Recap: why does an HTML form only allow GET and POST

I know we've been through a ton of information, so I'll keep this brief.

HTML forms play a very important role because it allows a user to send data to an HTTP server.

Duh Clyde, pretty obvious!

But ***how*** the form data is sent to the server is not determined by the user, but by the forms *method* attribute. And this is defined by the designer of the form – the developer, which is you.

Current position

Right now, the HTML rules for forms only allow us to define
a **GET** or **POST** (and yes, DIALOG) method.

Arguments in Favor of having additional method types

There has been a lot of debate around only having **GET** or **POST** for a very long time.
One argument in Favor of allowing other method types is that many (front-end) developers do not understand server-side code. How then will the front-end developer allow the user of the form to **DELETE** or **PUT** a resource when the only methods are **GET** or **POST**?
Answer – they can't.

In other words, if a site is relying on a static/traditional HTTP server that depends on the available HTTP methods to fulfil a request, then HTML as it currently stands will **not** allow for full interaction with the resources on that server.
Ideally, it should be possible to put a HTML form in front of any HTTP server supporting **GET**, **POST**, **PUT**, and **DELETE**, and have that form be able to perform any of those actions without the need for server-side code that falls outside the HTTP specification.
However, right now we're stuck with having to use only GET and POST.

So then, this begs the question whether the <form> element is really that useful?

Is the <form> element still useful?

You may be wondering then: “If we aren’t able to send a DELETE request, how is this action performed?”

The answer is that we have work arounds.

We can use AJAX, as we spoke about earlier.
Server-side programming languages (like Node.js) and programmers have also made special provision allowing us to get by without **PUT** and **DELETE** support in HTML. But talking about server language here is beyond the scope of this course.
Bottom line: although we are unable to precisely define the method type of a <form> beyond GET and POST, it doesn’t stop us from getting the job done. So, in reality, only being able to use POST and GET does not affect us at all. In other words, the <form> element is still highly relevant and powerful.
I hope you find all of this as interesting as I do ;)

See you in the next lectures.