**REST**

**REST (Representational State Transfer) — это архитектурный стиль. В отличие от SOAP, который является протоколом, он не подчиняется строгим стандартам, более гибкий, простой и легковесный.**

**В основе REST лежат стандартные HTTP-методы (GET, POST, PUT, DELETE), а основным форматом данных для него является JSON. Также он позволяет передавать данные в виде текста, XML, HTML и другие.**

**JSON**

**JSON (JavaScript Object Notation) — это легковесный формат обмена данными, который удобен для чтения и написания для человека, а также для анализа и генерации машинами. Его синтаксис основан на языке JavaScript, но сам JSON существует независимо и может использоваться с другими языками программирования.**

**SOAP**

**SOAP (Simple Object Access Protocol) - это протокол для обмена структурированными сообщениями в распределенной среде.**

**SOAP использует XML для передачи сообщений и может работать с различными сетевыми протоколами. Для отправки запросов используется метод POST, в отличие от REST, который может использовать все стандартные HTTP-методы.**

**XML**

**XML (eXtensible Markup Language) — это расширяемый язык разметки, предназначенный для структурирования, хранения и передачи данных. В отличие от HTML он не используется в верстке.**

**WSDL**

**Для описания функциональности веб-сервиса используется файл, написанный на WSDL (Web Services Description Language), основанный на XML, в котором содержится информация о получаемых и отправляемых данных, форматах сообщений, методах.**

**WSDL является обязательным элементом любого SOAP-сервиса, без которого он не может существовать.**

**В тестировании вы будете в первую очередь работать с файлами WSDL. Об особенностях их практического использования мы поговорим на следующих уроках.**

* **API (Application Programming Interface) – интерфейс прикладного программирования. По сути, это механизм, позволяющий разным программам общаться друг с другом.**
* **Типы API: Локальные API для взаимодействия в рамках одной системы и удалённые API для связи через сеть.**
* **Способы вызова API**
* **Вызов функции системой: Это автоматизированный процесс, при котором программные компоненты одной системы общаются между собой через внутренние API.**
* **Вызов метода другой системой: Здесь одна система использует API другой системы для получения доступа к функциям или данным. Это основа микросервисной архитектуры, где сервисы взаимодействуют через хорошо определённые интерфейсы.**
* **Вызов метода человеком через GUI: Пользователи взаимодействуют с API через графический интерфейс, который скрывает сложность API. Примеры включают веб-формы, которые отправляют данные через API.**
* **Как работает удаленный API:**
* **Вызов операции с использованием методов GET, POST и других HTTP-методов**
* **Входные данные передаются через HTTP Request (HTTP-запрос).**
* **Выходные данные получаются через HTTP Response (HTTP-ответ).**

**GET: GET**-запросы используются для получения данных от API.

**POST: POST**-запросы используются для отправки новых данных API.

**PUT: PUT**-запросы используются для обновления уже существующих данных.

**PATCH: PATCH**-запросы (как и PUT) используются для обновления уже существующих данных. Разница в том, что с помощью PATCH запросов можно обновить несколько записей за раз.

**DELETE: DELETE**-запросы используются для удаления существующих данных.