

Развертывание GraphHopper в качестве веб-сервиса для построения маршрутов

[Обсудить в форуме](#) Комментариев — 7

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/graphhopper-deployment.html>

Рассмотрен процесс развертывания GraphHopper в качестве Веб-сервиса.

Содержание

- [1 Введение](#)
- [2 Запуск GraphHopper с помощью Jetty Runner](#)
- [3 Запуск GraphHopper с помощью Jetty](#)
- [4 Ссылки](#)

Введение

[GraphHopper](#) - это быстрый роутинговый движок, написанный на Java. К его отличительным особенностям можно отнести:

- работу с данными OpenStreetMap
- эффективное использования памяти
- широкие возможности по настройке
- свободная лицензия Apache

К роутинговым движкам относится программное обеспечение, позволяющее прокладывать маршрут на карте из одной точки в другую по имеющемуся графу. Существует большое количество программ данного рода, одна из наиболее известных - [Open Source Routing Machine \(OSRM\)](#), написанная на C++.

Так как код GraphHopper полностью [открыт](#), то вы можете использовать необходимые Java-классы в своём собственном приложении. Кроме того, GraphHopper может быть использован в качестве Веб-сервиса маршрутизации, предоставляющего соответствующий HTTP API. Именно процедуре развертывания GraphHopper в качестве сервиса маршрутизации и посвящена в данная статья. Пример Веб-приложения, использующего GraphHopper в качестве роутингового движка можно посмотреть [тут](#). Отметим, что данное приложение вы получаете автоматически при разворачивании GraphHopper в качестве Веб-сервиса.

Для того, чтобы использовать GraphHopper в качестве Веб-сервиса можно самостоятельно собрать собственный [WAR-файл](#) либо воспользоваться уже готовой сборкой. Мы будем использовать как раз готовую сборку. К первому варианту следует прибегать в тех случаях, если готовая сборка вас чем-то не устраивает и вы хотите что-то в ней изменить. Подробнее про сборку GraphHopper [тут](#).

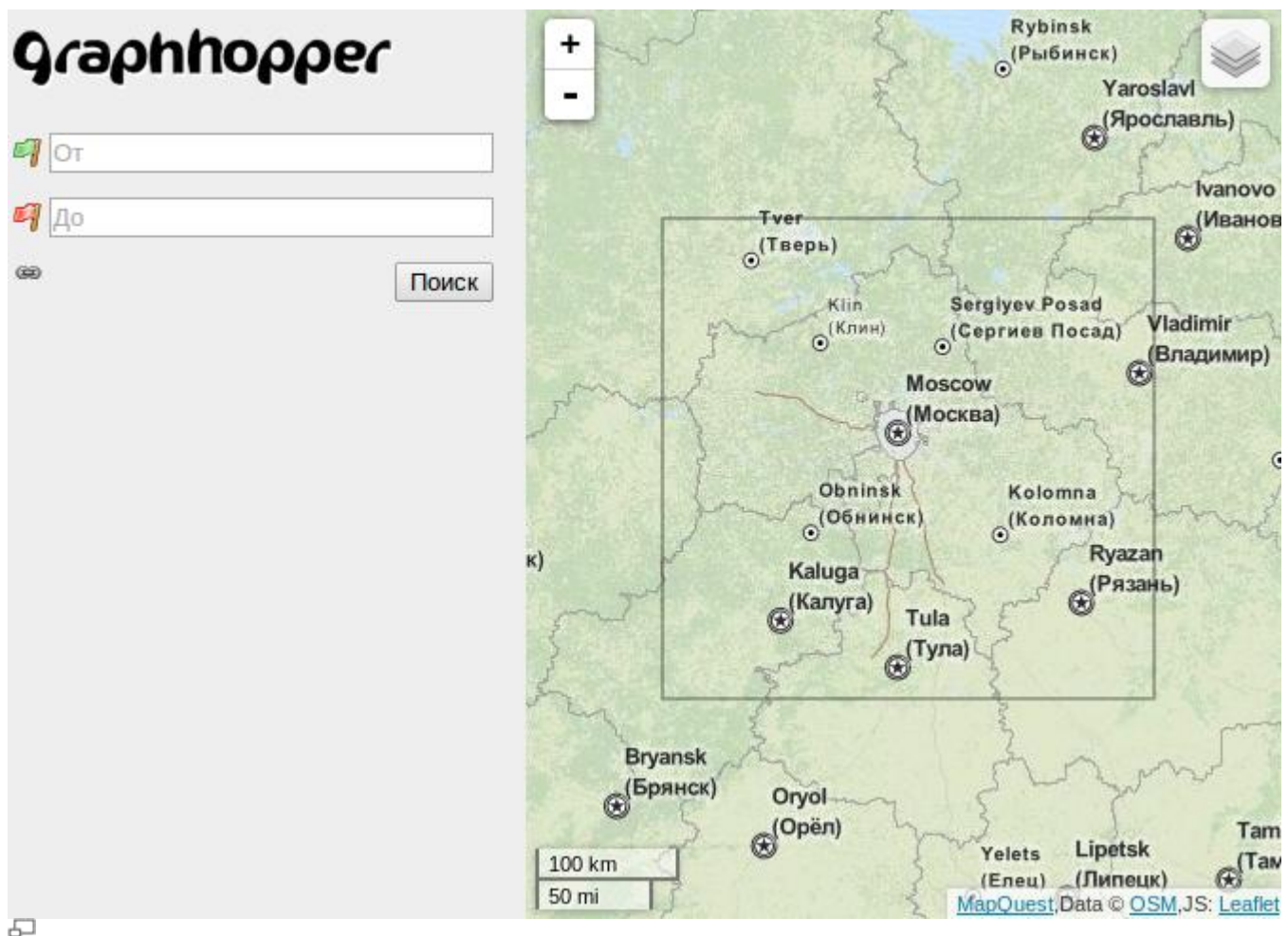
Для развертывания WAR-файла нам потребуется воспользоваться каким-либо контейнером сервлетов. Мы рассмотрим два варианта: развертывание GraphHopper: с помощью [Jetty Runner](#) и [Jetty](#). Первый вариант следует использовать, если вы хотите "по-быстрому" оценить, что из себя представляет GraphHopper. Второй же вариант подходит для промышленного использования.

Запуск GraphHopper с помощью Jetty Runner

Jetty Runner предназначен для быстрого запуска Веб-приложений непосредственно из командной строки и не требует установки полноценного Веб-сервера.

1. Для дальнейшей работы создадим в домашней директории каталог *graphhopper* (все файлы, с которыми мы будем работать, должны быть размещены здесь). В Linux:
`mkdir ~/graphhopper`
2. Воспользуемся версией Jetty Runner 9.1.2 [jetty-runner-9.1.2.v20140210.jar](#). Поместим его в директорию *graphhopper*. Отметим, что для запуска Jetty Runner данной версии в системе должна быть установлена JRE7 или JRE8. Если же у вас нет возможности обновить JRE, то для запуска Jetty Runner на JRE6 можно попробовать воспользоваться [старыми версиями](#) Jetty Runner.
3. На сегодняшний день последняя стабильная версия GraphHopper - 0.2, master-версия имеет ряд проблем, например, неправильное указание направлений в описании маршрута [#171](#) (уже вроде исправили) и поэтому мы будем работать именно с версией 0.2. Скачиваем [graphhopper-web-0.2.war](#).
4. Скачиваем пример конфигурационного файла GraphHopper [config-example.properties](#).
5. Скачиваем данные проекта OpenStreetMap на основе которых будет осуществляться роутинг. Для примера возьмём [Москву](#).
6. Запускаем GraphHopper:
`java -Dgraphhopper.config=config-example.properties -Dgraphhopper.osmreader.osm=RU-MOS-140322.osm.pbf -jar jetty-runner-9.1.2.v20140210.jar --port 8989 graphhopper-web-0.2.war`

Первый запуск займёт какое-то время (порядка 1 минуты), это связано с построением графа роутинга. Если в процессе запуска не возникло никаких ошибок, то открыв в браузере страницу по адресу <http://localhost:8989/>, вы увидите следующую картину:



Веб-приложение GraphHopper

По данному адресу (<http://localhost:8989/>) расположено Веб-приложение, с помощью которого можно указать начальную и конечную точки и просмотреть получившийся маршрут, а также его текстовое описание с указанием того куда и через сколько метров следует поворачивать. В некоторых случаях достаточно просто вставить ссылку на это приложение в свою страницу и тем самым предоставить клиентам возможность построения маршрутов.

Если же вы не хотите использовать Web-приложение GraphHopper, а только HTTP API, то вам следует ознакомиться со следующей [документацией](#). Пример HTTP запроса к GraphHopper:

<http://localhost:8989/api/route?point=55.422779,37.254639&point=55.646599,38.023682&locale=ru-RU>.

Также имеется API для получения информации об инстансе GraphHopper: <http://localhost:8989/api/info>.

Запуск GraphHopper с помощью Jetty

Jetty — это свободный контейнер сервлетов, написанный полностью на Java. Может использоваться как HTTP-сервер или в паре со специализированным HTTP-сервером. Если у вас еще не установлен Jetty, то установите его. Дальнейшие инструкции предполагают, что установлен Jetty версии 9, который слушает порт 8180.

1. В директорию *webapps* контейнера Jetty поместите WAR-файл GraphHopper, [graphhopper-web-0.2.war](#), переименовав его в *graphhopper.war*.
2. Сюда же поместим файл с данными OpenStreetMap, [Москва](#).
3. В корень директории, куда установлен Jetty (уровень вложенности на котором находится директория *webapps*), поместите файл с настройками GraphHopper [config-example.properties](#), переименовав его в *config.properties*.
4. Отредактируйте файл *config.properties*, добавив в него полный путь до файла с данными OpenStreetMap (замените путь на свой):

```
# where to find osm/pbf file
osmreader.osm=/usr/local/jetty/webapps/RU-MOS-140322.osm.pbf
```

5. Перезапустите Jetty. В Linux:

```
sudo service jetty restart
```

После того как вы перезапустили Jetty, в директории *webapps* будет создана новая директория, содержащая роутинговый граф, в нашем случае это будет директория *RU-MOS-140322.osm-gh*. Если вы сразу же после перезапуска Jetty попытаетесь перейти по адресу <http://localhost:8180/graphhopper/> (либо Jetty вообще не запустился), то ничего не увидите. Нужно подождать некоторое время, пока граф построится (директория *RU-MOS-140322.osm-gh* будет непустой), либо если Jetty не запустился, то тоже подождать и запустить его еще раз.

6. Откройте страницу <http://localhost:8180/graphhopper/>. Вы должны увидеть карту и следующее сообщение:

```
GraphHopper API offline? http://localhost:8180
```

Причина возникновения этого сообщения заключается в том, что Веб-приложение ожидает, что HTTP API доступно по адресу <http://localhost:8180>, а не по <http://localhost:8180/graphhopper/>. Данную проблему можно решить двумя путями: либо взять исходники GraphHopper, отредактировать там путь и собрать собственный WAR-файл, либо настроить Jetty таким образом, чтобы GraphHopper открывался по адресу <http://localhost:8180>. Этим способом мы и воспользуемся: просто переименуйте файл *graphhopper.war* в *root.war* и перезапустите Jetty.

Ссылки

1. [GraphHopper](#)
2. [GraphHopper docs](#)
3. [Routing Web API Docs](#)
4. [ШТОСМ: Graphhopper](#)

[Обсудить в форуме](#) Комментариев — 7

Последнее обновление: 2015-01-11 08:59

Дата создания: 23.03.2014

Автор(ы): [Denis Rykov](#)