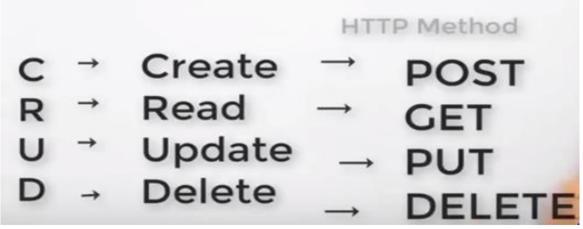

				HTTP Method
C	→	Create	→	POST
R	→	Read	→	GET
U	→	Update	→	PUT
D	→	Delete	→	DELETE

AJAX basically refers to making asynchronous request in JavaScript, traditionally sending/receiving XML (although nowadays, JSON is often used instead of XML). So that's the technique you use on client-side.

REST is a concept for HTTP request exchange, so you're making RESTful request calls (e.g. 'get') against the REST-API you implemented on server side.

1) Rest – **webHttpBinding**
2) SOAP - **basicHttpBinding** and **wsHttpBinding**

- **webHttpBinding** is the REST-style binding, where you basically just hit a URL and get back a truckload of XML or JSON from the web service
- **basicHttpBinding** and **wsHttpBinding** are two SOAP-based bindings which is quite different from REST. SOAP has the advantage of having WSDL and XSD to describe the service, its methods, and the data being passed around in great detail (REST doesn't have anything like that - yet). On the other hand, you can't just browse to a wsHttpBinding endpoint with your browser and look at XML - you have to use a SOAP client, e.g. the WcfTestClient or your own app.

So your first decision must be: REST vs. SOAP (or you can expose both types of endpoints from your service - that's possible, too).

Then, between basicHttpBinding and wsHttpBinding, there differences are as follows:

- basicHttpBinding is the very basic binding - SOAP 1.1, not much in terms of security, not much else in terms of features - but compatible to just about any SOAP client out there --> great for interoperability, weak on features and security
- wsHttpBinding is the full-blown binding, which supports a ton of WS-* features and standards - it has lots more security features, you can use sessionful connections, you can use reliable messaging, you can use transactional control - just a lot more stuff, but wsHttpBinding is also a lot *heavier" and adds a lot of overhead to your messages as they travel across the network