**HTTP протокол**

• HTTP протокол используется для передачи данных между браузером и сервером.

• Запросы бывают разных типов: GET, POST, PUT, DELETE.

**GET (получить) запросы**

• GET запросы используются для получения данных с сервера.

• URL-адрес запроса является основным идентификатором запроса.

• GET запросы могут содержать параметры, которые передаются на сервер для настройки ответа.

• Идентепотерные GET запросы характеризуются отсутствием изменений параметров запроса (т.е. из раза в раз на сервер отправляется одинаковый запрос)

**Примеры использования GET запросов**

• При увеличении фотографии на сайте, браузер делает GET запрос для получения данных.

• При настройке фильтров в поиске на сайте, URL-адрес запроса может содержать параметры, которые указывают на настройки фильтра.

**Различия между GET и POST запросами**

• GET запрос используется для получения данных с сервера, но не для их модификации.

• POST запрос используется для отправки данных на сервер и для обновления информации.

• POST запросы считаются более безопасными, так как их сложнее подделать.

**Особенности GET и POST запросов**

• GET запросы имеют только URL-адрес и параметры, в то время как POST запросы имеют тело запроса, в которое можно поместить дополнительные данные.

• GET запросы не могут загружать файлы на сервер, в то время как POST запросы могут.

**Использование PUT и DELETE запросов**

• PUT запрос используется для создания новых объектов на сервере, а DELETE запрос - для удаления объектов.

• Эти запросы не имеют тела запроса и используются в REST API.

**Интерфейс и API**

• Интерфейс - это то, через что мы взаимодействуем с объектом, например, пульт от телевизора или руль автомобиля.

• API (application programming interface) - это интерфейс программы, который позволяет нам взаимодействовать с программным продуктом.

• API может быть разных видов, например, библиотеки или программные продукты.

**Примеры интерфейсов**

• Шариковая ручка, дверная ручка и интерфейс телевизора.

• В каждом случае интерфейс позволяет пользователю взаимодействовать с объектом, используя только те функции, которые разработчик разрешил использовать.

**Публичные и приватные методы и свойства**

• Методы и свойства, которые доступны извне, называются публичными, а те, которые доступны только внутри объекта, называются приватными.

• Приватные методы и свойства могут быть защищены от взлома, но это не гарантирует полной безопасности.

**Интерфейс библиотеки**

• Интерфейс библиотеки включает в себя описание методов, их параметров и возвращаемых значений.

**Веб-серверы и REST API**

• Веб-серверы - это программные продукты, которые принимают запросы и возвращают ответы.

• REST API определяет правила формирования URL-адресов, типы запросов и ответов.

• Важно знать, какие запросы и параметры отправлять, и какой ответ ожидать.

**REST API и URL-адреса**

• REST API использует один и тот же URL-адрес для всех запросов, но типы запросов определяют, какие параметры и ответы ожидать.

• Важно правильно формировать URL-адреса и отправлять запросы на сервер.