**Инструкция по работе веб-API с помощью коллекций Postman**

**1. Создание коллекции**

Коллекция в Postman — это серия HTTP-запросов. Postman сохраняет все аспекты запросов, включая заголовки и тела сообщений. Поэтому **вы можете запускать запросы последовательно как полуавтоматические тесты**.

Начнем с создания новой коллекции. Можно щелкнуть стрелку раскрывающегося списка на кнопке «Создать» и выбрать «Коллекция».

Когда появится диалоговое окно CREATE A NEW COLLECTION, можно назвать нашу коллекцию, к примеру, как тестируемый нами интернет-магазин **«Dikiyi\_sbor»**.Наконец, нажимаем кнопку «Создать», чтобы наша новая коллекция появилась в списке слева.

Как только коллекция создана, можно навести на нее курсор, чтобы открыть две кнопки меню. Кнопка со стрелкой открывает выпадающую правую панель, которая обеспечивает доступ к средству запуска коллекции. И наоборот, кнопка с многоточием открывает раскрывающееся меню, содержащее ряд операций над коллекцией.

**2. Добавление запросов в коллекцию**

Добавление запросов в коллекцию — это следующий шаг. Для этого нужно выбрать созданную коллекцию и нажать на кнопку "Add Request". В открывшемся окне можно задать метод запроса, URL, заголовки и тело запроса. Например, для отправки GET-запроса к API нужно выбрать метод GET и указать URL.

**Пример:**

GET https://api.example.com/users

Для POST-запросов нужно также указать тело запроса. Например, для создания нового пользователя:

POST https://api.example.com/users

{

"name": "John Doe",

"email": "john.doe@example.com"

}

После настройки запроса его можно сохранить в коллекцию, нажав на кнопку "Save". Это позволяет вам легко повторно использовать запросы и делиться ими с коллегами. Вы также можете добавлять комментарии к запросам, чтобы объяснить их назначение или указать на особенности их использования.

**3. Тестирование и автоматизация запросов**

Postman позволяет не только отправлять запросы, но и автоматизировать их тестирование. Для этого можно использовать встроенный язык тестирования Postman — JavaScript. В разделе "Tests" можно написать скрипты, которые будут выполняться после получения ответа на запрос.

Пример теста:

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Response time is less than 200ms", function () {

pm.expect(pm.response.responseTime).to.be.below(200);

});

Эти тесты проверяют, что код ответа равен 200 и что время ответа меньше 200 миллисекунд. Postman также поддерживает выполнение тестов в коллекциях, что позволяет автоматизировать процесс тестирования. Можно создавать сложные сценарии тестирования, которые будут выполняться автоматически, что значительно упрощает процесс проверки качества API.

**4. Как работать с коллекцией Dikiyi\_sbor**

В нашей коллекции 5 запросов с методом POST:

* Добавление одного товара
* Отправка формы с корзины
* Изменение количества товаров
* Удаление товара
* Отображение пустой корзины

Используется переменная {{main\_url}} = <https://www.sibdar-spb.ru/>

В тестах используются следующие скрипты:

* **Проверка статус кода:**

pm.test("Status code is 200", **function** () {

pm.response.to.have.status(200);

});

* **Проверка, что тело ответа содержит строку:**

pm.test("Body matches string", **function** () {

pm.expect(pm.response.text()).to.include("Корзина пуста, необходимо это исправить");

});

* **Проверка, что статус код имеет статус ОК:**

pm.test("Status code name has string", **function** () {

pm.response.to.have.status("OK")}).

Авторизация на сайте не требуется, так как на сайте нет личного кабинета и заказы принимаются через обратную связь по телефону.

Коллекция имеет ряд заголовков и заполненные

данные в теле запроса.

**5. Советы по улучшению работы в Postman**

**Группируйте запросы:** Разделяйте запросы на логические группы внутри коллекций. Это поможет лучше организовать тестирование и упростит навигацию. Например, вы можете создать группы для запросов, связанных с пользователями, продуктами, заказами и т.д.

**Документируйте запросы:** Добавляйте описания и комментарии к запросам. Это поможет вам и вашим коллегам лучше понимать, что делает каждый запрос. Вы также можете использовать встроенные возможности Postman для генерации документации, которая будет автоматически обновляться при изменении запросов.

**Используйте переменные:** Переменные позволяют делать запросы более гибкими и повторно используемыми. Например, вместо жестко закодированных значений используйте переменные для URL, токенов и других параметров. Это особенно полезно, если вам нужно тестировать один и тот же запрос с различными значениями.

**Автоматизируйте тестирование:** Используйте встроенные возможности Postman для автоматизации тестирования. Это поможет вам быстро выявлять ошибки и обеспечивать качество API. Вы можете создавать сложные сценарии тестирования, которые будут выполняться автоматически, и получать уведомления о результатах тестирования.

**Используйте коллекции для документации:** Коллекции Postman можно использовать не только для тестирования, но и для документации API. Вы можете добавлять описания и комментарии к запросам, а также генерировать документацию, которая будет автоматически обновляться при изменении запросов.

**Следите за обновлениями:** Postman постоянно развивается и добавляет новые функции. Следите за обновлениями и используйте новые возможности, чтобы сделать процесс тестирования еще более эффективным.