*REST*-архитектура базируется на системе свойств и ограничений, у неё довольно много особенностей. Отметим наиболее важные:

* 1. Архитектура приложения должна быть типа клиент-сервер. *HTTP-*протокол также работает только в клиент-серверных системах.
  2. Сервер не должен хранить какой-либо информации о клиентах. В запросе должна храниться вся необходимая информация для обработки запроса и, если необходимо, идентификации клиента. Состояние сессии сохраняется на стороне клиента. Это также согласуется с концепцией *HTTP*-протокола.
  3. Наличие кэша. Клиенты и промежуточные узлы могут кэшировать ответы от сервера (напоминаем, **кэширование** — это процесс сохранения данных локально, который позволяет быстрее получить к ним доступ при будущих запросах).
  4. Слои системы. В *REST-*системе клиент не может точно определить, взаимодействует ли он напрямую с сервером или между ними есть промежуточный узел такой, как прокси-сервер или туннель. Добавление промежуточного узла не должно приводить к обновлению кода приложения на клиентской или на серверной частях.
  5. Код по требованию. *REST* может позволить расширить функциональность клиента за счёт загрузки кода с сервера, например *Java Script*.
  6. Унифицированный интерфейс. Данное ограничение является фундаментальным требованием *REST-*сервисов. Унифицированный интерфейс имеет 4 признака:

Идентификация ресурса. Каждая единица информации однозначно определяется *URL* — это значит, что *URL* по сути является первичным ключом для единицы данных. Этот идентификатор не меняется в зависимости от изменения состояния ресурса.

Управление ресурсом через представления. Представление в *REST* используется для выполнения действий над ресурсами. Представление являет собой текущее состояние ресурса, передаваемое в теле запроса или ответа. Формат представления может быть *XML*, *HTML*, *JSON*, *YAML* или даже обыкновенный текстовый формат.

Запрос и ответ должны хранить в себе всю необходимую информацию для их обработки. Не должно быть дополнительных сообщений для обработки одного запроса. Другими словами, отсутствие состояния, сохраняемого между запросами к ресурсам.

Состояние ресурса передается через содержимое запроса или ответа, параметры строки запроса, заголовки запросов и ответов. В случае необходимости ссылки на ресурс могут содержаться в теле ответа (или заголовках) для формирования *URI* или извлечения запрошенных объектов.

Перечисленные выше признаки выглядят сложно из-за того, что они изложены в виде сухой теории. Однако рекомендуем зафиксировать информацию в удобном для вас формате и вернуться к ней позже, после прохождения этого и следующего модуля. С учётом полученной практики понимание изложенных выше особенностей архитектуры *REST* будет более полным.