| PUT | POST |
| --- | --- |
| RFC-2616 clearly mention that PUT method requests for the attached entity to be stored into the server under the supplied [Request-URI](https://restfulapi.net/resource-naming/).  If the Request-URI refers to an already existing resource – an update operation will happen, otherwise create operation should happen if Request-URI is a valid resource URI (assuming the client is allowed to determine resource identifier). | The POST method is used to request that the origin server accept the entity attached in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.  It essentially means that POST request-URI should be of a collection URI. |
| PUT method is [idempotent](https://restfulapi.net/idempotent-rest-apis/). So if you send retry a request multiple times, that should be equivalent to a single request modification. | POST is NOT idempotent. So if you retry the request N times, you will end up having N resources with N different URIs created on the server. |
| Use PUT when you want to modify a singular resource that is already a part of resources collection. PUT replaces the resource in its entirety. Use PATCH if request updates part of the resource. | Use POST when you want to add a child resource under resources collection. |
| Though PUT is idempotent, we should not cache its response. | Responses to this method are not [cacheable](https://restfulapi.net/caching/), unless the response includes appropriate Cache-Control or Expires header fields.  However, the 303 (See Other) response can be used to direct the user agent to retrieve a cacheable resource. |
| Generally, in practice, always use PUT for UPDATE operations. | Always use POST for CREATE operations. |