

# Lightning Connect vs REST callouts

Some weeks ago, in [this post](#), I tried to solve one of the questions that was asked to me in London’s Calling presentation about how to unit test Lightning Connect apps. There was still another question in the session that encouraged me to perform an investigation on the differences in performance of using a Lightning Connect custom adapter vs implementing the same behaviour through REST callouts.

The investigation has consisted of performing a set of measures using three different scenarios: REST Callouts, Lightning Connect custom adapter with high-volume external datasource and Lightning Connect custom adapter with no high-volume external datasource. If you don’t know the differences between a high-volume and a no high-volume external datasource, you can take a look at [this post](#).

Concretely, for each of the scenarios I have measured:

- **Total time:** time since the restaurant service class performs either the callout or the SOQL query requests until the list of restaurants is returned to it.
- **Network time:** time since the Http request is performed until the Http response is deserialised into a list of restaurants, which is the same piece of code in the three cases.

I have measured the network time in order to be able of discriminating possible delays of my network connection. This is, I will determine the average cost of each of the options calculating the (**total time – network time**). These are the results I have obtained for ten different loads of the same data (a list of nearby restaurants) in the three different cases I have described:

Callouts			LC + no high volume			LC + high volume		
Total	Network	Diff	Total	Network	Diff	Total	Network	Diff
290	266	24	351	294	57	241	206	35
446	434	12	192	157	35	182	151	31
403	380	23	280	230	50	199	172	27
264	248	16	137	96	41	302	274	28
399	382	17	172	138	34	213	177	36
118	97	21	246	194	52	314	279	35
305	291	14	300	225	75	150	128	22
202	189	13	331	256	75	210	179	31
380	367	13	227	183	44	284	262	22
260	238	22	376	343	33	264	233	31
	AVG	17.5		AVG	49.6		AVG	29.8

As you can see, **using a Lightning Connect custom adapter implies an overhead over using REST callouts in all the cases**. This result makes sense, as there is an extra processing that the custom adapter code has to do to convert the SOQL query in a callout before performing the Http request.

We can see as well that **using a high volume external datasource gives better results than using a non-high volume one**. As I spotted in my post about high-volume datasources, in a no high-volume datasource Salesforce assigns an Id over the scenes to each one of the external objects you access in the system (it stores an equivalence between the real ExternalIds of the objects and the “virtual” Ids it had assigned to them somehow). So, probably, the extra time taken by the no high-volume case is the time invested in creating this virtual Id an storying it in a table.

The changes I have made in the application code for doing the measurements can be seen in [this commit](#).

So, why would I prefer to implement a **Lightning Connect custom adapter vs using REST callouts** directly if the performance I get can be lower? With Lightning Connect we have the ability of directly using Salesforce capabilities over our external objects without having to store them in the system. This is, external objects:

- Can be directly used in **list views, detail pages and SOQL**.
- You can create **relationships** between External Objects and Standard or Custom Objects to seamlessly integrate legacy data.
- **Salesforce SOAP and REST APIs** are available for external objects.
- **Field Level Security** is also available for them.
- If you use a **no high-volume external datasource**, you can even enable **Chatter feeds** on External Object pages or navigate them in **Salesforce1**.

Obviously, if your **business requirements** don’t need any of these Salesforce capabilities, **going for REST callouts can be the option** for you. Indeed in the [application](#) I developed for learning Lightning Connect I use two visualforce pages for showing data from the external system, so, using REST callouts would have been a good option for me!

And, instead of querying my external system with **Lightning Connect, what about importing the external data into custom objects** in Salesforce? If you use Lightning Connect:

- **You don’t need storage:** data is accessed in real-time from your external system, so, no storage is needed in Salesforce at all.
- **You don’t need synchronization:** you don’t need to perform any synchronization between your external system and Salesforce, as data lives in your external system!

In some cases you might consider the option of importing your data to Salesforce, and compare the **pricing of Lightning Connect vs the pricing of storing the copied data**, but bear in mind that if your external data changes, or if it has to be refreshed, for example, because new records are created in Salesforce, you will need some kind of synchronization between Salesforce and the external system, with the problems that this can entail.

In conclusion, you should study your concrete business case in order to decide whether to use Lightning Connect, REST callouts or copy your data into Salesforce. Know first all the advantages and drawbacks of each option, and then proceed!