



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА – Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий
Кафедра инструментального и прикладного программного обеспечения

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4
по дисциплине
«Разработка серверных частей интернет ресурсов»

Выполнил студент группы ИКБО-01-20

Баикин К. Е.

Принял

Волков М.Ю.

Практическая работа выполнена «__»_____ 2022 г. _____

«Зачтено» «__»_____ 2022 г. _____

Москва 2022

СОДЕРЖАНИЕ

Цель работы	3
Задание на практическую работу	3
Выполнение практической работы	4
Вывод.....	6
Ответы на вопросы к практической работе.....	6
Ссылка на удалённый репозиторий проекта	10
Список использованной литературы.....	10

Цель работы

Получить навыки создания интерфейсов прикладного программирования.

Задание на практическую работу

Предполагается реализация интерфейса прикладного программирования для доступа к некоторым данным по варианту. Предполагается реализация серверной части обработки запросов и тестирование данного интерфейса с использованием программы Postman. Для реализации данного сервиса предлагается использовать серверную конфигурацию, модернизированную в течение первых трех практических работ. Важной частью данной практической работы является сохранение функциональности реализованной в практической работе No3. То есть интерфейс предлагается создать уже в существующем веб-приложении. Также предполагается использование темы практической работы No3 для продолжения модернизирования собственной системы. Изменение темы согласовывается отдельно с преподавателем. Хранение данных предполагается уже в существующей базе данных.

Технические требования к реализации интерфейса:

1. Доступ как минимум к 2 независимым сущностям.
2. Реализация как минимум операций группы CRUD (создание, чтение, обновление, удаление). Приветствуется реализация дополнительной функциональности.
3. Тестирование всех функциональных возможностей созданного интерфейса с использованием программы Postman.

Варианты:

- Кофейня,
- Строительный магазин,
- Автосервис,
- Магазин электроники,
- Портфолио,
- Библиотека,
- Ресторан,
- Погода,
- Ломбард,
- Магазин игрушек,
- Индивидуальная тема.

Выполнение практической работы

Для выполнения данной работы за основу была взята предыдущая. Интерфейс прикладного программирования представляет из себя REST API. Работа дополнена структурой из одного модуля содержащего 2 php файла (эндпоинта). Первый используется для взаимодействия с таблицей пользователей, вторая – с таблицей ценностей. Каждый запрос принимает спецификатор действия – action, который определяет какой функционал будет выполняться.

Первый эндпоинт (/api/users.php) предоставляет обработку следующих запросов (каждый параметризован по схеме x-www-form-urlencoded):

1. POST запрос (спецификатор create) — создание пользователя. Принимает обязательные name и password.
2. GET запрос (спецификатор read) — получение списка пользователей. Принимает необязательные offset и limit.

Второй эндпоинт (/api/catalog.php) предоставляет обработку запросов прикладных задач для таблицы каталога:

1. POST запрос (спецификатор create) — создание товара. Принимает обязательные name, desc и price.
2. GET запрос (спецификатор read) — получение списка товаров. Принимает необязательные offset и limit.

Работа запросов была протестирована с использованием программы Postman, проверки представлены на рисунках 1-6.

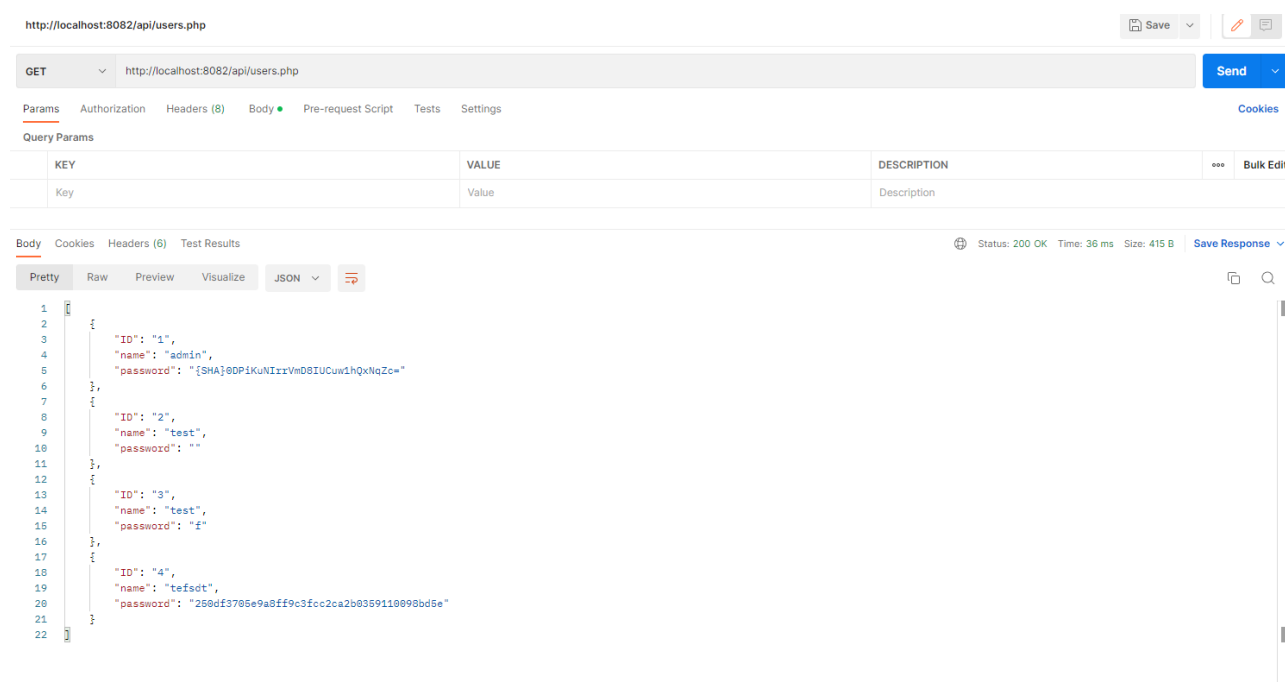


Рисунок 1 — Получение пользователей

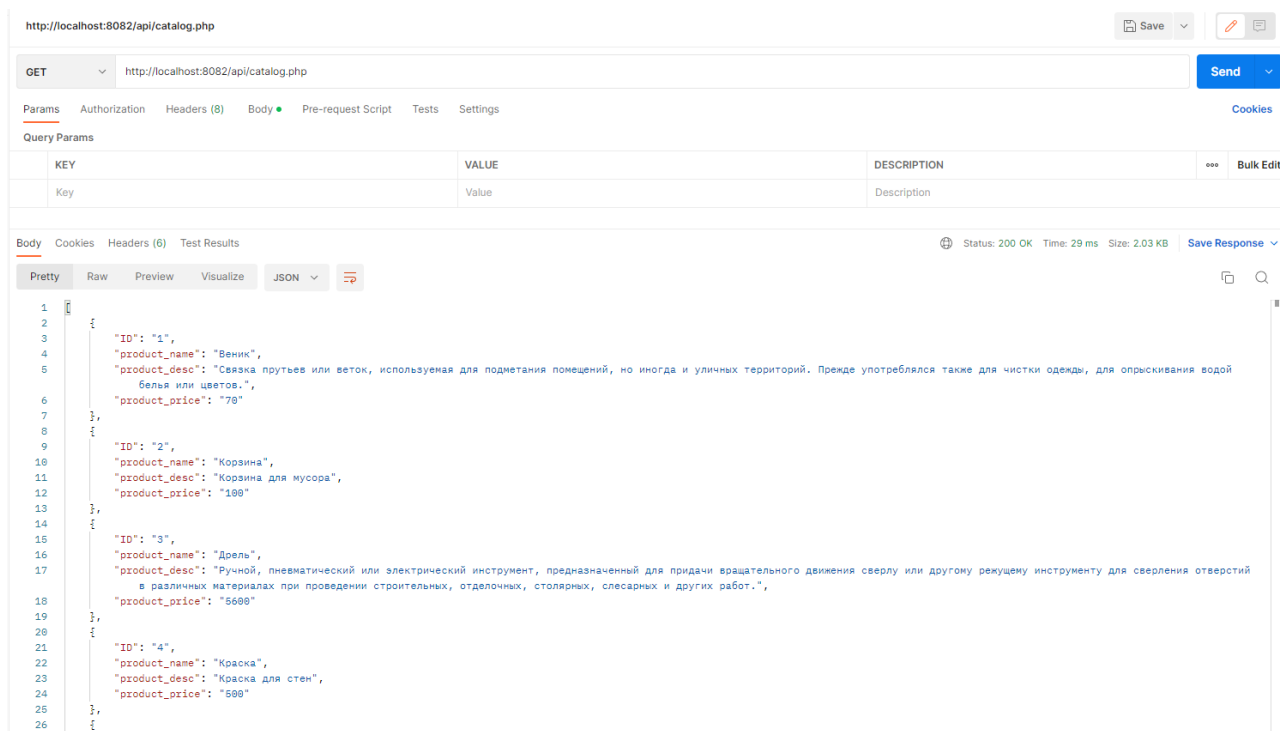


Рисунок 2 — Получение списка товаров

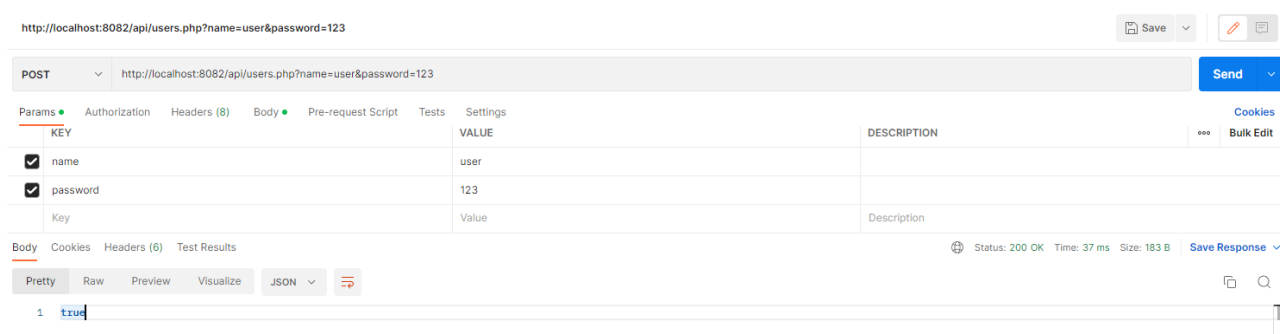


Рисунок 3 — Создание пользователя

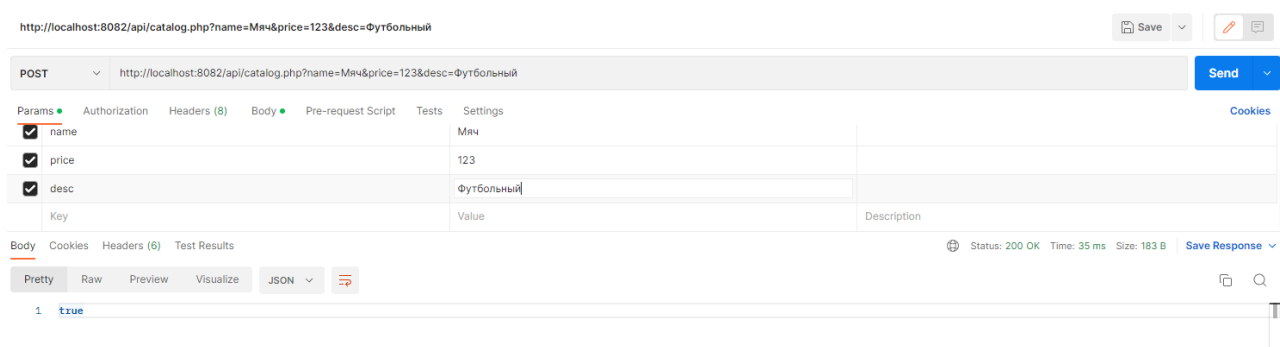


Рисунок 4 — Создание товара

Вывод

В ходе выполнения данной практической работы были получены навыки написания API веб-сервера на языке PHP, были опробованы различные типы запросов к базе данных MySQL, а также получен опыт написания продуктовой логики на языке PHP в объектно-ориентированном стиле.

Ответы на вопросы к практической работе

1. Что такое HTTP-запрос?

HTTP запросы - это сообщения, отправляемые клиентом, чтобы инициировать реакцию со стороны сервера.

2. Опишите существующие HTTP-запросы.

GET, POST, PUT, PATCH, DELETE, HEAD, CONNECT, TRACE, OPTIONS

3. Опишите обработку запроса на PHP. Что нужно использовать, как вычленивать параметры запроса?

```
$data=json_decode(file_get_contents("php://input"));
```

Данные, отправленные с помощью POST-запроса доступны в PHP в суперглобальном массиве `$_POST`.

4. Опишите создание HTML-форм на PHP.

Одно из главных достоинств PHP - то, как он работает с формами HTML. Здесь основным является то, что каждый элемент формы автоматически становится доступным вашим программам на PHP.

5. Что такое API?

Программный интерфейс приложения

6. Опишите API как средство интеграции приложений.

API позволяет интегрировать приложения уже на этапе их проектирования

7. Что такое Web API?

API, реализованное для веб-сервиса

8. Приведите пример API.

Telegram API

9. Что такое REST?

REST (от англ. Representational State Transfer — «передача репрезентативного состояния» или «передача „самоописываемого“ состояния») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

10. Как организована передача данных в архитектуре REST?

- Клиент-серверная архитектура
- Stateless
- Кэширование
- Единообразие интерфейса
- Layered system
- Code on demand

11. Как организована работа REST?

Через обращение к ресурсам по уникальным URL

12. Что такое SOAP?

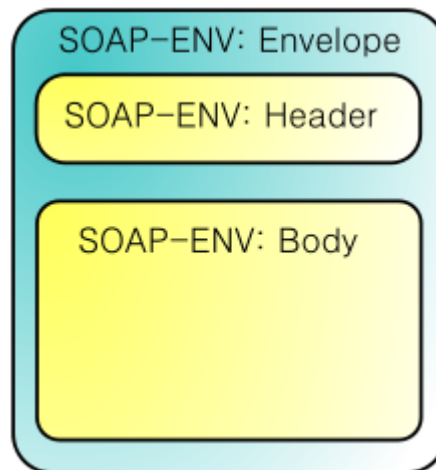
SOAP (от англ. Simple Object Access Protocol — простой протокол доступа к объектам) — протокол обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC).

13. Чем SOAP отличается от REST?

REST был создан для решения проблем SOAP. Поэтому у него более гибкая архитектура. Он состоит только из простых рекомендаций и позволяет разработчикам реализовывать рекомендации по-своему. Он допускает различные форматы сообщений, такие как HTML, JSON, XML и простой текст, в то время как SOAP допускает только XML. REST также является более легкой архитектурой, поэтому веб-сервисы RESTful имеют более высокую производительность.

14. Для чего нужен SOAP-процессор?

15. Опишите общую структуру SOAP-сообщения.



16. Что такое и что содержит Конверт (SOAP Envelope)?

Envelope — корневой элемент, который определяет сообщение и пространство имен, использованное в документе.

17. Что такое и что содержит Заголовок SOAP (SOAP Header)?

Header — содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации.

18. Что такое и что содержит Тело SOAP (SOAP Body)?

Body — содержит сообщение, которым обмениваются приложения.

19. Опишите SOAP-сообщение с вложением.

См. вопрос 15.

20. Что такое graphql?

Язык запросов данных и язык манипулирования данными с открытым исходным кодом для построения веб ориентированных программных интерфейсов. GraphQL был разработан как внутренний проект компании Facebook в 2012 году, а позднее в 2015 году был выпущен публично.

21. Что такое Распознаватели (resolvers) в graphql?

Используя распознаватель GraphQL понимает, как и где получить данные, соответствующие запрашиваемому полю.

22. Из чего состоит экосистема graphql, что нужно, чтобы использовать данную технологию?

На практике GraphQL API построен на трёх основных строительных блоках: на схеме (schema), запросах (queries) и распознавателях (resolvers).

23. Что такое валидация данных и для чего она нужна?

Валидация данных (англ. Data validation) — это процесс проверки данных различных типов по критериям корректности и полезности для конкретного применения.

24. Где и когда выполнять валидацию данных?

Желательно валидировать данные как можно раньше: это упрощает код и снижает нагрузку на центральные узлы системы при распределённом сборе.

25. Как выполнять валидацию данных?

Наибольшее практическое применение находят методы, которые можно применить сразу в момент ввода данных в систему:

Проверки типа данных

Простая проверка диапазона и ограничений

Проверка кода и перекрестных ссылок

26. Приведите пример с поэтапной валидацией данных.

Проверка данных формы на клиенте, проверка в коде сервера, проверка при занесении в базу данных.

27. Что такое запрос и мутация в graphql и чем они отличаются?

В REST любой запрос может привести к некоторым побочным эффектам на сервере, но по соглашению предполагается, что GET-запросы не используются для изменения данных. GraphQL похож, - технически любой запрос может быть реализован, для того, чтобы перезаписать данные. Однако полезно установить соглашение о том, что любые операции, вызывающие изменения данных, должны быть отправлены явным образом через мутации.

Ссылка на удалённый репозиторий проекта

Код решения доступен по ссылке: <https://github.com/kiruxa-projects/rschir-4>

Список использованной литературы

1. Официальная документация языка PHP, URL: <https://www.php.net/manual/ru/index.php>
2. Справочник языка | Руководство по PHP | PHP.RU , URL: <https://php.ru/manual/>
3. Простой REST API в PHP — Пошаговое руководство | Only to top, URL: <https://only-to-top.ru/blog/programming/2019-11-06-rest-api-php.html>