Items	Rest Sharp	Rest Assured

Get Request	RestClient restClient = new RestClient("https://restapi.demoqa.com/utilities/weather/city/");
	RestRequest restRequest = new RestRequest("Guntur",Method.GET);
	IRestResponse restResponse =
		RestAssured.baseURI = "https://restapi.demoqa.com/utilities/weather/city";
		RequestSpecification httpRequest =
		RequestSpecification request = RestAssured.given();
		Response response = request.request(METHOD.GET,"/hyderabad");

Response	string response = restResponse.Content;	string response = response.asString());
Status Code	int StatusCode = (int)restResponse.StatusCode;	int statusCode = response.getStatusCode();
Status Description	string statusDescription = restResponse.StatusDescription;	String statusCode = response.getStatusLine();

AddParameter to request	JObject jObjectbody = new JObject();
	jObjectbody.Add("FirstName", "Narayn");
	jObjectbody.Add("LastName", "Kaluri");
	jObjectbody.Add("UserName", "NaraynKaluri");
	jObjectbody.Add("Password", "Password@123");
	jObjectbody.Add("Email", "abc@hotmail.com");
	// Add the Json to the body of the request
	restRequest.AddParameter("application/json",jObjectbody,ParameterType.RequestBody);
		Map<String, String> params = new HashMap<String, String>();
		params.put("tag", "test");
		params.put("tag2", "test2");
		JSONObject jsonObj = new JSONObject(params);
		-- 	
		// JSONObject is a class that represents a Simple JSON.
		// We can add Key - Value pairs using the put method
		JSONObject requestParams = new JSONObject();
		requestParams.put("FirstName", "Virender");
		requestParams.put("LastName", "Singh");
		requestParams.put("UserName", "simpleuser001");
		// Add the Json to the body of the request
		request.body(requestParams.toJSONString());

Path parameter	var client = new RestClient ("URLName");
	var request = new RestRequest("Resource/{id}", Method.POST);
	request.AddParameter("name", "Yourvalue");
	request.AddUrlSegment("id", 123);
		http://localhost:/api/users/{userid}
		Response response = given()
		.baseUri("https://localhost")
		.basePath("api/users")
		.pathParam("userid", 2)

query parameter	request.AddParameter("foo", "bar", RequestType.QueryString);	http://localhost:8080/users?page=1&limit=50
		Response response = Request.queryParam("page","1")
		.queryParam("limit", "50")
		.get("http://localhost:8080/users");

BasePort		RestAssured.port = 80;

BasePath		http://localhost:/api/users/
		Response response = given()
		.baseUri("https://localhost")
		.basePath("api/users")

rootPath		rootPath - set the default root path of the response body so don't need to write entire path expect(). body("x.y.firstName", equalsTo("some")). body("x.y.lastName", equalsTo(..)). body("x.y.age", equalsTo(..)). body("x.y.gender", equalsTo(..)). when(). get(..); you can use a root and do: RestAssured.rootPath = "x.y"; expect(). body("firstName", equalsTo(..)). body("lastName", equalsTo(..)). body("age", equalsTo(..)). body("gender", equalsTo(..)). when(). get(..);
clear the parameters after the specification is created	request.Parameters.Clear();	RestAssured.**reset**();
redirect		RestAssured.**redirect**(true);

AddJsonBody	var param = new MyClass { IntData = 1, StringData = "test123" };
	request.AddJsonBody(param);
	--- 	
	req.AddParameter("text/xml", body, ParameterType.RequestBody);
	req.AddParameter("application/json", body, ParameterType.RequestBody);
		// Add the Json to the body of the request
		request.body(requestParams.toJSONString());

HTTPS	//bypass ssl validation check by using RestClient object
	var restClient = new RestClient(baseUrl);
	restClient.RemoteCertificateValidationCallback = (sender, certificate, chain, sslPolicyErrors) => true;
	//bypass ssl validation check globally for whole application.
	ServicePointManager.ServerCertificateValidationCallback += (sender, certificate, chain, sslPolicyErrors) => true;
		RestAssured.useRelaxedHTTPSValidation(); //disable usual https verification

add Headers	request.AddHeader("Accept", "application/json");	given().header("MyHeader", "Something")
		given().headers("MyHeader", "Something", "MyOtherHeader", "SomethingElse")

set content type	request.AddHeader("Content-Type", "application/json; charset=utf-8");	Response resp1 = given().header("Content-Type","application/json")
		RestAssured.given().contentType("application/json")
Get Body	string Body = restResponse.Content;	ResponseBody body = response.getBody();
		ResponseBody body = response.body().asString();

Headers	Dictionary<string, string> HeadersList = new Dictionary<string, string>();
	
	foreach (var item in restResponse.Headers)
	{
	string[] KeyPairs=item.ToString().Split('=');
	
	HeadersList.Add(KeyPairs[0],KeyPairs[1]);
	}
	
	Assert.AreEqual("application/json”,HeadersList["Content-Type"],"Content-type not matching");
		Headers allHeaders = response.headers();
		
		// Iterate over all the Headers
		for(Header header : allHeaders)
		{
		System.out.println("Key: " + header.getName() + " Value: " + header.getValue());
		}

	// Parsing JSON content into element-node JObject
	var jObject = JObject.Parse(restResponse.Content);
	
	//Extracting Node element using Getvalue method
	string city=jObject.GetValue("City").ToString();
	// Let us print the city variable to see what we got
	Console.WriteLine("City received from Response " + city);
		// First get the JsonPath object instance from the Response interface
		JsonPath **jsonPathEvaluator** = response.jsonPath();
		
		// Let us print the city variable to see what we got
		System.out.println("City received from Response " + jsonPathEvaluator.get("City"));

Parsing Output	JObject jobj = JObject.Parse(json);
	JArray a = (JArray)jobj["d"];
	IList<Person> person = a.ToObject<IList<Person>>();
	Console.WriteLine(person[0].Name);
	// John Smith
	Console.WriteLin
		import org.json.JSONArray;
		import org.json.JSONObject;
		
		JSONObject obj = new JSONObject(json);
		String pageName = obj.getJSONObject("pageInfo").getString("pageName");
		System.out.println(pageName);
		
		JSONArray arr = obj.getJSONArray("posts");
		for (int i = 0; i < arr.length(); i++) {
		String post_id = arr.getJSONObject(i).getString("post_id");
		System.out.println(post_id);
		}

Post request	RestClient restClient = new RestClient("https://restapi.demoqa.com/customer");
	
	//Creating Json object
	JObject jObjectbody = new JObject();
	jObjectbody.Add("FirstName", "Narayn");
	jObjectbody.Add("LastName", "Kaluri");
	jObjectbody.Add("UserName", "NaraynKasdfsdfluri");
	jObjectbody.Add("Password", "Passwovasdfsdaf23");
	jObjectbody.Add("Email", "abcdafsad@hotmail.com");
	
	RestRequest restRequest = new RestRequest("/register" ,Method.POST);
	
	//Adding Json body as parameter to the post request
	
	restRequest.AddParameter("application/json",jObjectbody,ParameterType.RequestBody);
	
	IRestResponse restResponse =
		RestAssured.baseURI ="https://restapi.demoqa.com/customer";
		RequestSpecification request = RestAssured.given();
		
		JSONObject requestParams = new JSONObject();
		requestParams.put("FirstName", "Virender"); // Cast
		requestParams.put("LastName", "Singh");
		requestParams.put("UserName", "sdimpleuser2dd2011");
		requestParams.put("Password", "password1");
		requestParams.put("Email", "sample2ee26d9@gmail.com");
		
		request.body(requestParams.toJSONString());
		Response response = request.post("/register");
		
		int statusCode = response.getStatusCode();
		System.out.println("The status code recieved: " + statusCode);
		
		System.out.println("Response body: " + response.body().asString());

Deserialize JSON Response	var arr = JsonConvert.DeserializeObject<MyDetail>(response.Content);
	Console.WriteLine(myDetails.FirstName+" , " + myDetails.LastName);
	-- 	
	 public class MyDetail
	 {
	  public string FirstName {
	   get; set;}
	  public string LastName {
	  get;
	  set;}
	}
		RegistrationFailureResponse responseBody = body.as(RegistrationFailureResponse.class);
		 Assert.assertEquals("User already exists", responseBody.FaultId);
		 Assert.assertEquals("FAULT_USER_ALREADY_EXISTS", responseBody.fault);
		 -- 	
		public class RegistrationFailureResponse {
		 String FaultId;
		 String fault;
		}

Basic Authorization	restClient.Authenticator = new HttpBasicAuthenticator("ToolsQA", "TestPassword");
	-- 	
	var client = new RestClient("http://example.com");
	client.Authenticator = new SimpleAuthenticator("username", "foo", "password", "bar");
	
	var request = new RestRequest("resource", Method.GET);
	client.Execute(request);
		RestAssured.authentication = basic("username", "password");

Preemptive Authentication		RestAssured.authentication = preemptive().basic("username", "password"); send the credentials withut waiting for an unauthorized response

Digest Authentication		given()auth()
		.digest("username", "password")
		.when
		sends the password in encoded form avoids cleartext PWD

OAuth2	var client = new RestClient("http://example.com/myapi/oauth/token");
	RestRequest request = new RestRequest() { Method = Method.POST };
	request.AddHeader("content-type", "application/x-www-form-urlencoded");
	request.AddHeader("Accept", "application/json");
	request.AddParameter("grant_type", "client_credentials");
	request.AddParameter("client_id", "client-app");
	request.AddParameter("client_secret", "secret");
	var response = client.Execute(request);
	var token = JsonConvert.DeserializeObject<Dictionary<string,object>>(response)["access_token"].ToString();
	return token.Length > 0 ? token : null;
		RequestSpecification request = RestAssured.given().auth()
		.oauth2(accessToken)
		.when()

Parse XML	XmlDocument doc = new XmlDocument();
	doc.Load("d:\\product.xml");
	XmlNodeList nodes = doc.DocumentElement.SelectNodes("/Store/Product");
	string product_id = "", product_name = "", product_price="";
	foreach (XmlNode node in nodes)
	{
	product_id = node.SelectSingleNode("Product_id").InnerText;
	product_name = node.SelectSingleNode("Product_name").InnerText;
	product_price = node.SelectSingleNode("Product_price").InnerText;
	MessageBox.Show(product_id + " " + product_name + " " + product_price);
	}
	--- 	
	XmlDocument _doc = new XmlDocument();
	_doc.LoadXml(xmlText);
	XmlNodeList _fnames = _doc.GetElementsByTagName("Product_name");
		<custmer>
		<id>1212</id>
		<firstName>john</firstName>
		<lastName>Smith</lastName>
		</custmer>
		
		import io.restassured.path.xml.XmlPath;
		String xml = get("http://parabank.parasoft.com/parabank/services/bank/customers/12212/").andReturn().asString();
		// Retrieve the values to be checked from the XML as a String
		XmlPath xmlPath = new XmlPath(xml).setRoot("customer");
		String cusID = XmlPath.from(xml).get("customer.id");
		String firstName = xmlPath.getString("firstName");
		String lastName = xmlPath.getString("lastName");
		--- 	
		String stringResponse = validation.asString();
		XmlPath xmlPath = new XmlPath(stringResponse);
		String status = xmlPath.get("ncresponse.@STATUS");

Request XML response	request.RequestFormat = DataFormat.Xml;	Response res =given ().param ("query", "Churchgate Station")
		.param ("key", "Xyz")
		.when()
		.get ("/maps/api/place/textsearch/xml")
		.then()
		.contentType(ContentType.XML)
		.extract().response();
		System.out.println (res.asString ());

Parse Json by JSONpath	string jsonPath = "$..Products[?(@.Price >= 50)].Name";
	JObject jObject = JObject.Parse(restResponse.Content);
	IEnumerable<JToken> parseVal = jObject.SelectTokens(jsonPath);
	string jsonValue = string.Join(",",parseVal);
		String pageName = JsonPath.read(json, "$.pageInfo.pageName");
		System.out.println(pageName);
		
		Integer posts = JsonPath.read(json, "$.posts.length()");
		for(int i=0; i < posts; i++) {
		String post_id = JsonPath.read(json, "$.posts[" + i + "].post_id");
		System.out.println(post_id); }

PUT request + Serialization	{
	static void Main(string[] args)
	{
	var client = new RestClient("http://192.168.1.123:8800/");
	var request = new RestRequest("/api/v1/apps/", Method.POST) { RequestFormat = DataFormat.Json };
	request.AddBody(new FromBodyData() { Counter = 10 });
	var response = client.Execute(request);
	if (response.StatusCode == HttpStatusCode.Created)
	{
	Console.WriteLine("'{0}' deployment started remotely on '{1}'.", appName, hostName);
	}
	}
	-- 	
	------------------------ 	
	public class FromBodyData
	{
	public int Counter { get; set; }
	}
	}
		Serializing POJOs in REST Assured
		public class Car {
		
		String make;
		String model;
		int year;
		
		public Car() {
		}
		
		public Car(String make, String model, int year) {
		this.make = make;
		this.model = model;
		this.year = year;
		}
		
		public String getMake() {
		return this.make;
		}
		
		public void setMake(String make) {
		this.make = make;
		}
		
		public String toString() {
		return "My car is a " + this.year + " " + this.make + " " + this.model;
		}
		}

OAuth1	var client = new RestClient("https://api.xing.com")
	{
	Authenticator = OAuth1Authenticator.ForProtectedResource(<ConsumerKey>,<ConsumerSecret>,<AccessToken>,<AccessTokenSecret>)
	};
	var request = new RestRequest(XingEndpoints.UsersMeContacts);
	request.Method = Method.GET;
	request.AddQueryParameter("limit", "0");
	// Same with request.AddParameter
	var response = client.Execute(request);
		given().accept(ContentType.JSON)
		.auth()
		.oauth(consumerKey, consumerSecret, accessToken, tokenSecret)

Token based Authoziation	var client = new RestClient("https://api.us.onelogin.com/auth/oauth2/token");
	var request = new RestRequest(Method.POST);
	request.AddHeader("cache-control", "no-cache");
	request.AddHeader("content-type", "application/json");
	request.RequestFormat = DataFormat.Json;
	
	request.AddHeader("authorization", "client_id:yourClientID, client_secret:yourClientSecret");
	
	request.AddParameter("application/json", "{\n\"grant_type\":\"client_credentials\"\n}", ParameterType.RequestBody);
	IRestResponse response = client.Execute(request);
	
	RootObject returnData = deserial.Deserialize<RootObject>(response);
	if (returnData.data[0].access_token != null)
	{
	access =
		RequestSpecification request = RestAssured.given().auth()
		.oauth(consumerkey,consumerSecret,accessToken,tokenSecret)
		.when()

Authenticators	HttpBasicAuthenticator IAuthenticator NtlmAuthenticator OAuth1Authenticator OAuth2Authenticator SimpleAuthenticator	Basic Authentication Digest Authentication Form Authentication OAuth 1 and OAuth 2