хранятся в базе данных. Если существует большой интерес, можно создать еще один **Post** запрос на адрес <https://localhost:5001/api/SpaceXEvent/create_test_data> . Это запрос добавляет тестовые данные в базу данных затем если выполнить Get запрос на <https://localhost:5001/api/SpaceXEvent/events>, в базе будет еще 4 объекта. И так далее. Проверить, что объекты находятся в базе данных а не в памяти, можно перезапустив программу и выполнив снова **Get** <https://localhost:5001/api/SpaceXEvent/event> . Количество объектов не изменится. Так же можно Drop таблицу БД. Так же в контроллере есть другие методы, обновления и удаления.  
  
 Возможные ошибки.

Сервер недоступен:

Порт 5000 работает на HTTP, а 5001 на HTTP**S.** Нужно внимательно проверять адрес отправки.  
Если не работает один, то попробовать на другой.Статус ответа 400, 500 или любой другой:   
Большая вероятность того, что не верно указан адрес, а также что не верно выбран метод отправки запроса. Перепутан post и Get.  
  
  
Еще раз список всех команд, чтобы не бегать по документу.   
  
***CREATE database spacexeventsdb;***

***\c spacexeventsdb***

***CREATE TABLE links ( id bigint GENERATED BY DEFAULT AS IDENTITY, article text NOT NULL, CONSTRAINT "PK\_links" PRIMARY KEY (id) );***  
  
 ***CREATE TABLE space\_x\_event ( id bigint GENERATED BY DEFAULT AS IDENTITY, title text NOT NULL, event\_date\_utc timestamp with time zone NOT NULL, event\_date\_unix bigint NOT NULL, details text NOT NULL, "LinksId" bigint NOT NULL, CONSTRAINT "PK\_space\_x\_event" PRIMARY KEY (id), CONSTRAINT "FK\_space\_x\_event\_links\_LinksId" FOREIGN KEY ("LinksId") REFERENCES links (id) ON DELETE CASCADE );***

***INSERT INTO links (article) VALUES ('https://example.com');  
  
INSERT INTO space\_x\_event (title, event\_date\_utc, event\_date\_unix, details, "LinksId") VALUES ('TestTitle', '2023-07-14T19:19:02.031727Z', 1231231231, 'DetailTest', (SELECT id FROM links WHERE article = 'https://example.com'));***  
  
  
Не забываем менять базу данных в «build.\appsettings.json».

«"Default": "Server=localhost;Port=5432;**Database= *spacexeventsdb***;User Id=postgres;Password=root;"».