

попробовать свои силы по взлому хотя бы простейших вариантов современных криптографических систем. Три из шести предлагаемых для проведения лабораторных работ посвящены теории чисел, четвертая – историческим системам защиты информации, пятая и шестая – самым актуальным современным криптосистемам RSA и Эль Гамала. Изучаемые темы разработаны по следующему алгоритму: сначала идет изложение необходимого теоретического материала, затем разбирается решение типичных для данной темы задач и, наконец, предлагаются задания по вариантам для выполнения конкретной самостоятельной или лабораторной работы. Приведенные примеры решения типовых задач по изучаемой теме делают материал доступным для понимания, облегчают его усвоение обучающимися, в том числе и при самостоятельной работе.

Также для курсантов специальности «Телекоммуникационные системы» введен факультативный курс «Защита информации». Факультатив дополняет знания курсантов в теории конечных полей и способствует более качественному усвоению практических аспектов курса «Прикладная математика».

Таким образом, преподавательским составом кафедры высшей математики проводится целенаправленная и поступательная работа по приведению тематики и содержания учебного процесса в соответствие современным требованиям.

Список используемых источников

1. Мур, А. Руководство по безопасности бизнеса: практическое пособие по управлению рисками / А. Мур, К. Хиарден. – М. : Филинь, 1998. – 328 с.
2. Лебедев, С.А. Философия науки : учеб. пособие для вузов / С.А. Лебедев. – 2-е изд., перераб. и доп. – М. : Юрайт, 2015. – 296 с.
3. Липницкий, В.А. Защита информации : практикум / В.А. Липницкий, Л.В. Михайловская, Е.В. Валаханович. – Минск : ВА РБ, 2012. – 87 с.
4. Липницкий, В.А. Современная прикладная алгебра. Математические основы защиты информации от помех и несанкционированного доступа : учеб.-метод. пособие / В.А. Липницкий. – Минск, 2006. – 88 с.

УДК 004.82

Е.А. Германович, И.А. Гурин, А.Л. Михняев

Брестский государственный технический университет

ВАЛИДАЦИЯ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ OVERPASS API

На примере веб-проекта, реализующего доступ к картографической информации, демонстрируется применение Overpass API. Целью проекта является получение информации о расположении банкоматов. Применение API OpenStreetMap и Overpass API позволяет получить доступ к картографической информации и извлекать данные из базы OSM по пользовательским запросам. Результатами запроса являются данные в формате JSON. Приведены примеры запросов и результатов их выполнения. Опыт разработки будет полезен специалистам в области веб-проектирования и мобильных приложений.

Ключевые слова: API, веб-приложения, API Overpass, API OpenStreetMap, валидация картографических объектов.

Введение. API (англ. Application Programming Interface – программный интерфейс приложения) – это набор способов и правил, по которым различные программы «общаются» между собой и обмениваются данными (далее API).

Разнообразные API механизмы [1] позволяют решать широкий круг задач взаимодействия программных компонентов. Примерами различных типов API являются [3]: веб-API (Web APIs), SDK (Software Development Kit), API операционных систем, API DB/DWH, графические API (DirectX, OpenGL), API социальных сетей (Facebook, Twitter, Instagram), файловые API (Amazon S3, Google Cloud Storage и др.), мобильные API (Android API, iOS API) и т.д.

В контексте работы с сервером, API (см., например, [8]) – это механизмы, которые позволяют двум программным компонентам взаимодействовать друг с другом, как правило, на основе технологии «клиент-сервер», используя набор определений и протоколов. Например, Национальный Банк Республики Беларусь имеет подробное описание класса Currency как компонента API для получения официального курса белорусского рубля по отношению к иностранным валютам [2]. Подробное описание механизмов API дано, например, в [1], [4] и др.

Нами было реализовано веб-приложение с использованием API OpenStreetMap (OSM) [5] для получения информации о банкоматах в Беларуси. OSM – это свободная, изменяемая пользователями карта мира, создаваемая и поддерживаемая в режиме свободного доступа. OSM предоставляет данные о различных географических объектах (дорогах, зданиях, офисах и пр.)

При реализации проекта, кроме API OSM, был также использован API Overpass. Он позволяет отправлять запросы на извлечение данных из базы OSM и получать данные в формате JSON. Примеры реализации запросов даны во фрагментах 1 и 2.

Фрагмент 1. Текст запроса на получение информации о банкоматах в Минске:

```
[out:json];
area["ISO3166-1"="BY"]->.searchArea;
(
  node["amenity"="atm"](area.searchArea);
);
out body;
>;
out skel qt;
```

В приведенном запросе ограничивается область поиска банкоматов только территорией Беларуси (ISO3166-1="BY") и выполняется поиск всех узлов (node), имеющих тег "amenity" равный "atm", что обозначает узлы, являющиеся банкоматами. Также "atm" может являться тегом в том случае, если банкомат находится непосредственно в банке. В таком случае тег "amenity" будет равен значению "bank", а появившийся тег "atm" будет содержать в себе значение "yes" (показывать, что в выбранной банковской точке присутствует банкомат).

Ответом на запрос является JSON-строка с данными о банкоматах, включая их координаты, адрес и режим работы.

Для фильтрации результатов по выдаваемой банкоматом валюте мы можем, в соответствии с параметрами API, добавить в запрос необходимые условия. Во фрагменте 2 приведен пример запроса для получения информации о банкоматах города Минска, которые выдают валюту «евро» или «белорусские рубли».

Фрагмент 2. Текст запроса на получение информации о банкоматах, выдающих обозначенную валюту в г. Минске

```
[out:json];
area["ISO3166-1"="BY"]->.searchArea;
(
  node["amenity"="atm"](area.searchArea)[ATM_currency="EUR" or
  ATM_currency="BYN"];
);
out body;
>;
out skel qt;
```

Запрос содержит условие поиска банкоматов с тегом "ATM_currency" со значениями "EUR" или "BYN". Ответом на запрос ожидается JSON-строка с данными о банкоматах, удовлетворяющих условия.

Использование API OSM и Overpass позволяет получить различную информацию о расположении объектов, удовлетворяющих условия. Под условиями мы понимаем ограничение параметров поиска по территории и другим признакам. Исходя из общего представления об описанных API, они могут быть использованы в таких проектах, как мобильные приложения, веб-сайты, десктопные приложения для географической локализации различных ресурсов и услуг и т. д.

Валидация позволяет улучшить качество данных, устранить ошибки и несоответствия, а также упростить использование таких данных в различных приложениях и сервисах. Кроме того, валидация объектов может быть полезна для создания точных и детальных карт, которые могут использоваться в различных сферах, таких как туризм, навигация, планирование маршрутов, геоинформационные системы и другие.

Одна из проблем, с которой часто сталкиваются клиенты, заключается в том, что снятие наличных денег в банкоматах сторонних банков требует дополнительных расходов из-за взимаемой комиссии. Для решения этой проблемы может быть использовано наше приложение, которое предлагает возможное решение. Приложение для валидации банкоматов позволяет пользователям найти ближайший банкомат банка, выпустившего используемую банковскую карту. В приложении использованы средства интеграции банка клиента и его партнеров. Пользователю предоставляется возможность выбрать нужный банк в списке на рисунке 1.



Рисунок 1 – Фрагмент интерфейса программы

Запрос, генерируемый автоматически, выдает информацию о расположении на карте ближайших банкоматов, обслуживаемых выбранным банком (рисунок 2). На основе полученной информации пользователь выбирает оптимальный маршрут до точки назначения.

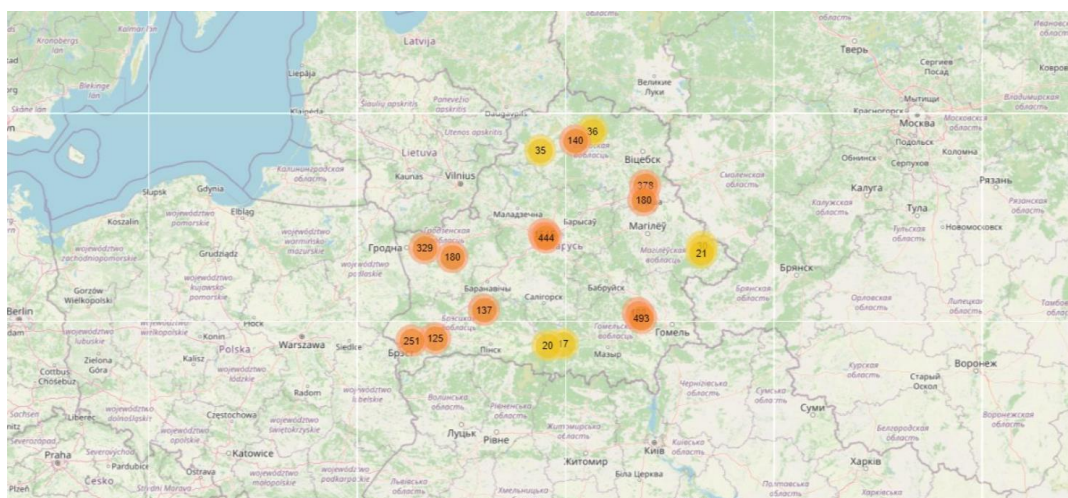


Рисунок 2 – Фрагмент карты с отображением объектов (банкоматов), удовлетворяющих заданному условию

Описанный способ применения API может быть использован для разработки приложений различных типов для решения широкого круга задач. Кроме того, приведенное описание демонстрирует применение одного из API – OpenStreetMap.

Представленные идеи могут быть расширены за счет дополнения различных функций, интеграции с другими сервисами, включением мультязычной поддержки и прочего. Такой подход может быть использован при создании корпоративных, учебных, социальных, игровых приложений и сервисов.

Список использованных источников

1. Amazon API Gateway. Amazon Web Services [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/what-is/api>. – Дата доступа: 20.04.2024.
2. API Национального банка Республики Беларусь [Электронный ресурс]. – Режим доступа: <https://www.nbrb.by/apihelp/exrates>. – Дата доступа: 20.04.2024.
3. API от А до Я (теория и практика) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/768752>. – Дата доступа: 30.03.2024.
4. Google Cloud API [Электронный ресурс]. – Режим доступа: <https://console.cloud.google.com/apis/library>. – Дата доступа: 20.04.2024.
5. OpenStreetMap [Электронный ресурс]. – Режим доступа: <https://www.openstreetmap.org>. – Дата доступа: 20.04.2024.
6. Головкин, В.А. Глубокое обучение для детектирования объектов на изображениях документов / В.А. Головкин // Физика, математика, информатика. Вестник БрГТУ. – 2017. – № 5. – С. 2–9.
7. Головкин, В.А. Распознавание крон деревьев на изображении с применением нейросетевых методов / В. А. Головкин // Физика, математика, информатика. Вестник БрГТУ. 2019. – № 5. – С. 18–24.
8. Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API [Электронный ресурс]. – Режим доступа: <https://tproger.ru/articles/web-api>. – Дата доступа: 16.04.2024.