DIFFERENCE BETWEEN PUT AND POST VERBS IN REST SERVICES

Both PUT and POST can be used for creating.

You have to ask "what are you performing the action to?" to distinguish what you should be using. Let's assume you're designing an API for asking questions. If you want to use POST then you would do that to a list of questions. If you want to use PUT then you would do that to a particular question.

**Great both can be used, so which one should I use in my RESTful design:**

You do not need to support both PUT and POST.

Which is used is left up to you. But just remember to use the right one depending on what object you are referencing in the request.

Some considerations:

* Do you name your URL objects you create explicitly, or let the server decide? If you name them then use PUT. If you let the server decide then use POST.
* PUT is idempotent, so if you PUT an object twice, it has no effect. This is a nice property, so I would use PUT when possible.
* You can update or create a resource with PUT with the same object URL
* With POST you can have 2 requests coming in at the same time making modifications to a URL, and they may update different parts of the object.

**An example:**

[I wrote the following as part of another answer on SO regarding this](http://stackoverflow.com/questions/256349/what-are-the-best-common-restful-url-verbs-and-actions/256359#256359):

**POST:**

Used to modify and update a resource

POST /questions/<existing\_question> HTTP/1.1

Host: www.example.com/

Note that the following is an error:

POST /questions/<new\_question> HTTP/1.1

Host: www.example.com/

If the URL is not yet created, you should not be using POST to create it while specifying the name. This should result in a 'resource not found' error because <new\_question> does not exist yet. You should PUT the <new\_question> resource on the server first.

You could though do something like this to create a resources using POST:

POST /questions HTTP/1.1

Host: www.example.com/

Note that in this case the resource name is not specified, the new objects URL path would be returned to you.

**PUT:**

Used to create a resource, or overwrite it. While you specify the resources new URL.

For a new resource:

PUT /questions/<new\_question> HTTP/1.1

Host: www.example.com/

To overwrite an existing resource:

PUT /questions/<existing\_question> HTTP/1.1

Host: www.example.com/