Eve - это фреймворк Python REST API с открытым исходным кодом. Он позволяет легко создавать и развертывать высоко настраиваемые, полнофункциональные веб-сервисы на основании структуры REST. Eve написана и поддерживается Николой Яроччи.

Eve предоставляет надежную, многофункциональную, REST-централизованную реализацию API. Вам просто нужно настроить параметры и поведение API, подключить источник данных, и вы готовы к работе.

Настройки API хранятся в стандартном модуле Python (по умолчанию используется settings.py), что делает настройку довольно тривиальной задачей. Кроме того, можно расширить некоторые ключевые функции, а именно аутентификацию и авторизацию, проверку данных и доступ к данным, предоставив движку EVE пользовательские объекты.

Eve работает на базе Flask и Cerberus и предлагает встроенную поддержку хранилищ данных MongoDB. Поддержка бэкендов SQL, Elasticsearch и Neo4js обеспечивается расширениями сообщества.

Кодовая база тщательно протестирована под Python 2.7, 3.5+ и PyPy.

Установка Eve достаточно проста, так как является обычным пакетом Python.  
$ pip install eve

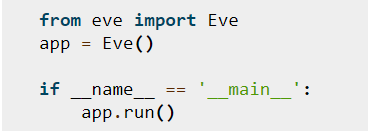
После установки пакета необходимо создать два файла в директории проекта: **run.py** и **settings.py**

Файл **run.py** содержит основной код программы.

В файле **settings.py** находится описание конфигурации приложения: подключение к базе данных, разрешенные http запросы, домены, заголовки, а так же описание структуры базы данных.

#### Минимальное приложение

Минимальное приложение Eve выглядит примерно так:



Сохраните его как run.py. Затем создайте новый текстовый файл со следующим содержимым:



Сохраните его как settings.py в том же каталоге, где хранится run.py. Это файл конфигурации Eve, стандартный модуль Python, и он говорит Eve, что ваш API состоит только из одного доступного ресурса - people.

Теперь вы готовы запустить свой API:



Теперь вы можете использовать API:





Точки входа API придерживаются принципа HATEOAS и предоставляют информацию о ресурсах, доступных через API. В нашем случае доступен только один дочерний ресурс - people.

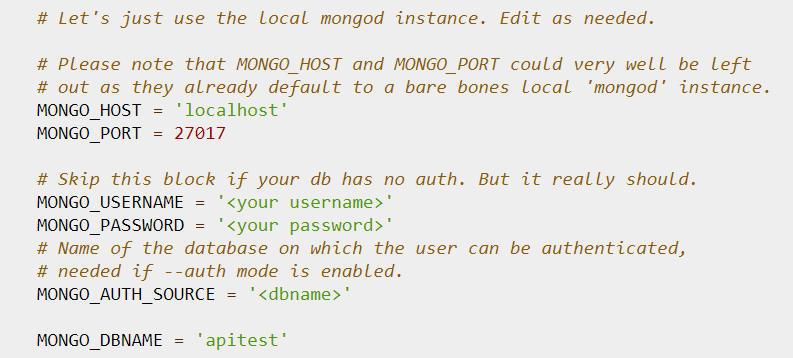
Попробуйте запросить people прямо сейчас:





На этот раз мы также получили список \_items. \_links относятся к доступному ресурсу, поэтому вы получаете ссылку на родительский ресурс (домашнюю страницу) и на сам ресурс. Если вы получили ошибку тайм-аута от pymongo, убедитесь, что предварительные требования выполнены. Скорее всего, серверный процесс mongod не запущен.

Давайте подключимся к базе данных, добавив следующие строки в settings.py:



Из-за “лени” MongoDB нам на самом деле не нужно создавать коллекции баз данных. На самом деле нам даже не нужно создавать базу данных: GET запросы на пустую/несуществующую БД будут обслуживаться правильно (200); DELETE/PATCH/PUT получат соответствующие ответы (404), а POST запросы создадут базу данных и коллекции по мере необходимости. Однако такая автоматически управляемая база данных будет работать очень плохо, поскольку в ней отсутствуют индексы и какая-либо оптимизация.

#### Возможности EVE

#### Конфигурация

Как правило, настройка Eve лучше всего выполняется с помощью конфигурационных файлов. Сами конфигурационные файлы являются файлами Python. Однако Eve сначала отдаст приоритет настройкам на основе словаря, затем попытается найти файл, переданный в переменную среды EVE\_SETTINGS (если она установлена), и, наконец, попытается найти settings.py или файл с именем файла, переданным флагу настроек в конструкторе.