* Instructor: Now that we had a look at how we can send requests, how we can post data, how we can get data let's finish this basic request sending up by making sure that we can also clear our posts so that we can delete data.
* And for that, of course, definitely feel free to pause this video and try it on your own first.
* Try making sure that whenever this onClearPosts method runs, which will happen when we click that, Clear Posts button up there, that whenever this method runs that you send a request that deletes all the posts on the backend.
* You learned everything you need for that, so here's your chance to pause the video and try it on your own.
* Were you successful? Well, it's not that hard.
* It's almost the same type of request we sent before.
* Now let's keep the structure of sending the requests in the post service.
* So in there I will add a new method which I'll name, deletePosts, of course, you can name it also clearPosts or whatever you want.
* Now in there I will use the HTTP client because that is our central client for all HTTP requests you wanna send from inside our Angular app.
* And there we can now use the delete method which will send a, well, delete request.
* Now this method requires a URL to which it should send the request.
* And since I want to clear all posts, I will grab this URL which targets the overall posts node and send my request there, so that all the posts there get deleted.
* Now, if I want to be informed about that deletion process in the component, which in this case I want, but if you chose a different solution, that's fine too, but if I want to be informed in a component, I will return my observable here and I will not subscribe here in the service.
* But instead now in the app component in onClearPosts I can reach out to the post service and call deletePosts and since this returns an observable, we now have to subscribe here.
* Now why do I want to subscribe here? Well, if I deleted all posts, I also want to clear my loadedPosts array here in the component.
* So I will add a method here where I don't really care about the result of our request or of the response but I know that this function here will only run if it's succeeded and therefore here I will then simply just set this loadedPosts equal to an empty array again to reset it.
* And with these changes to the service and to the component, if we go back to our application and we let this reload, if it click Clear Posts, you see, eventually it gets rid of all the posts and on Firebase, we also see that this posts node does not exist anymore.
* And, of course, we can still add new posts here.
* So all our old functionality still is there.
* The only thing we have, of course, is that currently we got no logic to automatically update our list down there once we added a new post.
* Now you can definitely score some bonus point if you implement this.
* I'll not do it here because it's not really related to the fetching of data, just to how you react to the sending of a post being done.
* So with that, we had a thorough look at different kinds of requests we can send.
* Now what if something goes wrong?