КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образование учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

**по практической работе №11**

**«Обмен данными с веб-сервером в формате JSON»**

**по МДК01.03.**

Работу выполнил студент 393 гр.:

Кашицын Артем Андреевич

Преподаватель:

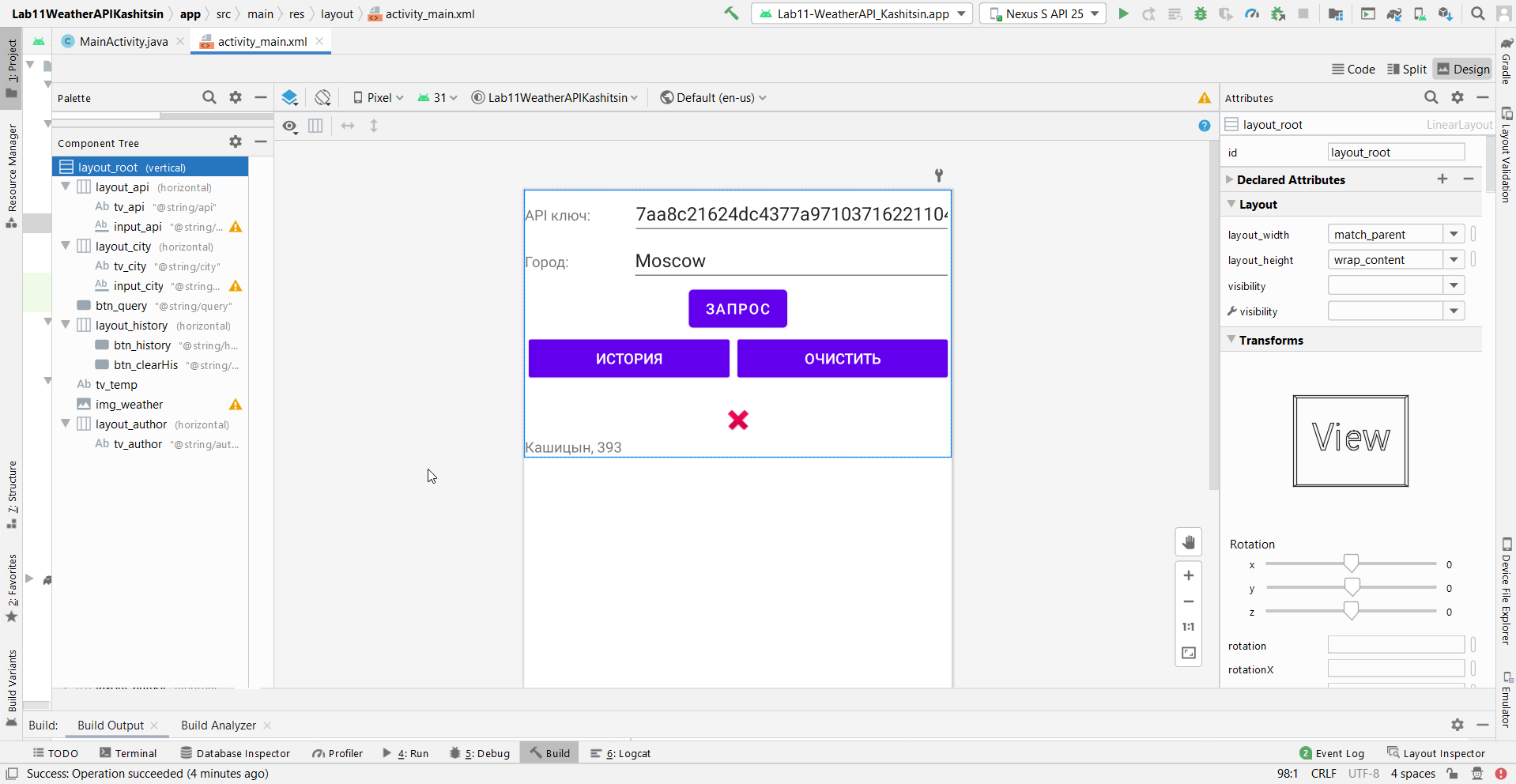
Фомин Александр Валерьевич

Санкт-Петербург 2021

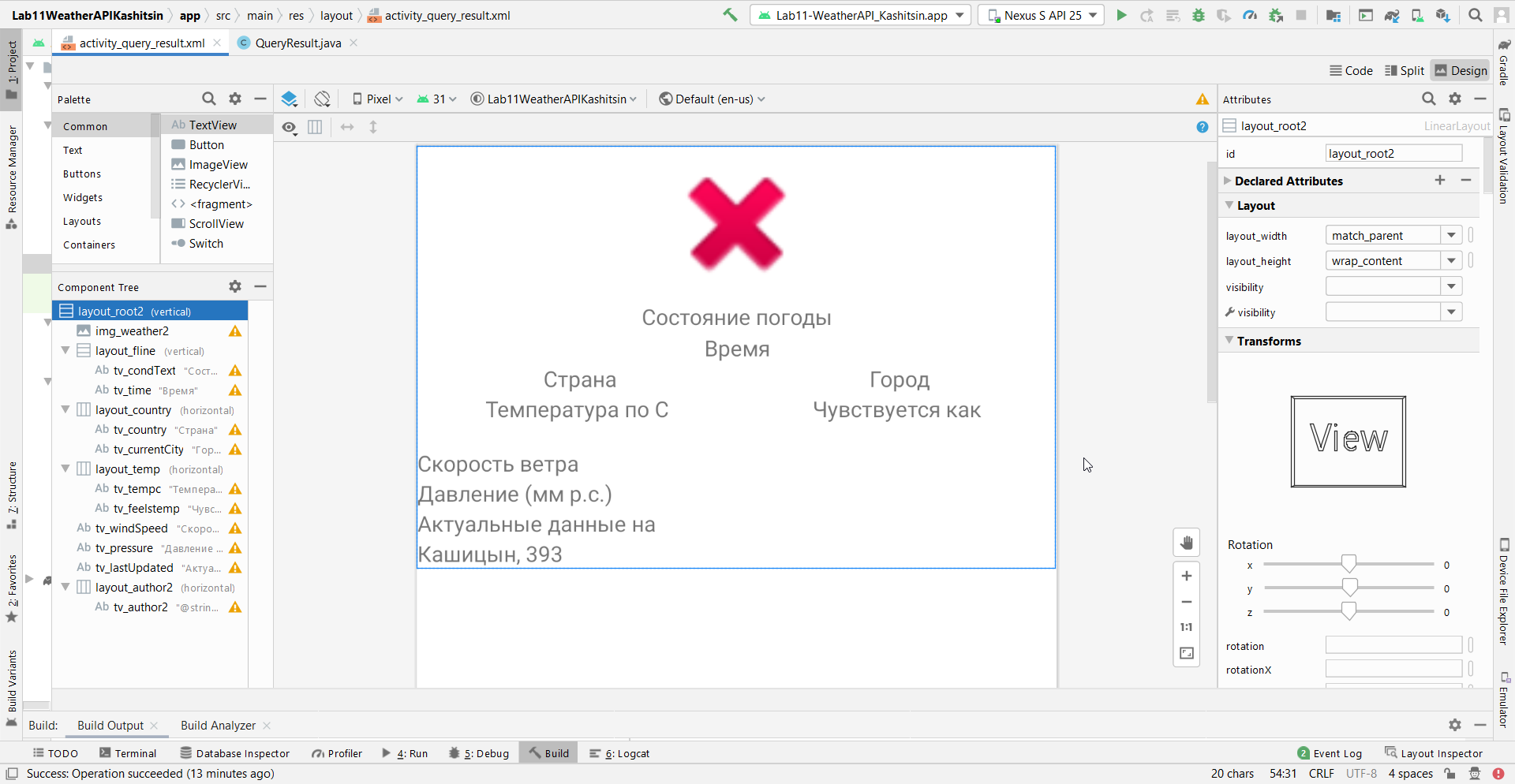
**ХОД РАБОТЫ**

Приложение состоит из двух Activity: MainActivity и QueryActivity.

На форме MainActivity расположены 4 TextView с надписями: «API ключ», «Город», а также TextView для записи туда значения температуры и TextView с отображением авторства. 2 EditText поля для ввода API ключа и ввода города для определения температуры. 3 кнопки «Запрос» - получение данных с сервера и переход на вторую форму с подробными показателями погоды; «История» - вызывает диалоговое окно со списком, представляющий историю запросов, где отображается дата и время запроса, текст запроса, температура по Цельсию, температура по ощущениям, состояние погоды; «Очистить» - удаление истории. ImageView для отображения изображения состояния погоды.



На форме QueryActivity расположен ImageView для отображения изображения состояния погоды. 10 TextView для отображения: состояния погоды в текстовом виде, текущего времени в выбранном городе, выбранной страны, выбранного города, температуры по цельсию, температуры то как она чувствуется, скорости ветра, давления, времени обновления данных на сервере, авторства



В код приложения входят 3 класса: History и 2 класса для работы с базой данных приложения: DBHistory, DBImages.

Класс History состоит из 5 полей:

Date – дата и время запроса

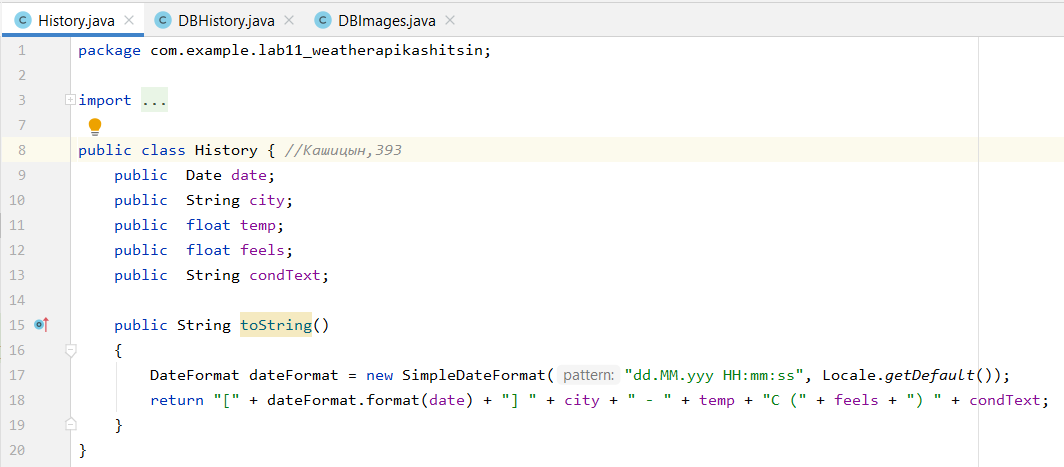
City – город в запросе

Temp – температура в городе

Feels – температура в городе по ощущениям

condText – состояние погоды

Функция toString возвращает текстовое значение экземпляра класса в виде: [дата и время запроса] Город – Температура (Как чувствуется) Состояние погоды



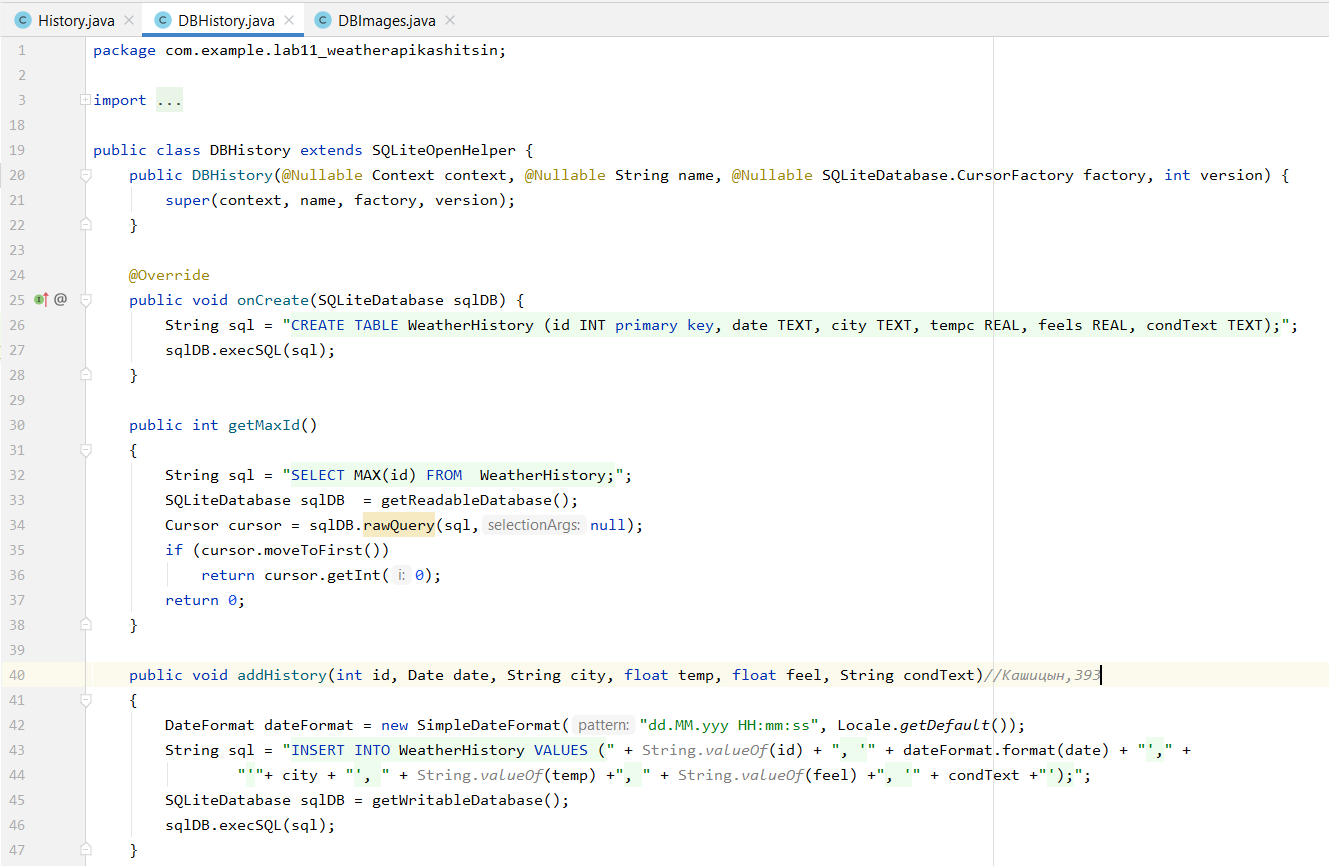
Класс DBHistory при первом запуске приложения создает таблицу в базе данных со столбцами, соответственными полям класса History.

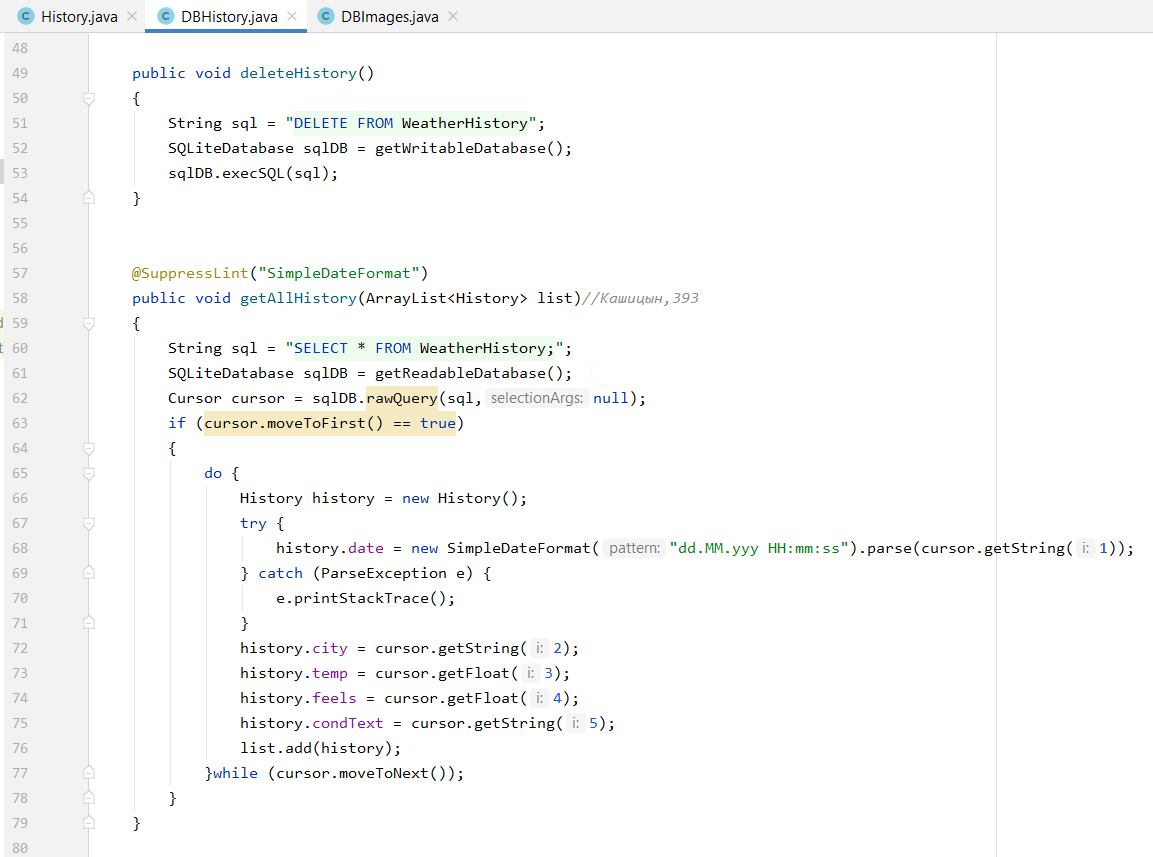
Функция getMaxId возвращает Id последней записи

Процедура addHistory добавляет запись по всем столбцам в таблице. В начале полученная дата запроса приводится к необходимому виду для хранения в базе.

Процедура deleteHistory удаляет все записи в таблице

Процедура getAllHistory, получая ArrayList, собирает все записи в таблице и создает экземпляр класса History, заполняются все поля класса с таблицы и добавляет экземпляр в ArrayList.

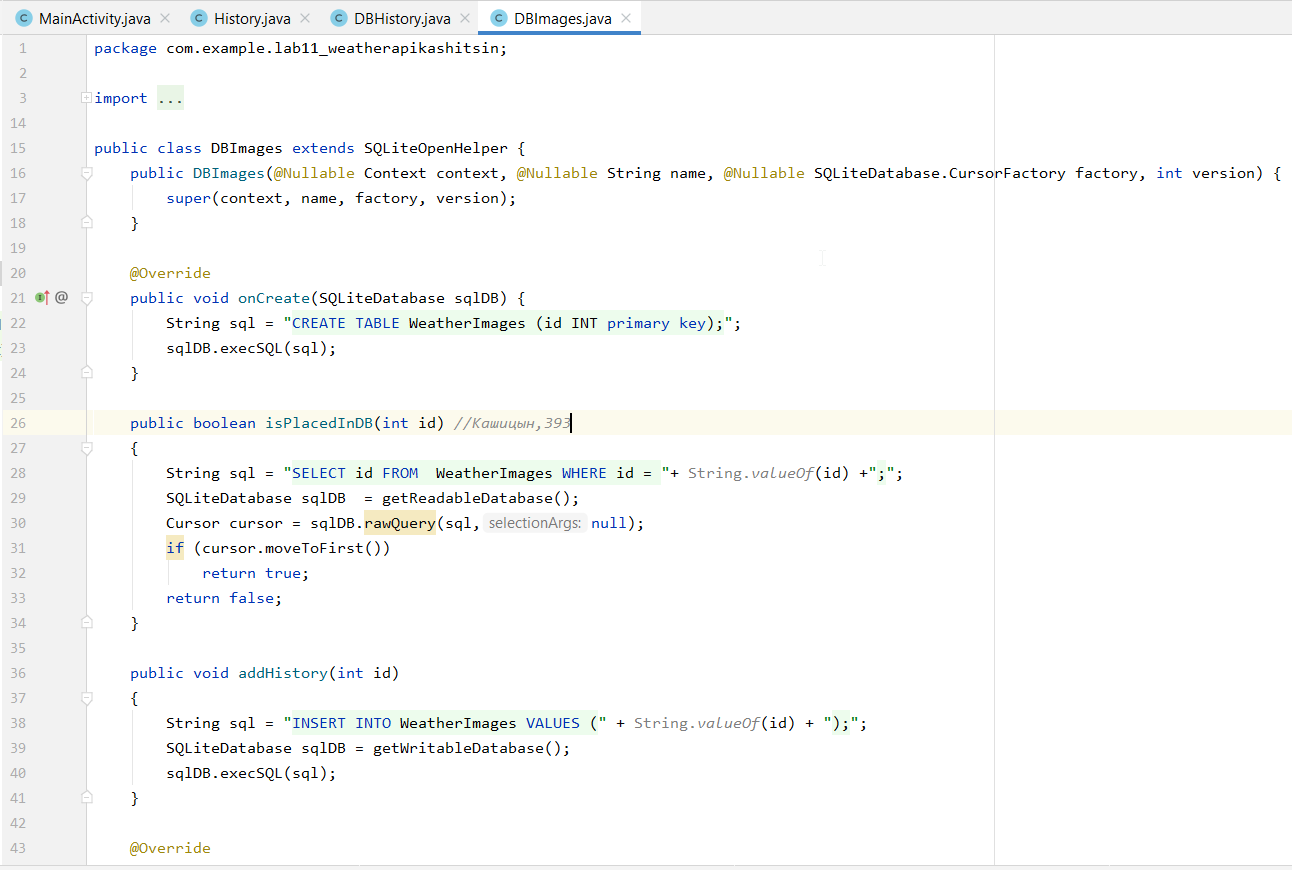




Класс DBImages при первом запуске приложения создает таблицу в базе данных со столбцом Id – полученный параметр current –> condition –> code в ответе от сервера. Code – код состояния погоды, по нему или по тексту состояния можно идентифицировать изображение, отправляемое сервером.

Функция isPlacedInDB возвращает true, если Condition Code уже есть в таблице, иначе – false

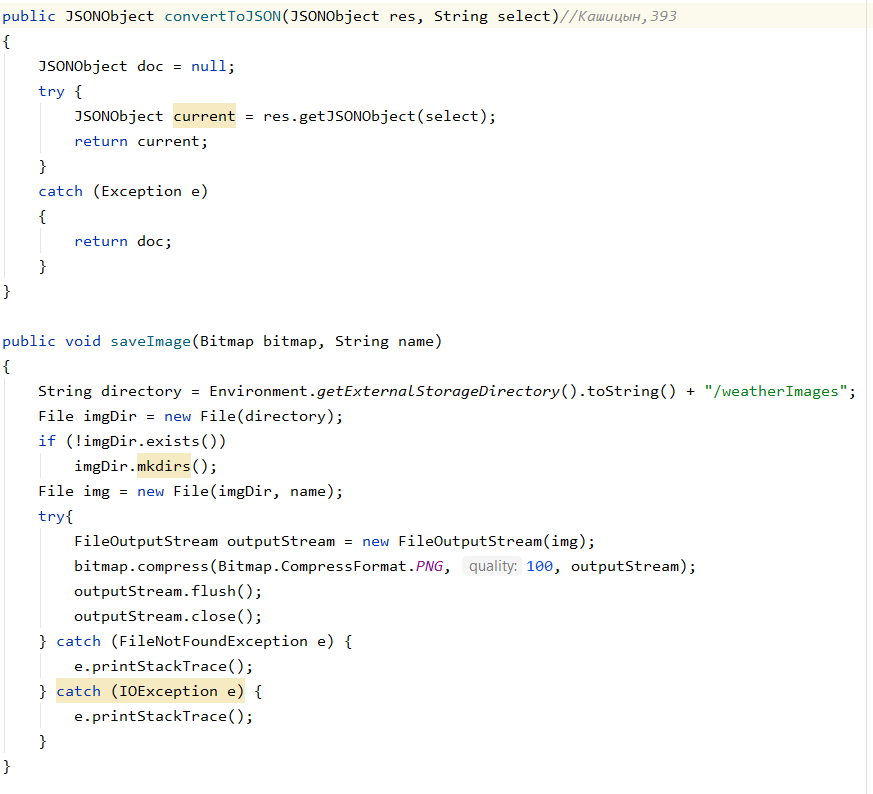
Процедура addHistory добавляет новый Condition Code



Код MainActivity.java

Функция convertToJSON получает JSON объект и возвращает новый JSON объект по указанному параметру. Необходима для отделения выбранной ветви с параметрами от основного JSON ответа сервера.

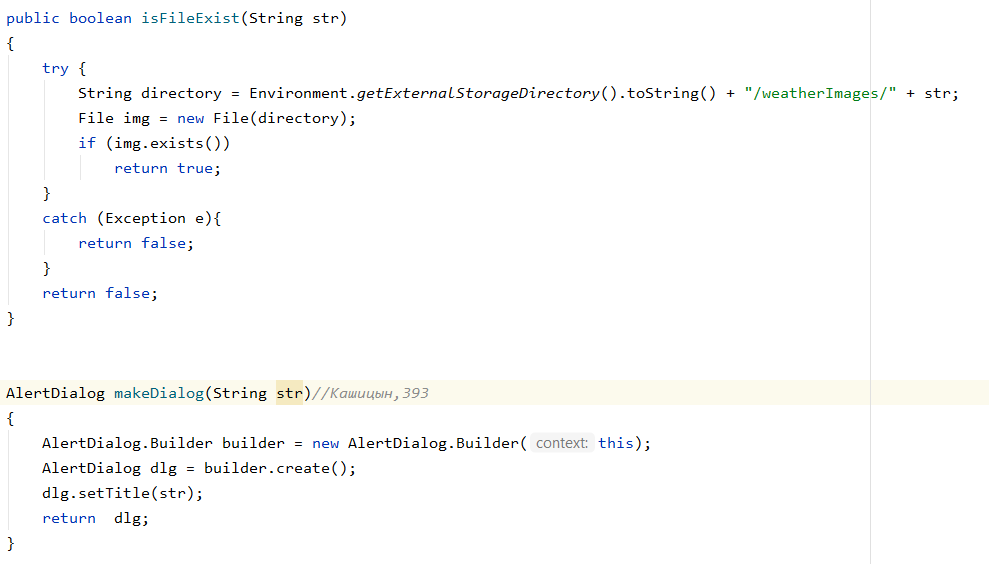
Процедура saveImage: создание (если ещё не создано) папки в памяти телефона, откуда запускается приложение. Туда будет положен файл с названием, полученным от сервера (Condition -> Code). Полученный bitmap будет сохранён в PNG формате с качеством 100 в указанный путь.

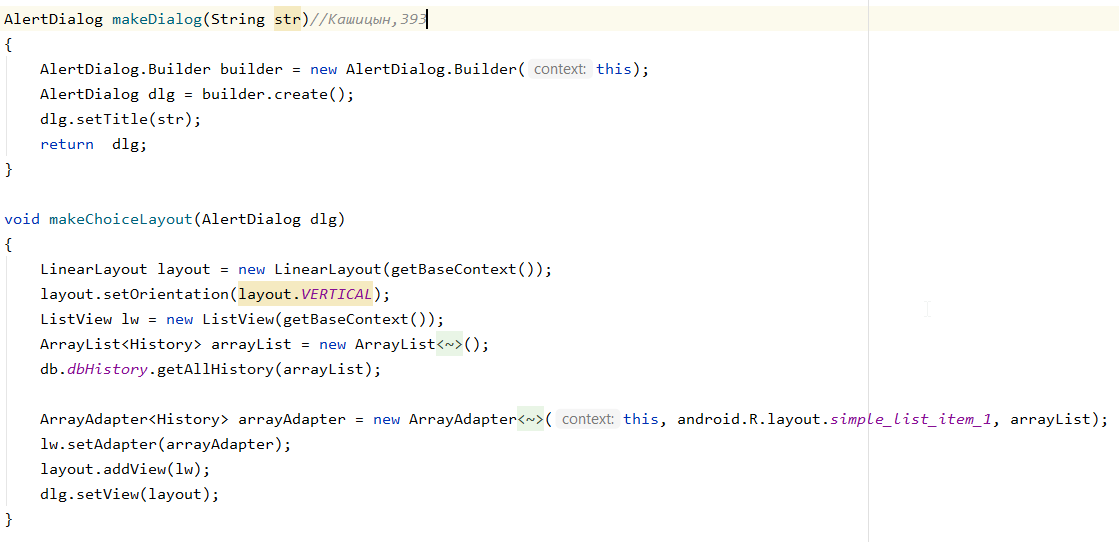


Функция isFileExist – по полученному имени файла, возвращает true, если файл найден, иначе – false

Функция makeDialog создает AlertDialog с указанным заголовком

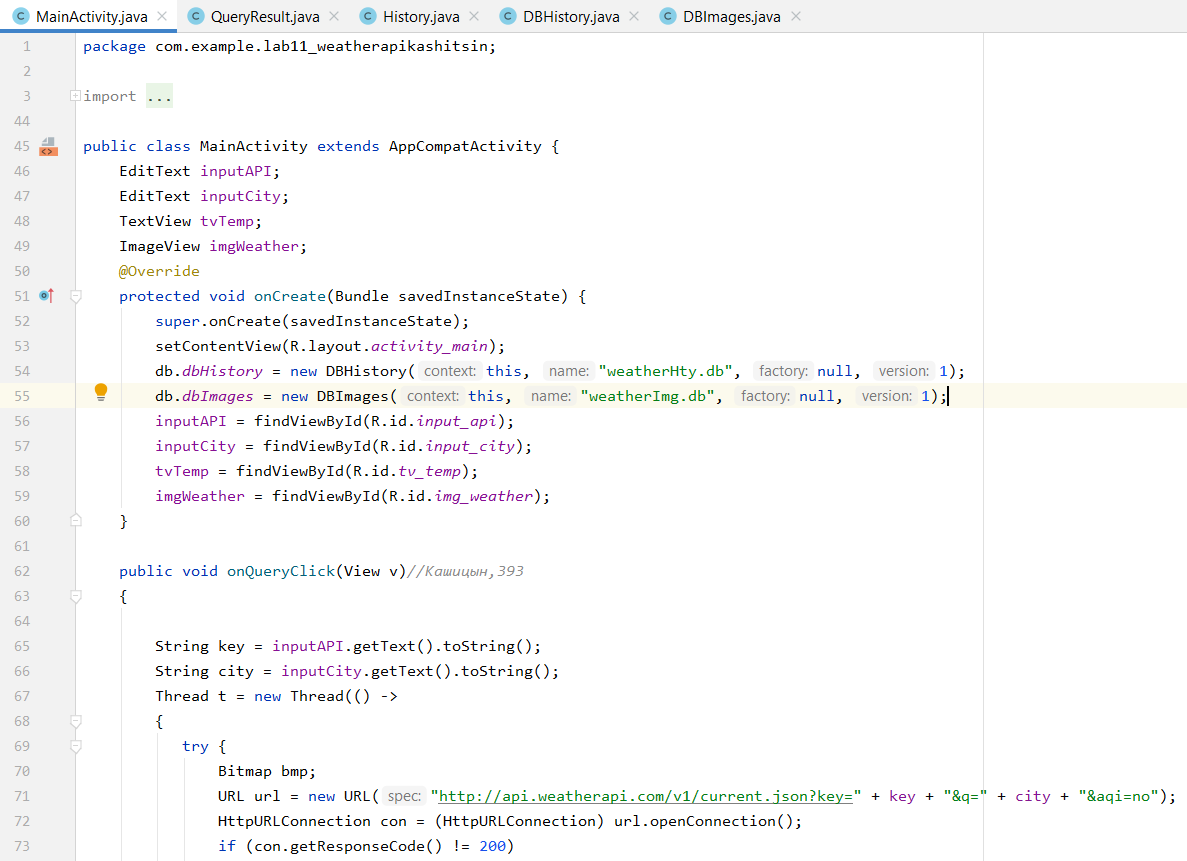
Функция makeChoiceLayout создает layout с ListView, в который через ArrayAdapter в виде списка будет помещен ArrayList, содержащий в себе историю запросов. Созданный layout устанавливается в полученный AlertDialog.





Функция по клику «Запрос»

Получаем API ключ и запрашиваемый город. Состовляем необходимый GET запрос. Открываем соединение. Если сервер не вернул ответ 200 (ОК), то вызываем ошибку, если код 400 – проблема с составлением запроса (Город введен некорректно). Если сервер вернул код 200, получаем массив данных ответа и прекращаем соединение. Массив в виде строки разбираем на три параметра JSON: location (данные связанные с выбранным городом), current (погодные данные), condition (входит в current, представляет собой данные о состоянии погоды: текст состояния, адрес картинки состояния, код состояния). Если код состояния есть в базе, и файл с таким названием существует, то в bitmap загружаем существующий файл. Иначе – открываем соответствующий запрос на картинку с состоянием погоды и записываем картинку в bitmap, сохраняем картинку в файл с названием в виде кода состояния погоды и добавляем код состояния в базу. Загружаем картинку в ImageView. На вторую Activity будем передавать данные полученные из JSON ответа: город, текст и код состояния погоды, страна, где находится город, время в выбранном городе, время последнего обновления данных на сервере, температура по цельсию, скорость ветра, давление, температура, как она ощущается

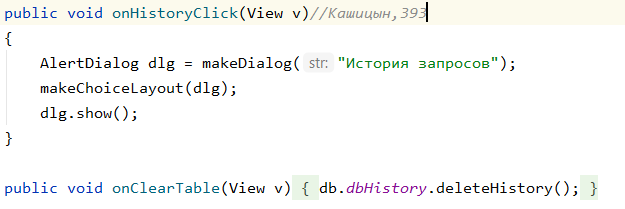






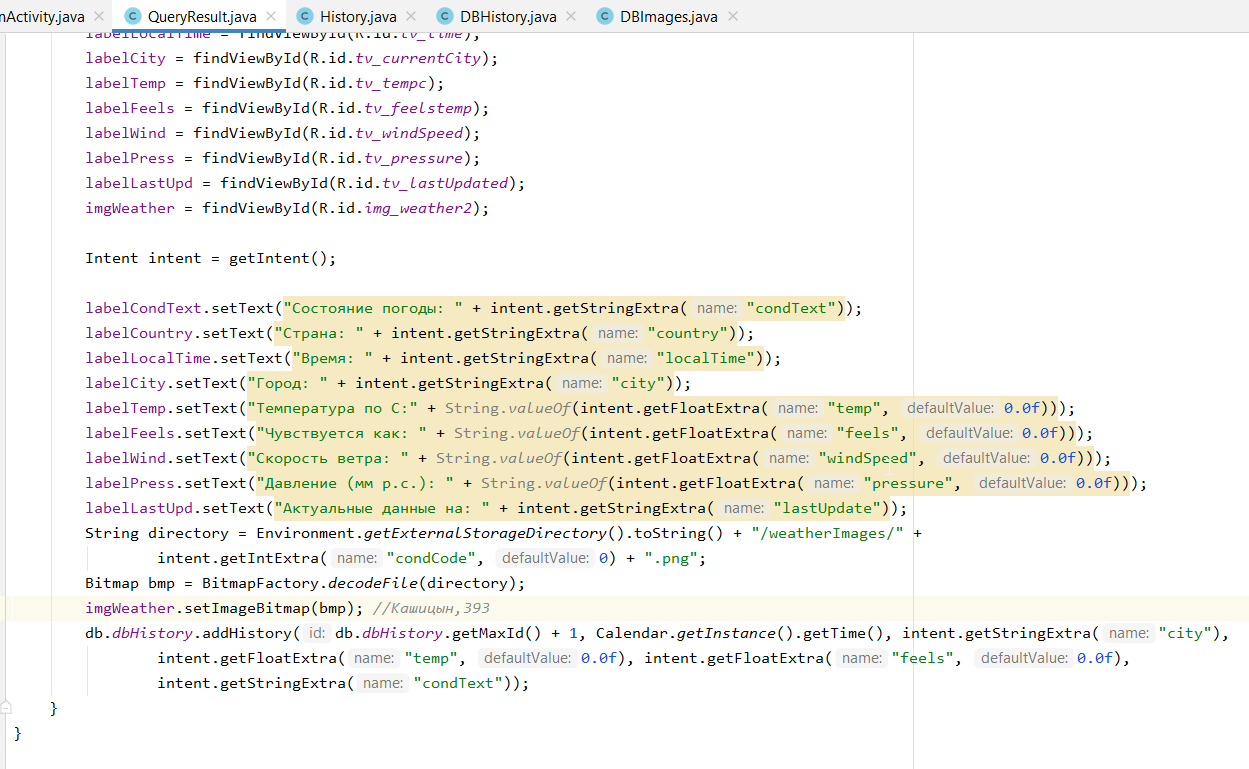
По кнопке «История» открываем диалоговое окно с историей

По кнопке «Очистить» удаляем историю с базы



Код QueryActivity получаем и записываем полученные текстовые данные, достаем файл по conditionCode и загружаем картинку. Добавляем в базу новый составленный экземпляр History с необходимыми данными.



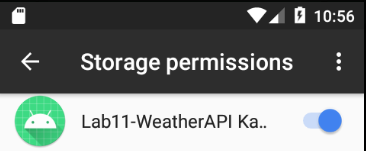


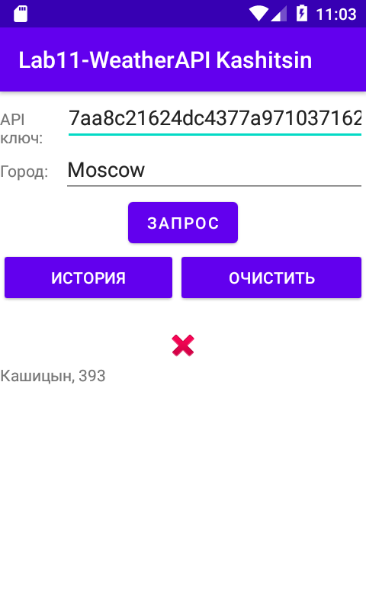
Перед запуском необходимо предусмотреть необходимые разрешения: доступ в интернет, работа с файлами во внутреннем хранилище

AndroidManifest:

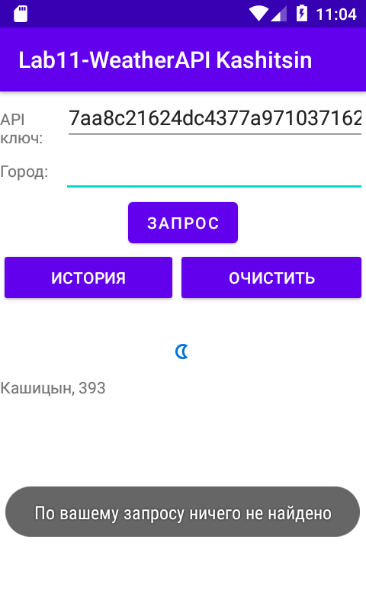
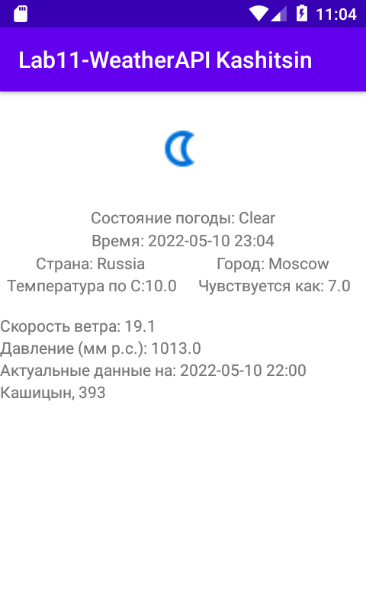


Необходимо отдельно на самом устройстве разрешить приложению работу с файлами. Находим в настройках App Permission -> Storage. В списке находим приложение и разрешаем доступ.



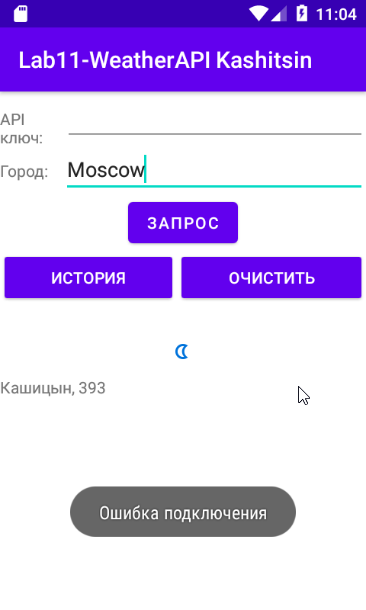
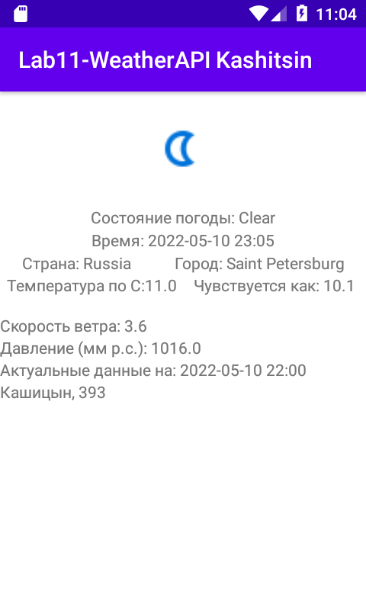
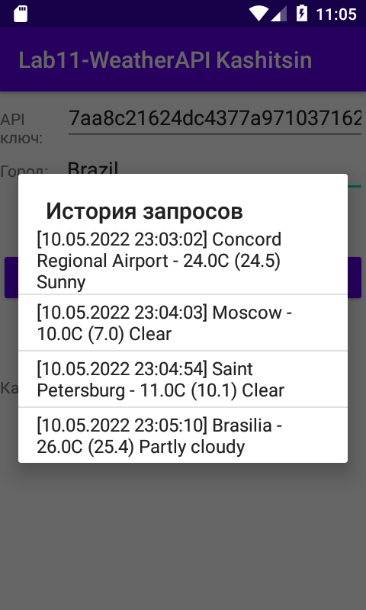
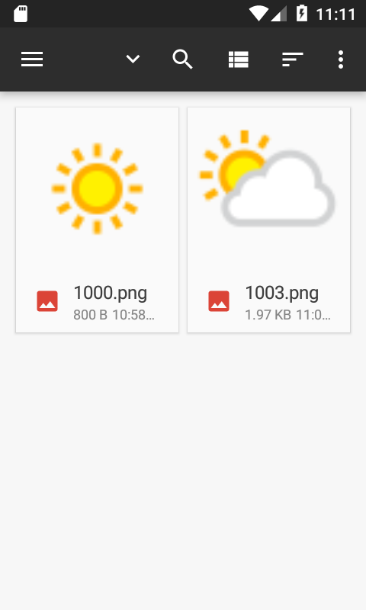
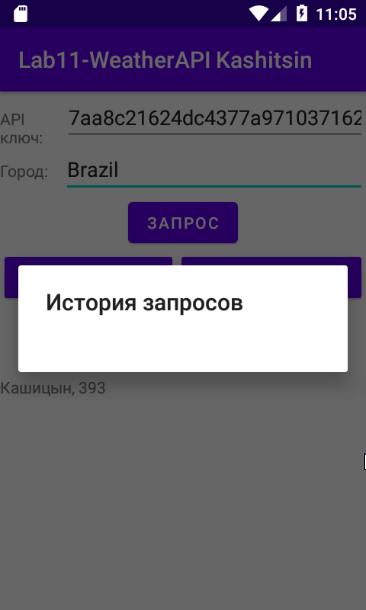
**ИТОГОВОЕ ПРИЛОЖЕНИЕ**

Первый запуск



Запрос - Москва

Отсутствие запроса



Очистка истории

Сохраненные изображения

История запросов

Отсутствие ключа

Запрос – Бразилия

Запрос – Санкт-Петербург

Отсутствие ключа